Universidade de Aveiro Departamento de Electrónica, Telecomunicações e
2013 Informática

Daniel Filipe Albuquerque

Sistema de Localização com Ultrassons

Ultrasonic Location System

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2013

Daniel Filipe Albuquerque

Sistema de Localização com Ultrassons

Ultrasonic Location System

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Electrotécnica, realizada sob a orientação científica do Doutor Paulo Jorge dos Santos Gonçalves Ferreira, Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, do Doutor José Manuel Neto Vieira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Carlos Alberto da Costa Bastos, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

The thesis research was supported by Fundação para a Ciência e a Tecnologia with the doctoral grant: (SFRH/BD/45560/2008). Which is funded by Programa Operacional Potencial Humano of QREN Portugal 2007-2013 and by the national budget through MCTES.

To my respected parents, Acácio and Helena, to my supportive brother and sister, Diogo and Diana, and to my loving wife, Luciana.

o júri / the jury

presidente / president

Prof. Doutor Jorge Adelino Rodrigues da Costa
Professor Catedrático da Universidade de Aveiro.

vogais / examiners committee

Prof. Doutor Fernando Xavier Álvarez Franco
Professor Titular da Universidade de Extremadura de Espanha.

Prof. Doutor Paulo Jorge dos Santos Gonçalves Ferreira
Professor Catedrático da Universidade de Aveiro.

Prof. Doutor Manuel José Cabral dos Santos Reis
Professor Associado com Agregação da Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro.

Prof. Doutor José Manuel Neto Vieira
Professor Auxiliar da Universidade de Aveiro.

Prof. Doutor Carlos Alberto da Costa Bastos
Professor Auxiliar da Universidade de Aveiro.

Prof. Doutor João Manuel de Oliveira e Silva Rodrigues
Professor Auxiliar da Universidade de Aveiro.

agradecimentos / acknowledgements

First and foremost, I would like to express my gratitude to my supervisors Professor Paulo Jorge Ferreira, Professor José Neto Vieira and Professor Carlos Bastos for their constant support and knowledge during my doctoral research. I also thank Professor José Neto Vieira for his motivation and patience to receive me every week and sometimes three-four times a week.

I would also like to thank Professor Fernando Álvarez from the University of Extremadura that generously received me and exchanged some ideas in my research field. I would like to extend my thanks to the following professors from the University of Aveiro, Professor Guilherme Campos, Professor João Rodrigues, Professor Tomás Oliveira e Silva, Professor Paulo Pedreiras and Professor Francisco Vaz for their encouragement and insightful comments during the period of my research.

I sincerely acknowledge my PhD colleagues, Isabel Duarte, Sérgio Lopes, Bruno Jesus, Ricardo Correia, Miguel Drummond and Nelson Silva a source of friendship, advices and collaboration. I also like to acknowledge my graduation friends: Ana Santos, André Gomes, António Alves, Daniel Macedo, Daniel Paulino, Liliana Costa, Miguel Veiga, Ricardo Filipe, Ricardo Rolhas and Sérgio Tafula for the great time we had during those five years. I also thank my closest friends, André and Catarina for all the support and friendship that they gave me during this journey.

I would like to extend my sincere thanks to University of Aveiro especially the Department of Electronics, Telecommunications and Informatics for supporting this work. I also extend my thanks to the Portuguese Foundation for Science and Technology for funding my work with a scholarship.

Finally, I want to thank all my friends and family, especially my wife, for their encouragement and constant support to finish this thesis with success.

palavras-chave

Localização, Ultrassons, Propagação Acústica, Acústica de Salas, Efeito de Doppler, Tempo de Voo, Sincronização de Relógios, OFDM, Deteção de Pulso, Comunicação Assíncrona, Resposta Impulsional.

resumo

Esta tese apresenta um sistema de localização baseado exclusivamente em ultrassons, não necessitando de recorrer a qualquer outra tecnologia. Este sistema de localização foi concebido para poder operar em ambientes onde qualquer outra tecnologia não pode ser utilizada ou o seu uso está condicionado, como são exemplo aplicações subaquáticas ou ambientes hospitalares. O sistema de localização proposto faz uso de uma rede de faróis fixos permitindo que estações móveis se localizem. Devido à necessidade de transmissão de dados e medição de distâncias foi desenvolvido um pulso de ultrassons robusto a ecos que permite realizar ambas as tarefas com sucesso. O sistema de localização permite que as estações móveis se localizem escutando apenas a informação em pulsos de ultrassons enviados pelos faróis usando para tal um algoritmo baseado em diferenças de tempo de chegada. Desta forma a privacidade dos utilizadores é garantida e o sistema torna-se completamente independente do número de utilizadores. Por forma a facilitar a implementação da rede de faróis apenas será necessário determinar manualmente a posição de alguns dos faróis, designados por faróis âncora. Estes irão permitir que os restantes faróis, completamente autónomos, se possam localizar através de um algoritmo iterativo de localização baseado na minimização de uma função de custo. Para que este sistema possa funcionar como previsto será necessário que os faróis possam sincronizar os seus relógios e medir a distância entre eles. Para tal, esta tese propõe um protocolo de sincronização de relógio que permite também obter as medidas de distância entre os faróis trocando somente três mensagens de ultrassons. Adicionalmente, o sistema de localização permite que faróis danificados possam ser substituídos sem comprometer a operabilidade da rede reduzindo a complexidade na manutenção. Para além do mencionado, foi igualmente implementado um simulador de ultrassons para ambientes fechados, o qual provou ser bastante preciso e uma ferramenta de elevado valor para simular o comportamento do sistema de localização sobre condições controladas.

keywords

Location, Ultrasounds, Acoustic Propagation, Indoor Acoustics, Doppler Effect, Time of Flight, Clock Synchronization, OFDM, Pulse Detection, Asynchronous Communication, Impulse Response.

abstract

This thesis presents a location system based exclusively on ultrasonic signals, without using any other technology. This location system was designed to operate in environments where the use of other technologies is not possible or the use of them is limited, such as underwater applications or hospital environments. The proposed location system uses a network of fixed beacons allowing the mobile stations to locate. Due to the necessity of data transmission and distance measurement an ultrasonic pulse robust to echoes was developed that allows to perform both tasks with success. The location system allows that mobiles locate themselves only listening to the information in the ultrasonic pulse sent by the beacons, for that an algorithm based on time difference of arrival is used. Therefore, the user privacy is guaranteed as well as the complete independence of the system number of users. To simplify the network implementation it is only necessary to manually define the position of some of the beacons, called anchor beacons. These will allow the remaining autonomous beacons to locate themselves by an iterative location algorithm based on a local cost function minimization. For this system to work properly the beacons must synchronize their clocks and measure the distance between them. Therefore, this thesis proposes a clock synchronization protocol which also allows to measure the distance between the beacons by exchanging only three ultrasonic messages. Additionally, the location system permits that damaged beacons may be replaced without compromising the network operability reducing the maintenance complexity. Additionally, a simplified ultrasonic simulator for indoor environments was developed, which has proved to be very accurate and a valuable tool to simulate the location system behavior under controlled conditions.

CONTENTS

LIST OF FIGURES

This page intentionally contains only this sentence.

List of Tables

This page intentionally contains only this sentence.

ACRONYMS

1D one dimensional. 16

2D two dimensional. 11, 14–16, 38, 99, 127, 130

3D three dimensional. 11–13, 17, 35, 58, 127, 130

ACF auto-correlation function. 75, 77, 78, 85

AOA angle of arrival. 10, 11, 14

BER bit error rate. 61, 88–94, 96, 176

BPSK binary phase shift keying. 13, 53, 58, 62, 63, 88

CDMA code division multiple access. 12, 13

DBPSK differential binary phase shift keying. 3, 4, 88–92, 94, 96, 176

DFT discrete Fourier transform. 33, 66, 67, 162

DSSS direct-sequence spread spectrum. 12

dw-MDS distributed weighted-multidimensional scaling. 128, 130, 144, 177

FDMA frequency division multiple access. 125

FFT fast Fourier transform. 61, 90, 91

FHSS frequency hopping spread spectrum. 13

FIR finite impulse response. 66

FT Fourier transform. 69, 153

GNSS global navigation satellite systems. 2

GPS global positioning system. 2, 7, 108

GSM global system for mobile communications. 7

ID identification. 10, 11, 14, 110, 111, 171

IDFT inverse discrete Fourier transform. 33, 66, 67

IFFT inverse fast Fourier transform. 61, 62

IFT inverse Fourier transform. 31, 161

IR impulse response. 21–24, 26, 31–34, 37, 38, 44, 47, 50, 53, 91–94, 153, 154, 157, 176

ISI inter-symbolic-interference. 61, 63, 65, 91, 92, 94, 96

LOS line-of-sight. 157

LS location system. 1–3, 7–16, 18

MDF medium-density fiberboard. 156

MDS multidimensional scaling. 127, 128

MEMS micro electro-mechanical systems. 14

MF matched filter. 66–68, 72, 75, 78, 85, 87, 151, 163, 165

MSR main-to-side lobe ratio. 75–77, 85

NLOS non-line-of-sight. 123, 157

OFDM orthogonal frequency division multiplexing. 3, 4, 61–63, 66–68, 75–79, 81–83, 85–92, 94, 96, 150, 151, 153, 164, 165, 176, 178

PAPR peak-to-average power ratio. 79–83, 86, 150, 176, 178

PC personal computer. 10, 11, 17, 48, 147

PL powerline. 14

PLP powerline positioning. 14

PMEPR peak-to-mean envelope power ratio. 80–83, 86

PSK phase shift keying. 61, 88

QAM quadrature amplitude modulation. 61, 88

RF radio frequency. 2–4, 9, 11–14, 17, 18, 21, 24, 105, 175–177

RFID radio frequency Identification. 14–16

SNR signal to noise ratio. 66, 69, 70, 72–74, 77, 78, 85, 90–93, 163

SPL sound pressure level. 53

TDMA time division multiple access. 12, 13, 125

TDOA time difference of arrival. 3, 4, 17, 99, 126, 177

TOA time of arrival. 11, 126, 134

TOF time of flight. 3, 4, 9, 11, 12, 14, 17, 18, 61, 62, 65, 75, 76, 78, 87, 116, 124, 150, 152, 164, 176, 178

TPSN timing-sync protocol for sensor networks. 4, 107, 110–112, 177

UDP user datagram protocol. 15

US ultrasounds. 9–14, 17, 18, 21, 24, 58, 61, 63, 65, 80, 87, 94, 96, 100, 101, 105, 113, 121, 123

UWB ultrawideband. 2

WiFi wireless fidelity. 15

WLAN wireless local area network. 14–16

WSN wireless sensor network. 104, 105, 107

This page intentionally contains only this sentence.

List of Symbols

α absorption coefficient in air. 28–31, 64, 65

P acoustic air pressure. 25, 27–31

T air temperature. 25, 26, 28, 30

A_T atmospheric attenuation. 31

χ_2^2 chi-squared distribution with 2 degrees of freedom. 69, 70, 72

n_τ delay in samples due to the wave time of flight. 65

$\delta_{i,j}$ distance measurement between the node i and j. 124

d distance between two nodes. 106, 109, 124

f_D Doppler frequency shift. 52, 58

$d_{i,j}$ euclidean distance between the position i and j. 124

ζ local clock for each node. 103, 104

ϕ local clock offset. 103, 104, 106, 107, 109, 126

f_{max} maximum frequency of the signal. 64

C node coordinates. 111, 124, 125

L number of beacons. 126

N_c number of carriers. 64, 67–71, 73, 76, 77, 89, 90

P_d probability of detection. 70–73, 75, 85

P_{fa} probability of false alarm. 70–73, 75, 85

τ propagation delay due to the wave time of flight. 106, 109, 126

T_p pulse duration. 64

N pulse size. 67, 70, 73, 75, 77, 89, 90

v receiver/source speed. 52

v_r relative speed between the source and the receiver. 52, 64

f_s sampling frequency. 32, 65, 90

σ^2 signal variance. 65, 69–71

B system bandwidth. 64, 90

γ threshold. 68, 70–72, 75, 90

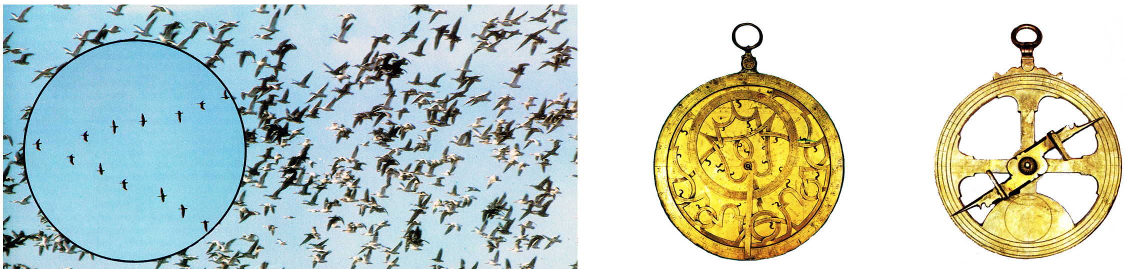
ς time stamp. 105–107, 109, 111, 117, 125, 126

c ultrasonic wave speed in air. v, 25, 26, 37, 52, 64, 107, 126

CHAPTER 1

INTRODUCTION

The need for location has been a companion of animals and humans during their evolution path. It has been used to get food and migrate to find better life conditions. For example, geese could travel several hundreds of kilometers keeping their relative position constant in the formation. Moreover, they could return to their initial position after traveling some thousands of kilometers without getting lost, see Figure 1.1a [1]. More recently, location techniques and systems, based on the Sun and stars, were used to conquer the seas and traveling far distances in open seas without getting lost. To accomplish that, several complex instruments were invented, Figure 1.1b presents an example. These instruments were used with two main goals: to display how the sky was or will be at a specific position or to obtain the latitude for the current position. The latitude could be easily obtain by observing the only available references, the sun during the daylight and the stars at night. Therefore, they were very used mainly by the Portuguese and the Castilians during the beginning of discoveries. Nevertheless, they had a huge problem, it was impossible to obtain longitude. Therefore to reach a specific place the ship was sailed to the desired latitude and then it must be sailed along the latitude line, to east or west, until reaching the desired destination. Moreover they could not be used if the sky was not clear. Since then, in order to accommodate the most demanding human needs for location, science has been providing new ways using several technologies for different purposes.



(a) Geese travel around 2700 Km in their migration. After the initial mess the geese rapidly get their known V formation (see the circle in the picture). Source: [1].

(b) Astrolabe (left) used to locate and predict the position of celestial objects, and Mariner's Astrolabe (right) used to obtain the latitude of a ship at sea. Source: [2].

Figure 1.1: Location is a valuable tool for animals and for humans.

Nowadays, location systems (LSs) are a built-in feature of several complex systems, such as robots. It is present in the automation industry, autonomous navigation, emergency callers, exploration of unreachable places on Earth and space, etc... Even at home, where robots start to have a daily presence, we have autonomous mowers able to cut the grass, vacuum cleaners that can clean the floor without human intervention and so many other examples. To exemplify the present day technology diversity, Figure 1.2 presents the technology currently used in several LSs for indoor and outdoor applications and their accuracy and maturity. The maturity was divided in three categories: *Research Labs*, *Custom Systems* and *Consumer Systems*. The *Research Labs* represents all systems that are still in the research laboratories and so not available to public. The *Custom Systems* represents all system that are available but their use is restrict or very specific to an application. Finally, the *Consumer Systems* represents all the systems that are easily available and could be found in several everyday objects. For outdoor location, the location systems are dominated by the global navigation satellite systems (GNSS) (where global positioning system (GPS) is the most known and used). The GNSS are based in the same principle of the ancient location systems, measuring the distance to artificial celestial objects. Nevertheless, there are several solutions based on cellular network infrastructure or wireless networks profiling that could improve the resultant position estimation [4]. In addition to those technologies (see Figure 1.2) there are also some custom systems that could be used to provide an accurate location (ultrawideband (UWB)) or a long battery life (Sunlight). However, those technologies are not provided in our everyday objects and their applicability is very specific.

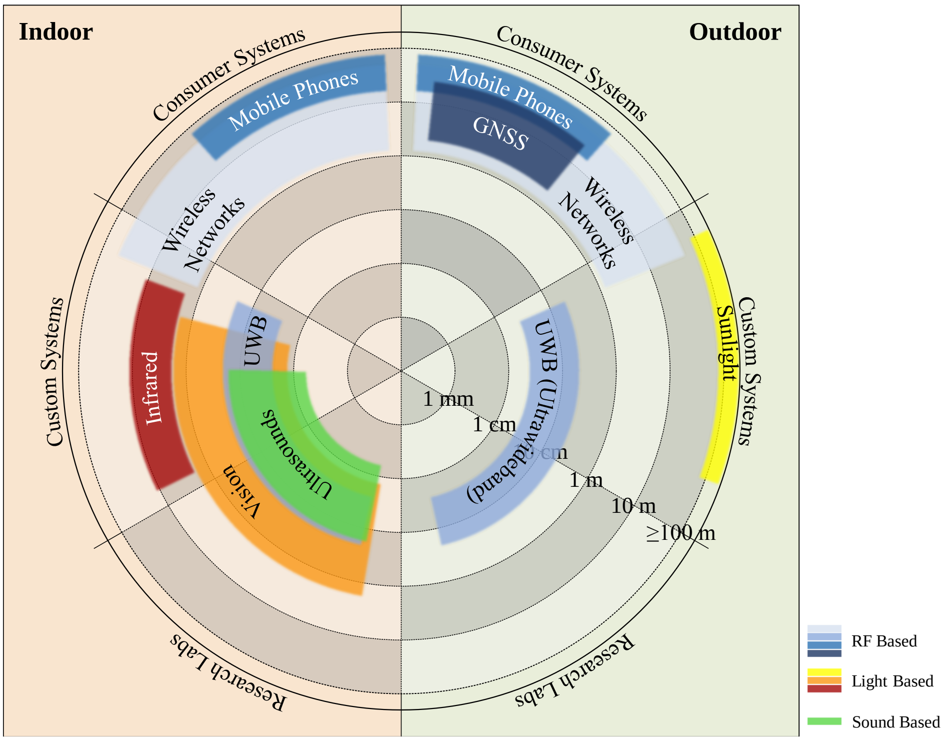


Figure 1.2: The accuracy and maturity of several technologies currently used for indoor and outdoor location. Information collected from [3, 4, 5].

As it can be seen in Figure 1.2 the outdoor location systems are already in a mature stage, however the same cannot be said for indoor location systems, where the mature system are not able to provide accurate indoor location information due to fading and multipath effects of the buildings on the radio frequency (RF) signal [6].

As a result of the increasing necessity to provide an indoor location system that is adequate to locate people, objects (e.g., robots) inside a building (e.g., factory, hospital, school, home, etc.) it is important to provide a LS that is able to provide a position with an accuracy in the centimeter order. Regarding the technology chart present in Figure 1.2 the best suited technology is the ultrasounds. Although, the ultrasonic based custom systems and the majority of the system in laboratory use an auxiliary RF-channel to perform location or the user needs to provide a set of a priori information.

The main goal of this thesis is the development of an indoor location system using only ultrasonic signals, resulting in a RF-free location system. The proposed LS is manly designed to be implemented in indoor environments, nevertheless, it suits any environment where the use of RF is not allowed or it is strongly mitigated and the ultrasonic propagation is favorable. One of the best examples of this environments is underwater. In such environment the RF waves are strongly attenuated due to the water conductivity, especially in the seawater case [7]. Additionally, the ultrasonic propagation in water is more favorable than in air, ultrasonic signals travel 4 times faster in water than in air.

In the proposed LS, the ultrasonic signals will be used to get distance information, from time of flight (TOF), and also to implement data communication. For data communication and TOF measurement, an ultrasonic pulse that uses orthogonal frequency division multiplexing (OFDM) and differential binary phase shift keying (DBPSK) is proposed. This proposed pulse was designed to be robust to the acoustic indoor environment demands.

The proposed location system uses a support network of fixed beacons, attached to the ceiling or walls of a room, and mobiles that need location information inside that particular room. In the network of beacons there are some special beacons, called anchor beacons. The anchor beacons have a predetermined position and they are the reference for all the network to compute the other beacons position. Each beacon is completely autonomous and as a result of this a clock synchronization protocol was proposed, this protocol will allow, not only, beacons to synchronize their clocks but also to obtain the distance measurements between them by exchanging only three messages from time to time. The beacons coordinates are obtained by the minimization of a local cost function using an iterative algorithm allowing to take advantage not only from the distance measurements to its neighbors but also from the distance measurements between the neighbors. This approach reduces human intervention during the network implementation and failure beacons can be easily replaced without compromising the network operation. Mobiles are responsible to locate themselves applying a very simple algorithm based on the time difference of arrival (TDOA). This algorithm uses the ultrasonic messages received from the nearby beacons with the message time-stamp and the beacon's coordinates. This approach allows the location system to be independent of the number of mobiles and will provide a high degree of privacy since the mobiles do not need to transmit any message.

In addition to this, a simplified model of the acoustical channel was developed resulting in an ultrasonic simulator for indoor environments. This simulator proved to be very accurate and a valuable tool to simulate the location system behavior under controlled conditions.

1.1 Objectives

The Ph.D. thesis work has been concentrated into achieve the following, initially proposed, objectives:

- A simplified simulator for ultrasound acoustical propagation in indoor environments;
- A technique to perform TOF measurements and data transmission over the shared ultrasonic acoustic channel;

- A method to perform location using ultrasonic signals and avoiding the use of an auxiliary radio frequency channel;

- A distributed algorithm to perform a mobile location using the TOF measurements from the network of beacons.

1.2 Contributions

This Ph.D. work presents several contributions that will be detailed. The first contribution is an ultrasonic room simulator that is able to deal with high bandwidth ultrasonic signals. This simulator is capable of handling the multiple reflections present in a room which could be produced by the room walls or by the objects inside it. Moreover, it takes into account the beam patterns of the used transducers as well as the ultrasonic wave propagation losses. Additionally the simulator may be used to simulate the Doppler effect for any movement. In the end, the proposed simulator proved to be a valuable tool to study and compare different location algorithms and acoustic pulses under controlled conditions.

Another, extremely relevant contribution is the ultrasonic pulse design that allows TOF measurement simultaneously with ultrasonic communication. This ultrasonic pulse is robust to environments with strong multipath when used to measure the TOF and to transmit data. This pulse makes use of OFDM and DBPSK. Additionally, some techniques are suggested, to reach, and sometimes surpass, the most frequently used TOF measurement pulse, the chirp pulse.

By combining this first two contributions the author designed a RF-free ultrasonic location system with a reduced setup time and low maintenance cost, where failure nodes can be easily replaced without the need of a qualified technician. To accomplish that, the proposed location system uses a synchronous network of beacons where a proposed location algorithm will allow each beacon to compute its location using TOF measurements from its neighbors. The beacon location is performed using a modified version of the distributed weighted-multidimensional scaling proposed in [8]. Which is a distributed location algorithm that can deal with a variable number of neighbors and it can improve the accuracy of the location by minimizing a local cost function. Each beacon on the network periodically transmits an ultrasonic message. Thereafter, the mobiles, by collecting a set of this messages, can compute their location using common TDOA algorithm.

The last important contribution is related with the process of synchronizing the beacon's network and obtaining the information of distance between the beacons. The proposed synchronization technique was based in the timing-sync protocol for sensor networks (TPSN) proposed by Saurabh Ganeriwal, et. al, in [9]. This synchronization technique was modified using a three messages exchange mechanism which allows to take advantage of the neighbors synchronization processes. Therefore, by using the proposed approach the distance measurement process is embedded in the network synchronization process. As a result of this, each beacon will obtain the distance information of its neighborhood without exchanging any additional message.

1.3 Organization

This thesis is organized in seven chapters. Chapter 2 presents a brief introduction to indoor location systems and the related work on this field, as well as the state-of-art of indoor locations systems with a special emphasis on ultrasonic indoor location systems. An ultrasonic simulator for indoor environments is presented in Chapter 3. Chapter 4 focuses on the ultrasonic pulse that will allow data communication and distance measurement, comparing it to some commonly used techniques. Chapter 5 presents the proposed location system architecture with all the details about beacons location, clock synchronization and mobile location, as well as an exhaustive discussion with some examples of its performance and failure tolerance. Chapter 6 looks at the overall results, validating the used models by field tests and presents a final example combining all the algorithms proposed for the location system in Chapter 5. Finally, in Chapter 7 the Ph.D. work will be summarized and some proposals for future research directions will be introduced in order to improve the proposed location system.

This page intentionally contains only this sentence.

CHAPTER 2

INDOOR LOCATION SYSTEMS

The GPS is the most popular system for outdoor location achieving an accuracy between 20 to 30 m [10]. Recently, there has been great interest in using the antennas infrastructure of the global system for mobile communications (GSM) to perform outdoor location without the need of any additional hardware besides the mobile phones. The expected accuracy of such systems in urban areas is about 100 m [11, 12]. However, this accuracy is not enough for indoor applications. This chapter presents an introduction to indoor location and some notation that it will be used during the following chapters. This chapter, starts by presenting the indoor location system characterization in Section 2.1. After that, in Section 2.2 the three main technologies used in indoor location system are presented. Which will be detailed in the follow four sections: 2.3, 2.4, 2.5, and 2.6, and compared in Section 2.7. At the end of this chapter the typical approach followed by the ultrasonic location system state-of-art will be presented.

2.1 Location System Characterization

2.1.1 Levels of Location

An indoor LS can be characterized by its level of location which results from the LS accuracy and characteristics. Therefore, an indoor LS can be dived into three main location levels:

Building At building level the LS only needs to provide the information in which part of building the object is (e.g., first floor, north side; entrance; etc.). For that propose the accuracy needs to be around 10 m.

Room At room level the LS needs to provide the exact room of a building where the object is (e.g., bathroom, kitchen, master bedroom, etc). In a such system the accuracy needs to be around 1 m.

Exact At exact level the LS provides the exact position of the object inside a room in reference to the room or the building, normally it is presented as Cartesian coordinates (e.g. (4, 2); (3, 5, 1)). Resulting into an accuracy need of around 10 cm or less.

Figure 2.1 presents a comparison between this three levels of accuracy in a building floor. At the building level the LS gives the information that the object is in the first floor of that particular building. At the room level the LS, apart from the floor, could also specify that the object is in Room 2 of the first floor. In contrast, the exact level provides the exact position in the Room 2 where the object is (7, 9) in this example.

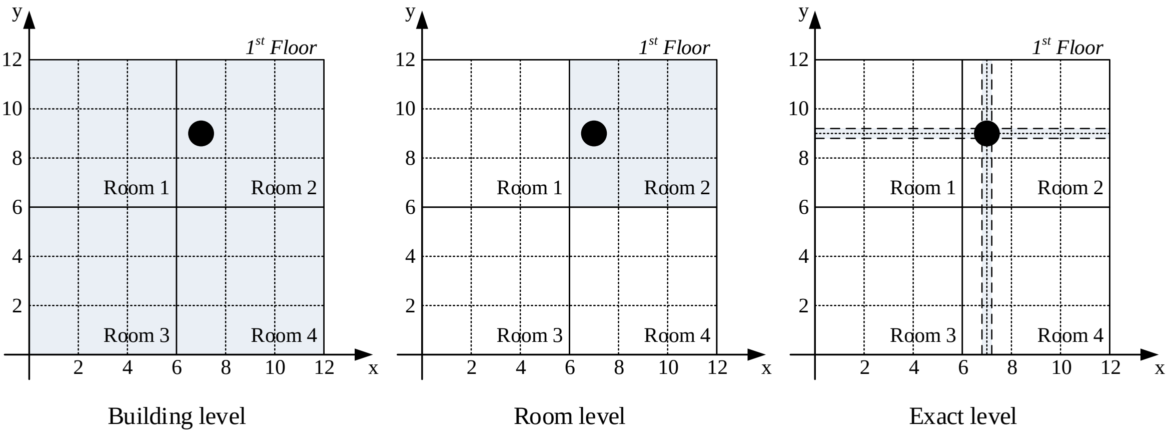


Figure 2.1: Comparison between the three levels of location.

2.1.2 Network Topology and Operation Mode

The indoor LS solutions can be classified by the network topology used and the operation mode. In the network topology the system is characterized by the use of an auxiliary network of reference nodes (designated by beacons). The network of beacons is normally utilized to assure complete field coverage where the location system must operate. Typically, in a LS with a network of beacons the nodes connect only with the beacons and they do not connect to other nodes. However, the system can operate without this network of beacons, this is typically used for mutual location (e.g. robot formation), in this topology the nodes connect to each others in order to perform the location. Figure 2.2 shows the difference between LS with an auxiliary network of beacons and without it.

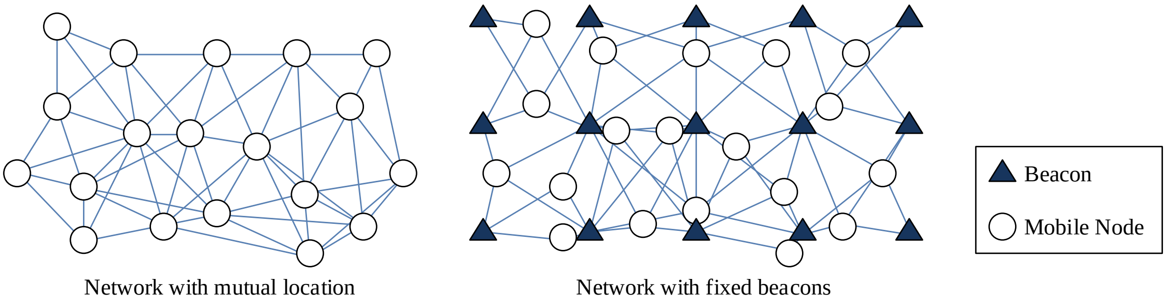


Figure 2.2: The use of an auxiliary network of beacons.

The operation mode can be classified according to where the location algorithm is performed.

In the **centralized mode** the location algorithm is computed in a central node, designed specifically for that propose, the nodes connect to this central node and send to it the relevant information to perform the nodes location, the central node computes all the nodes location and could share the location information with the remaining nodes. The option to share or not the location information depends on the LS purpose and security or privacy policies. In this type of systems the connectivity with the central node is mandatory, restricting the network range or increasing the communication complexity.

In the **distributed mode**, each node runs the location algorithm and it can compute its location based on the information exchanged with its neighbors. This approach typically reduces the connectivity problem to the central node. However, it is more demanding in respect to the nodes processing capabilities.

Figure 2.3 presents the difference between centralized and distributed location algorithms.

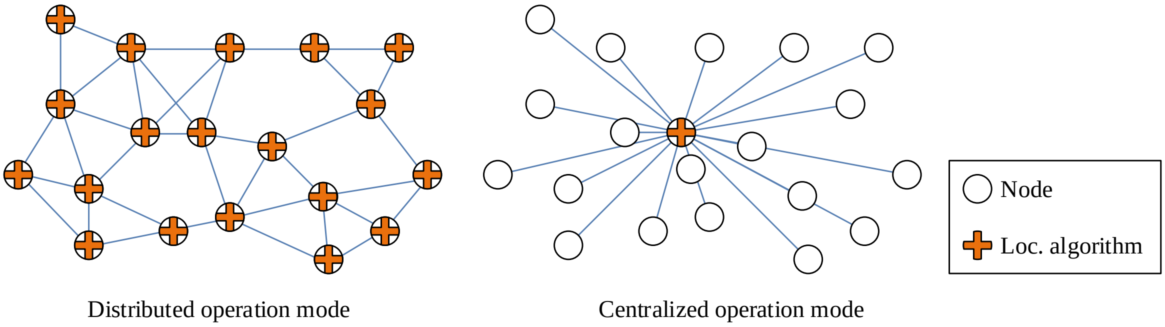


Figure 2.3: The use of a central node running the location algorithm.

2.2 Technologies for Indoor Environments

To achieve the levels of accuracy required in indoor environments, there are three major types of technologies that could be used [4]:

- Infrared based;
- Ultrasounds (US) based;
- RF based.

LS based on infrared radiation usually provides a room level location, they use a network of beacons that receive an infrared unique identifier from a source placed on the object, the system typically operates in the centralized mode. On the other hand, the majority of the LS based on US, provide the exact location of the node and they normally perform the node location by measuring the distance (using TOF) between nodes (in mutual location) and other reference points (with a network of beacons). Moreover, these systems can operate either in centralized or distributed mode. LS based on RF provides a room level location and sometimes the exact location, for that purposed, they usually use the power of the received signal from a network of beacons and compare it to a set of amplitudes, captured previously in several positions. This information is further used to infer about the node location (they typically operate in the centralized mode).

This three main technologies will be detailed and the state-of-art of each one will be presented in the following sections.

2.3 Location Systems based on Infrareds

This section presents two different location systems based on infrared: the **Active Badge** [13] and the **InfraRed Indoor Scout** [14]. The Active Badge is a centralized location system with a network of sensors that detect the location of an infrared emitter tag. The InfraRed Indoor Scout is a centralized location system that uses a stereo camera to detect the location of an infrared emitter tag.

2.3.1 Active Badge

The Active Badge [13] is one of the first indoor location systems, it was introduced in 1992. The Active Badge was designed to locate a tag inside a building, giving a symbolic position (e.g. the room where the tag is). Each tag sends a unique infrared identification (ID) with approximately 100 ms duration every 15 seconds. Therefore, a wired network of infrared sensors placed in the building receives the infrared signal and sends it to a master station, in the network, which processes all the received data and computes the tag symbolic location. The Active Badge is a very simple indoor location system, which provides room accuracy. The most expensive part of the system is the sensor network and it is a solution with some drawbacks: short infrared range, the need for line-of-sight operation and external light interference (e.g., Sun, light bulb). However, the tag's battery has a lifetime of one year, which is reasonable for office applications.

2.3.2 InfraRed Indoor Scout

The InfraRed Indoor Scout [14] uses an infrared emitting tag and a stationary mounted stereo camera to locate the objects. The stereo camera is built with two cheap USB cameras (with 120° lenses and an infrared filter) placed at 20 cm from each other. The stereo camera receives the infrared signal and the host personal computer (PC) measures its angle of arrival (AOA) at the two cameras, therefore, the tag location is computed by triangulation. The InfraRed Indoor Scout is a very cheap indoor location system that uses off-the-shelf components to compute tag location. Moreover the number of tags does not significantly increases the location processing cost. Furthermore the system can cover a room with 15.1×9 m with 16,7 cm of accuracy. Nevertheless, stereo camera calibration is one of the main issues, mainly because the optical cameras are far from ideal.

2.4 Location Systems based on Ultrasounds

This section presents eight different LSs based on US: the **Active Bat** [15], the **Cricket** location system [16, 17], the location system from **F. Rivard and et al.** [18], the **Parrot** [19], the **Dolphin** [20], the **3D-LOCUS** [21], the **M. Hazas and A. Hopper's** system [22] and the **J. Gonzalez and C. Bleakley** location system [23]. These location systems use two modes of operation: centralized and local. In centralized mode (Active Bat, 3D-LOCUS and Hazas' LS) the nodes send an unique ultrasonic pulse ID and a network of ultrasonic sensors, with well known locations, receive this pulse and compute the location of each node. In the local mode there are two setup possibilities: using an auxiliary sensor network (Cricket, Hazas' LS and Gonzalez's) or using a distributed system (Rivard's LS, Parrot and Dolphin). In the mode with an auxiliary network of ultrasonic sensors, each sensor in the network with well known location sends an unique ultrasonic pulse ID and the object receives this pulse and computes its location. In the distributed mode each object sends an unique ultrasonic pulse ID and its location (if it is known) and the other objects receive these pulses and compute their location.

2.4.1 Active Bat

The Active Bat [15] is a three dimensional (3D) location system that uses US and an auxiliary RF channel to perform the 3D location of a tag (10 × 6 × 2 cm). Every 200 ms a radio message is sent using the auxiliary RF channel (at 418 MHz), this message contains the ID (16 bits) of the tag that the system wants to localize. Therefore, the tag with that ID sends a 50 μs ultrasonic pulse at 40 kHz using an hemispherical array of 5 US transducers. A matrix of ultrasonic receivers, mounted in the ceiling at 120 cm from each other, receives the ultrasonic pulse and sends the times of arrival (TOAs) to a central PC. This central PC computes the tag location by multilateration. Results have shown an accuracy of 8 cm in 98% of the cases; moreover, the PC can locate up to 25 tags in each second. On the other hand, the ceiling ultrasonic receiver matrix increases the infrastructure cost, installation time and reduces the system scalability.

2.4.2 Cricket

The Cricket LS [16, 17] allows a tag to localize itself and get its orientation, providing user privacy, by using an ultrasonic and a RF receiver. The Cricket LS uses several ultrasonic and RF emitters (mounted on the ceiling or on the wall). These emitters provide the signal reference that allows the tag to localize and orientate itself. These emitters randomly send a 47 ms RF signal (at 418 MHz) and a 125 μs ultrasonic pulse (at 40 kHz). The RF signal carries the emitter location (3×2 bytes), its ID (8 bytes) and the room ID (8 bytes). The tags receive the RF signal and the ultrasonic pulse from each emitter and compute the distance to them (using the TOF). The Cricket LS uses cheap hardware, each emitter or receiver costs less than 10$ and provides an accuracy of 10 cm. Moreover, the Cricket Location System has a high level of scalability, although, the system consumes some power (about 15 mW per device).

2.4.3 Rivard's Location System

The LS from F. Rivard et al. [18] was designed to be used in multi-robot systems and it gives, to each robot, a relative location in two dimensional (2D). Therefore, this LS outputs the distance between robots and the angle between them. For that purpose, the system uses US signals (25.7 kHz) and an auxiliary RF channel (433.92 MHz) to measure the distance (using TOF) and the angle between robots. The system has one ultrasonic transmitter and two receivers (to compute the AOA). Moreover, each robot has a unique ID and it is synchronized with the other robots, to avoid the interference between the sent ultrasonic pulses. Each robot sends an ultrasonic pulse using round robin scheduling with a time watchdog (to keep the system working if some robot fails) and at the same time it signals the other robots using the RF auxiliary channel. The system presents an absolute average error of 8 mm and 3° with a standard deviations of 8 mm and 3° respectively. However the main bottleneck of the system is its power consumption, 1.73 W.

2.4.4 Parrot

The Parrot system [19] is based on a network of tags (Parrots). Each tag uses four ultrasonic transmitters and receivers (40 kHz) and an auxiliary RF channel for synchronization and data communication. The system randomly sorts the tags in a queue and the first tag in the queue sends an ultrasonic pulse and RF signal. The other tags receive these signals and compute the distance to that tag using the TOF (time difference between RF and the US pulse). Therefore, this tag goes to the end of the queue and the first tag in the queue behaves the same way of the previous one, and so on. Moreover, the RF link between the tags is used also to communicate the measured distances from one tag to the others. As a result of this, this LS eliminates the necessity of a central system to compute the tags location, so each tag computes its location locally from the information given by the other tags. This location system exhibits an accuracy of 2 cm with a standard deviation of 4 cm and it can work up to 15 m.

2.4.5 Dolphin

The Dolphin system [20] is a 3D distributed LS that uses US and a RF auxiliary channel. The Ultrasonic system uses 5 US transducers mounted in a cylindrical object to have an omnidirectional ultrasonic transducer. This system uses a distributed positioning algorithm to perform the multilateration in an interactive way (it is necessary to have some tags that know their exact location). Therefore, in the distributed system there is one master tag, one transmitter tag and several receiver tags, only tags with known position can become a master or a transmitter tag. In every positioning cycle, the master tag sends a RF message to synchronize all the tags (it is assumed that all the tags can receive this message), therefore, when the transmitter tag receives this message it sends an US pulse. Receivers, that are able to receive the ultrasonic pulse, compute the distance to the transmitted tag by measuring the TOF. Therefore, if a tag receives enough distance information, it can compute its location by multilateration. The Dolphin system presents very slow refresh rate of 1 to 2 times/sec, therefore it is only useful to compute the location of static tags. The system has an accuracy of 2 cm, however, it only operates in a range up to 3 m.

2.4.6 3D-LOCUS

The 3D-LOCUS [21] is a general centralized LS that uses sound (5 to 25 kHz) to measure the distances (using TOF) and RF (Bluetooth for data and 433 MHz for synchronization) or wires (BusCAN for data and LVDS for synchronization). The 3D-LOCUS can operate using two different access modes: time division multiple access (TDMA) or code division multiple access (CDMA). In both cases it uses direct-sequence spread spectrum (DSSS) (Golay codes with length of 32, 64 or 128) modulated with binary phase shift keying (BPSK). In the normal configuration, the system has a network of sensors mounted on the ceiling, connected by wires to a central node, and some moving tags, which are interconnected by RF. The system can work in three different configurations: Centralized (the moving tags send an US pulse and the central node compute their location), Privacy Oriented (the ceiling tags send an US pulse and each moving tag computes its location) or Bidirectional (both type of tags send an US pulse). Besides location, the system is also able to compute the sound speed. The system has an accuracy of 4.1 to 5.2 mm in 90% of the cases for TDMA mode and 8.6 to 11 mm for CDMA mode, with a wind flow of 2 m/s the system accuracy changes to 4.9 to 11.5 mm in 90% of the cases for TDMA mode and 7.5 to 13.7 mm for CDMA mode. Therefore, this system can operate not only in indoor environments but also in outdoor environments. However, the system only operates in the range from 1.02 to 4.94 m which limits the system applicability. Moreover, this system needs to have a calibration process to achieve the sub-centimeter accuracy.

2.4.7 Hazas' Location System

The Hazas and Hopper's LS [22] is one of the first systems that uses broadband ultrasonic transducers to perform 3D location. Moreover, it can avoid the auxiliary RF channel for synchronization. The system has two different modes of operation: centralized and private. In the centralized mode a network of receiver tags are placed in static positions with well known locations, and the tags, that the system wants to locate, transmit simultaneously a broadband ultrasonic pulse (a 511 bits Gold code with the rate of 20 kHz modulated BPSK at 50 kHz). The results showed that in this mode the system has an accuracy of 2.1 cm in 95% of the cases. In the private mode the system can adopt two different strategies: synchronous or asynchronous. In the synchronous strategy the system has to be synchronized by an auxiliary RF channel, in the asynchronous strategy it is possible to avoid it. For a tag to perform its location it is necessary to have a network of transmitter tags placed on static positions, with well known location, simultaneously transmitting a broadband ultrasonic pulse. The results showed that in this mode and using the synchronous strategy the system has an accuracy of 5 cm in 95% of the cases. Nevertheless, when the system adopts the asynchronous strategy the accuracy decreases to 25 cm in 95% of the cases. This is an accurate system that can be used in a real location system, it presents a high degree of scalability in both modes without reducing significantly the location update rate, this is manly due to the use of Gold codes (up to 513 transmitters). However, when the system works without the auxiliary RF channel for synchronization, the accuracy is reduced and the applicability in some scenarios could be compromised.

2.4.8 Gonzalez's Location System

Gonzalez and Bleakley introduced in [23] a 3D location and orientation system that uses US and an auxiliary RF channel for tags synchronization. The system uses a set of tags with well known position, these tags send a broadband ultrasonic signal using frequency hopping spread spectrum (FHSS). In order to accommodate the high signal bandwidth a normal piezoelectric US transducer operating at 41 kHz with 2 kHz of bandwidth was used. This transducers were electrically modified to increase its bandwidth (approximately up to 15 kHz). The tags with unknown position, compute their location and orientation using an extension of the MUSIC algorithm and TOF and AOA measurements. Therefore, all the tags are synchronized by a RF signal. To compute the AOA, the tags use an uniform circular array of 8 US sensors based on micro electro-mechanical systems (MEMS) technology that can operate from 10 to 65 kHz. However, the maximum frequency was reduced to 41.4 kHz due to the distance between the sensors (4 mm). The system is very accurate in location and orientation, it presents an accuracy of 1 cm in 95% of the cases and a mean error of 4.5° in orientation. However, it is a prototype system and it is not fully functional, some issues are under investigation by the authors.

2.5 Location Systems based on Radio Frequency

This section presents five different location systems based on RF: **Wideband PLP** [24], **LANDMARC** system [25], **RADAR** system [26], **COMPASS** system [27] and the LS proposed by **Claro and Carvalho** [28]. There are three different used technologies (powerline (PL), radio frequency Identification (RFID) and wireless local area network (WLAN)) but all of them are based on a survey of the RF signal characteristics in several locations. The object location is computed by comparing the received RF signal characteristics to those previously surveyed.

2.5.1 Wideband PLP

The Wideband powerline positioning (PLP) [24] is a LS based on RF fingerprint (the RF fingerprint is the survey of the RF signal characteristics in a specific location), therefore the system can use a map of the RF fingerprint to infer the tag location. This system uses a RF fingerprint map generated by the RF radiation from a powerline system (the electrical facility infrastructure behaves like a RF antenna). For that propose, the system injects in the electrical infrastructure a signal composed by 44 different frequencies (447, 448, 600, 601 and 500 to 20000 kHz in increments of 500 kHz). Therefore, the tag can infer about its location by using the RF fingerprint map surveyed before. The tests were conducted in a floor of a research laboratory that was adapted to imitate a residential environment. Therefore, the RF fingerprint (measures of the amplitude of the 44 frequencies) were taken in 66 locations in a grid of 90 × 90 cm, the accuracy for grid level was 100%. However, the practical tests reveal that the accuracy is reduced with time, 2 months after the RF fingerprint accuracy was reduced to 59%. Therefore, this type of LS needs to be calibrated regularly, which can be impossible in residential environments.

2.5.2 LANDMARC

LANDMARC location system [25] is a 2D location system based on RFID. The system can compute the location of active RFID tags (3-5 years of battery life). For that propose it is necessary to have a set of RFID readers (they have 45 m of maximum range). Each tag has a unique RFID (48 bits at 308 MHz) and sends its ID randomly with a mean period of 7.5 s. Therefore, the RFID readers set their maximum distance range into 8 different values and reports to a central node, which tags are in that particular range. Consequently, the central node can infer about the location of the RFID tag. In order to increase the accuracy of the system, reference RFID tags were placed in known locations and, in the central node, the measures obtained from this tags were compared to the measures obtained from unknown tags. Practical results showed that the location system can locate a tag with accuracy of 2 m in a room with 4 × 10 m with 4 RFID receivers, 16 RFID reference tags and 8 RFID tags with unknown location. However, the system takes 1 min to obtain the information of the 8 distance ranges, which reduces its applicability for moving tags. The RFID readers are very expensive which reduces the system scalability; still, the system can operate with 500 RFID tags, which are very cheap, small and light.

2.5.3 RADAR

The RADAR location system [26] is a 2D location and tracking system that uses the WLAN in the 2.4 GHz free band (WaveLAN) similar to the wireless fidelity (WiFi) (802.11b). The system can use the existing WLAN infrastructure (the access points) to perform the localization. These access points, with well known locations, broadcast every 4 s, 6 byte user datagram protocol (UDP) packet. Therefore, the tags compute its location by measuring the received signal strength from the access points and by applying the nearest neighbor(s) in signal space technique. This means that the system is going to select the location of the tag by comparing the received signal strength to a set of a *priori* signals strengths collected in known locations. These a *priori* signal strengths can be collected by two different methods: empirical method and radio propagation model method. In the empirical method the set of signal strengths are collected before in the real field. In the radio propagation method, the set of signal strengths are computed using a propagation model (wall attenuation factor) of the local. The RADAR location system produces a result with an accuracy less than 2.94 m in 50% of the cases and less than 4.69 m in 75% of the cases, using the empirical method in 70 places on a floor with 44×23 m and 3 access points. Moreover, using the radio propagation model method, the location system results showed an accuracy less than 4.3 m in 50% of the cases in the same floor. Therefore, the accuracy of the location system is enough to track objects with room accuracy. Moreover, the system is cheap and scalable, it uses off-the-shelf components for the infrastructure. However, the tags need to have WLAN hardware, which is a limitation to portability manly because WLAN hardware significantly increases the weight and the power consumption of the tags.

2.5.4 COMPASS

The COMPASS LS [27] is a 2D LS that uses the installed WLAN infrastructure, 802.11 based, and a digital compass to perform the tags location. The COMPASS location is based on the WLAN fingerprint of several known positions and orientations. Therefore, when a tag wants to determine its location it measures the received signal strength of several WLAN access points, and knowing its orientation (given by the digital compass) it is able to compute its location by a probabilistic positioning algorithm, which compares the collected data and the fingerprint data in the same direction. Practical results showed an accuracy less than 2 m in 70% of the cases (which is enough to track objects with room accuracy) in a floor with 15 × 36 m and 9 WLAN access points. The test results showed that the introduction of the digital compass turns the location system more accurate and can be useful in dynamic tags where the tag orientation can predict its movements. Moreover, the system is cheap and scalable and uses off-the-shelf components and the already installed infrastructure. However, the tags need to have WLAN hardware, which is a limitation to portability manly because WLAN hardware significantly increases the weight and the power consumption of the tags.

2.5.5 Claro's Location System

The P. Claro and N. Carvalho's LS [28] is a 2D LS based on the fingerprint of the installed WLAN infrastructure (802.11 based). The system uses a set of fingerprints (the received signal strength from a set of access points) in several known positions to train an artificial neural network in a central node. After training the artificial neural network the location system is able to compute the tags location. Therefore, the central node requests the tags to send the collected signal strengths. In order to compute the tag location the central node normalizes the received data and uses the trained artificial neural network to obtain the tag location. Practical results showed an accuracy less than 1 m for 50% of the cases and less than 3 m for 90% of the cases in a Shopping Center with several access points. This is an accurate and cheap location system with a high level of scalability; it uses the indoor access point from the WLANs and its uses off-the-shelf components (as the tests in a Shopping Center proved). However the tags need to have WLAN hardware, which is a limitation to portability manly because WLAN hardware significantly increases the weight and the power consumption of the tags.

2.6 Location Systems based on other Technologies

In this section one hybrid location system, the LuxTrace [29] is presented. It is based on two different technologies: RFID and light. It is based on a survey of the light conditions and RFID signal characteristics in several locations. Therefore, the object location is computed by comparing the received light conditions and RFID signal characteristics surveyed before.

2.6.1 LuxTrace

The LuxTrace [29] is a 2D location system that uses the light conditions fingerprint inside the buildings and RFID to perform the tags location. The tags have a solar cell, which measures the light intensity and provides energy, and a RFID reader. Therefore, the RFID tags, placed in the floor of the building, give the tag a reference to their position, as a result of combining these references with the light measurements provided by the solar cell it is possible for the tag to locate itself. Practical tests have shown that the system can produce an accuracy less than 28 cm in 80% of the measurements. However, these tests were a proof of concept, they were only done in controlled conditions and they only gave one dimensional (1D) location. The system has a high level of scalability and it is cheap, it uses the indoor light conditions and RFID tags. However it suffers from light interference as the interference from sunlight or malfunctioning lamps.

2.7 Comparing Location Systems

Table 2.1 presents a comparison between the location system mentioned above. For that purpose, it presents the used technology, the maximum system accuracy, the system scalability, the operation mode and the system cost. In the technology column it is presented the main type of technology used. The maximum accuracy value is presented as a (p%), which mean that in p percent of the cases the accuracy is less than a, or it can be presented as a ($\pm s$), where a is the accuracy and s is the standard deviation. The system scalability is divided in three degrees *None*, *Some* and *High*. The operation mode can be *Central*, if the node location is computed in a central computer or *Local*, if each node is able to compute its location. The system cost can be *Cheap* or *Expensive*.

As one can see from Table 2.1 to achieve the necessary accuracy level in 3D for indoor location, the ultrasounds technology is the best suited one. It gives the user an acceptable indoor accuracy, in 3D, that can be less than 1 cm in some cases [23, 21]. Moreover the RF based systems require the profiling of the entire location scenario to get a RF fingerprint and presents an accuracy from 1 to 5 meters approximately [24, 26]. Although, it does not need a dedicated infrastructure and it is cheaper. However, its performance depends of the installed infrastructure and on interference.

2.8 Typical Approach in Ultrasonic LS

The state of art survey presented previously shows that ultrasounds are a viable alternative to radio frequency or infrared-based location systems when a level of accuracy below one meter is required. Unfortunately, almost all of the known ultrasonic location systems use an auxiliary RF channel or a wired connection for measuring the propagation delay from the source to the receiver. As presented in Figure 2.4 beacons, or in some situations other mobiles, send an US pulse at the same time as a RF pulse. The mobile node after received both pulses measures the TOF by the arrival time difference of both pulses.

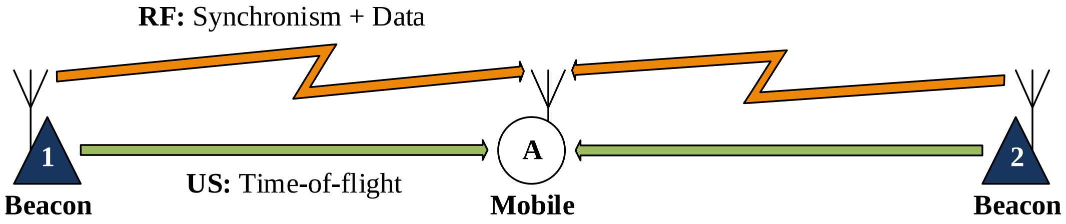


Figure 2.4: Typical approach used in the ultrasonic location systems.

The principal exception to this approach is the M. Hazas and A. Hopper's system [22]. Where, in the asynchronous mode, it can perform the object location without using the auxiliary RF channel and using TDOA as show in Figure 2.5.

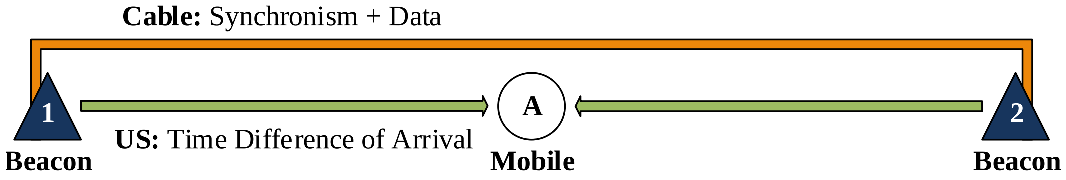

Cable: Synchronism + Data
1
A
2
US: Time Difference of Arrival
Beacon
Mobile
Beacon

Figure 2.5: Approach used by the Hazas and A. Hopper's system to avoid the RF channel.

Nevertheless, this asynchronous mode presents an accuracy less than 25 cm in 95% of the cases. Additionally, each node must know the exact position of each beacon and the beacons transmission time intervals. Moreover, all beacons are connected to a central PC. This results in a very low versatility and high complexity location system from the mobile point of view. Each time that the beacons network changes, even by beacon failure or by room topology changing, the information in each node must be updated.

Although the auxiliary RF channel is only used to allow very simple clock synchronization and delay measurement solutions, giving away two important advantages that US-based systems bear in reference to RF-based ones: the immunity to RF interference, the ability to safely operate in the presence of critical electronic instrumentation such as medical or life-support systems, and most important, it cannot be used in environments where the RF propagation is impossible, such as, underwater environments.

As presented in the introduction chapter the LS proposed in this work is a distributed indoor location system based exclusively on ultrasounds that can provide the node location, avoiding the use of the auxiliary RF channel or cables between beacons. Moreover, the mobile nodes do not need to have any *a priori* information about the network of beacons. They are able to perform their location using only the information provided by the network of beacons in the sent ultrasonic pulse, as one can see in Figure 2.6.

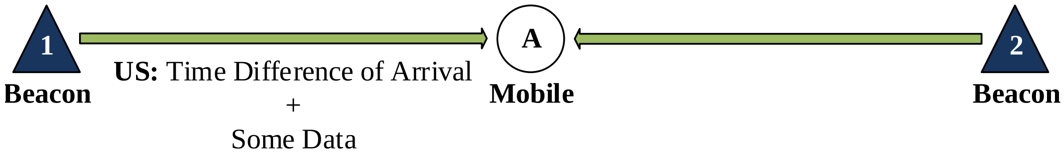
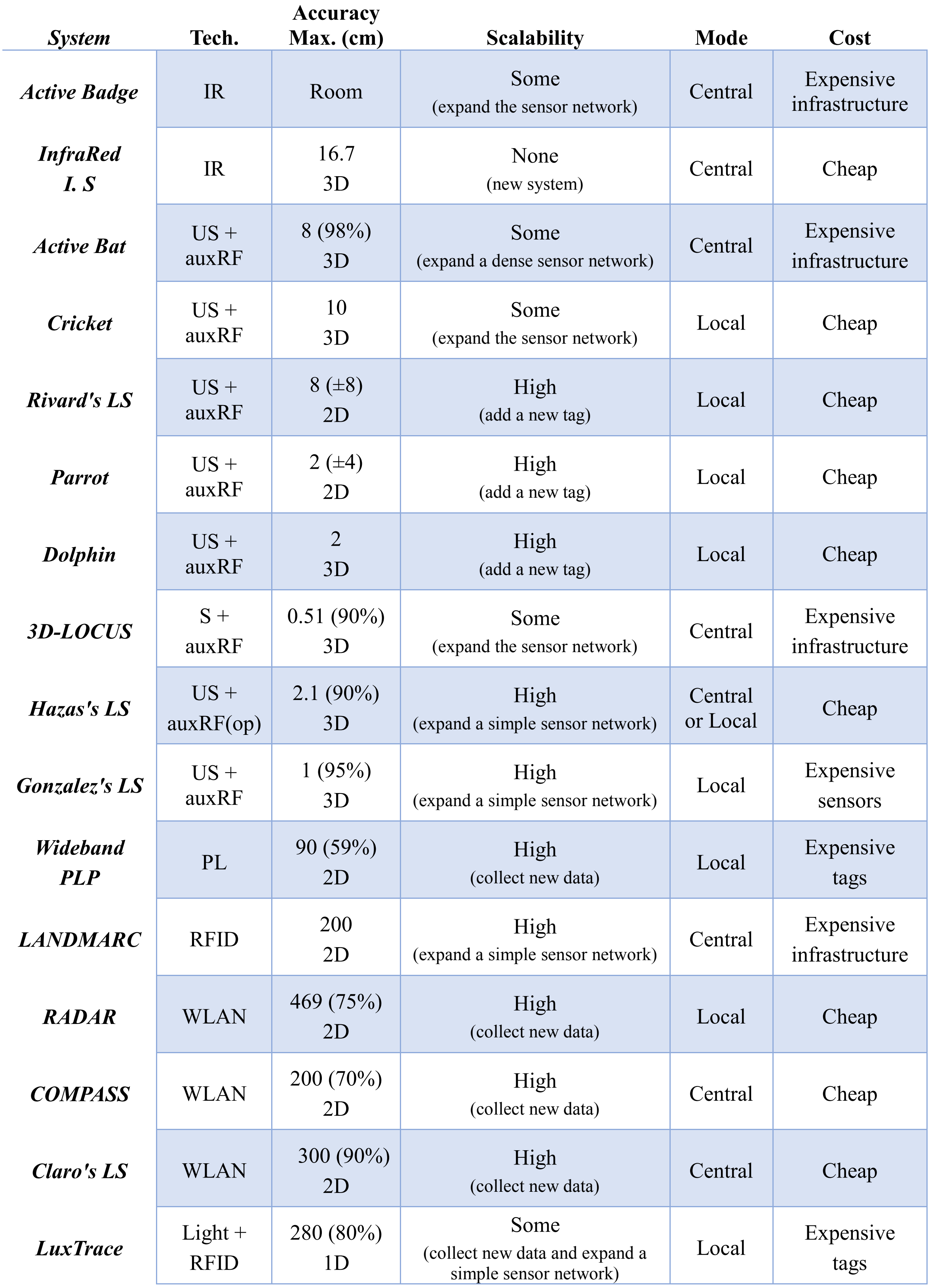


Figure 2.6: Approach followed in the proposed location system.

All the details about the proposed location system will be found in Chapter 5, where all the involved algorithms will also be described.

At the end of this chapter, it is important to mention the work of Carlos De Marziani and colleagues [30], published during the last stages of this thesis work, who proposed a way to measure distances using only ultrasounds. Carlos De Marziani et al. proposed in [30] a way to measure the distance between nodes using the round-trip TOF of acoustic signals. Their idea can be used directly to implement a location system with mutual location (without fixed beacons) obtaining the relative position. However, they do not implement any data transmission mechanism and therefore it is impossible to obtain an absolute position without having fixed nodes whose position is known by all the nodes. Moreover, the distance measurement is based on the round-trip TOF which means that for each distance measurement the system takes at least the double of the time, forcing nodes to be almost stationary for more time than the minimum necessary. Furthermore, for a node to obtain the distance to its neighbors it must ask for it, resulting in the loss of privacy.



System	**Tech.**	**Accuracy** **Max. (cm)**	**Scalability**	**Mode**	**Cost**
Active Badge	IR	Room	Some (expand the sensor network)	Central	Expensive infrastructure
InfraRed I. S	IR	16.7 3D	None (new system)	Central	Cheap
Active Bat	US + auxRF	8 (98%) 3D	Some (expand a dense sensor network)	Central	Expensive infrastructure
Cricket	US + auxRF	10 3D	Some (expand the sensor network)	Local	Cheap
Rivard's LS	US + auxRF	8 (±8) 2D	High (add a new tag)	Local	Cheap
Parrot	US + auxRF	2 (±4) 2D	High (add a new tag)	Local	Cheap
Dolphin	US + auxRF	2 3D	High (add a new tag)	Local	Cheap
3D-LOCUS	S + auxRF	0.51 (90%) 3D	Some (expand the sensor network)	Central	Expensive infrastructure
Hazas's LS	US + auxRF(op)	2.1 (90%) 3D	High (expand a simple sensor network)	Central or Local	Cheap
Gonzalez's LS	US + auxRF	1 (95%) 3D	High (expand a simple sensor network)	Local	Expensive sensors
Wideband PLP	PL	90 (59%) 2D	High (collect new data)	Local	Expensive tags
LANDMARC	RFID	200 2D	High (expand a simple sensor network)	Central	Expensive infrastructure
RADAR	WLAN	469 (75%) 2D	High (collect new data)	Local	Cheap
COMPASS	WLAN	200 (70%) 2D	High (collect new data)	Central	Cheap
Claro's LS	WLAN	300 (90%) 2D	High (collect new data)	Central	Cheap
LuxTrace	Light + RFID	280 (80%) 1D	Some (collect new data and expand a simple sensor network)	Local	Expensive tags

Table 2.1: Comparison of some indoor location systems. (IR - infrared; US - ultrasounds; S - sound; auxRF - auxiliary radio frequency channel for communications; PL - powerline; WLAN - wireless LAN).

This page intentionally contains only this sentence.

CHAPTER 3

ULTRASONIC ROOM SIMULATOR

The ultrasonic room simulator is a valuable tool to study and compare the performance of different location algorithms and acoustic pulses in a controlled environment. By using a simulator it is very easy to change the parameters that affect the response of the acoustic system and to test extreme situations that are not easy to replicate in the real field. Moreover, it is much faster to use a simulator than to setup and carry out real tests under the same circumstances in order to test different algorithms and ultrasonic signals.

Commercially available acoustic simulators, like ODEON [31], are very expensive and since they are not open source, changes are not possible. However, there are examples of freely available simulators [32, 33, 34] but none of them take into account the Doppler effect which has 10000 more impact in US communication than in RF, as it will be seen in Section 3.7.

Indoor ultrasonic acoustic propagation presents problems similar to those experienced in electromagnetic waves propagation, such as multiple reflections, Doppler effect and propagation attenuation [34, 35]. Due to these problems, a good model for the ultrasonic channel is needed in order to properly design ultrasonic communication or location system. The proposed simulator will be based on changing and simplifying some techniques that were used in the acoustics and radio-frequency fields. The approach presented here will allow to produce a very fast and easy to use simulator to evaluate and compare different ultrasonic location algorithms and ultrasonic communication pulses.

Section 3.1 will describe the overall architecture that it will be used in the simulator where each part will be modeled by an impulse response. Thereafter, the process to obtain the impulse response for the transmitter and the receiver will be presented in Section 3.2 and for the sound propagation in Section 3.3. Afterwords, the direct path case will be presented and discussed in Section 3.4 and then it will be extended to the multiple reflection case in Section 3.5. Section 3.6 will discuss the simulator implementation and some results obtained in each process. And Section 3.7 will present the process of using the simulator to implement the Doppler effect and two simple examples to demonstrate the impact of considering the Doppler effect in ultrasonic signals. At the end of this chapter some possible future work that may be carried out to increase the simulator performance and applicability will be described.

3.1 Overall Architecture

The overall ultrasonic room simulator architecture is schematically presented in Figure 3.1. Where each transmitter block represents an ultrasonic transmitter somewhere in the room. This transmitter receives an electric signal $x_m(n)$ and transforms it into an acoustic wave that travels through the Room Channel. Moreover, this Room Channel represents a room where the acoustic waves can propagate and reflect in the walls before arriving at the receiver position. Therefore, the receiver block combines each received wave into a signal $y(n)$.

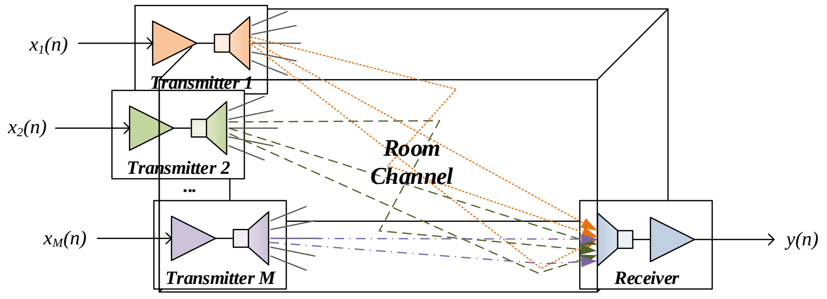


Figure 3.1: The overall ultrasonic room simulator architecture.

The presented architecture can be represented by the block diagram presented in Figure 3.2. Where $h_m(n)$ represents the impulse response (IR) that must be applied to the signal $x_m(n)$ before reaching the receiver. This approach brings the first two simplifications to the models to be used in the simulator:

1. all signals are digital and therefore discrete (time and value);
2. the transmitter, the receiver and the room effects are considered linear and time invariant, and therefore can be represented by an IR.

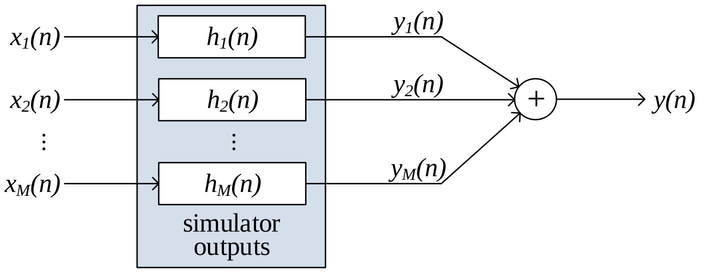


Figure 3.2: Block diagram of the simulator.

The output of the simulator are the IR for each individual source, $h_m(n)$. The user may compute a set of IRs at different receiver's positions and compare the effect of each source and input signal $x_m(n)$ in the received signal $y(n)$.

As can be seen in Figure 3.1, each $h_m(n)$ can be divided in three main IRs which represent the effect of the transmitter, $t_m(n)$, the receiver, $r(n)$, and the sound propagation, $p(n)$, see Figure 3.3.

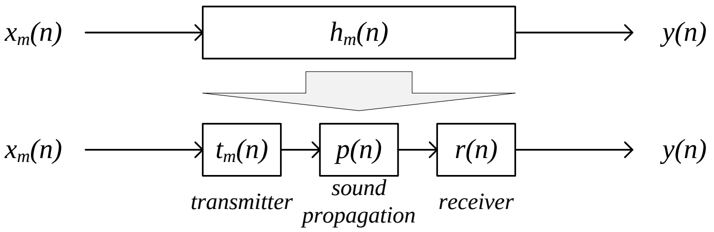


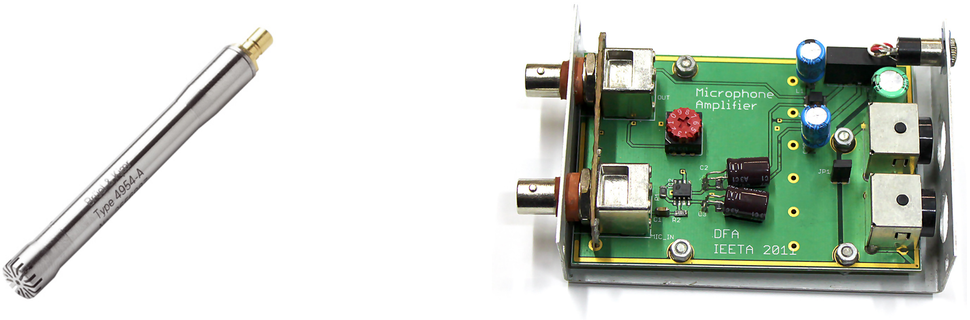
Figure 3.3: The division of $h_m(n)$ in three main IRs.

The transmitter IR, $t_m(n)$, models the effect of the transmission chain (sound card; cables; amplifier; speaker; etc.), in other words, it models all the path from the digital world up to the acoustic wave generated by the speaker. In contrast, the receiver IR, $r(n)$, models the effect of the reception chain (microphone; amplifier; cables; sound card; etc.), in other words, it simulates the path from the received acoustic wave by the microphone up to the digital world. Finally, the sound propagation IR, $p(n)$, models the acoustic wave propagation in a specific room (attenuation; distance delay; etc).

3.2 Transmitter/Receiver Impulse Response

In order to obtain the transmitter and the receiver IR two approaches can be employed. If the hardware manufacturer gives the IR (or frequency response) of their components then it will be possible to obtain the IR for all the chain. If not, measuring the IR of the system with calibrated hardware could be the only way. Note that, for a complete hardware description (e.g.: speaker and microphone) IR measures for each inclination and elevation angle must be obtained. However, this needs to be carried out just once because the measured IRs are considered to be independent of the room conditions. Both IRs must be provided to the simulator, a transmitter IR for a given transmitter output direction and a receiver IR for a given receiver input direction.

Thereafter a simple example of a completed reception chain will be modeled and the IR measured. For the reception chain were used a Brüel & Kjær 4954-A Microphone[1] and a microphone amplifier projected by the author, see appendix A, these two elements are depicted in Figure 3.4. The frequency response of the cables and the measurement systems are considered ideal for the typical ultrasonic frequencies (up to 100 kHz), therefore, they were not taken into account.



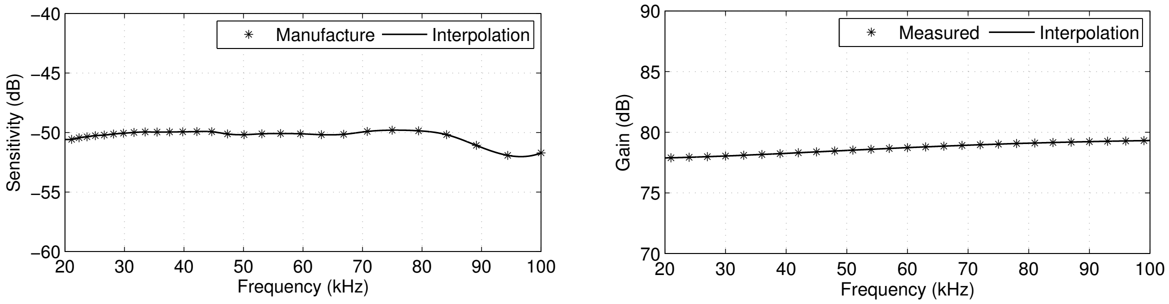
(a) Brüel & Kjær 4954-A microphone.

(b) Microphone amplifier

Figure 3.4: The two main components used in the Reception Chain.

The microphone's manufacturer, provides the microphone frequency response for several directions. However, in the example presented here only the case of signals that arrive from a single direction (that pointed out by the microphone) are considered. Nevertheless, the same procedure must be carried out for each direction. The frequency response of the microphone in the direction where it is pointing out is presented in Figure 3.5a. On the other hand, the amplifier's frequency response was measured with the Audio Precision Portable One Plus Audio Measurement System and its presented in Figure 3.5b.

[1] This microphone was chosen because it is a calibrated microphone with a well known frequency response that operates in the ultrasounds range (up to 100 kHz).

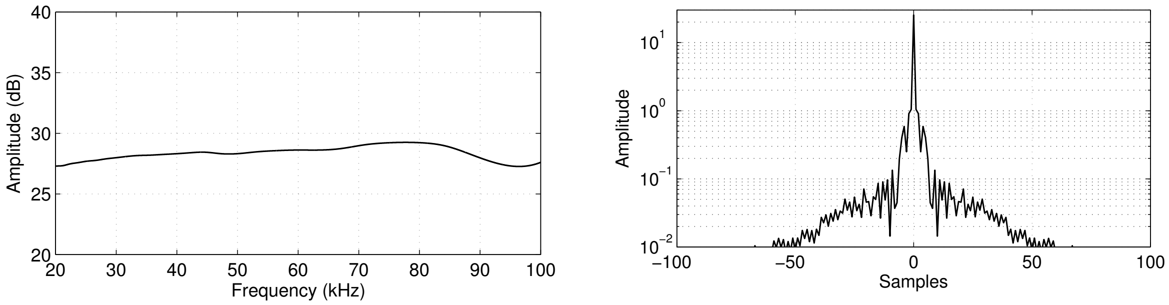


(a) Frequency response of the microphone.

(b) Frequency response of the microphone amplifier.

Figure 3.5: Frequency response of the microphone Brüel & Kjær 4954-A (at 0º) and the microphone amplifier.

By combining these two frequency responses it is possible to obtain the frequency response of the receiver chain as shown in Figure 3.6a and consequently the IR as shown in Figure 3.6b.



(a) Frequency response of the reception chain.

(b) Impulse response of the reception chain.

Figure 3.6: Frequency response and impulse response of the reception chain (microphone combined with microphone amplifier).

3.3 Sound Propagation Impulse Response

This is the most crucial element of the simulator. However, it is not possible to measure the IR for all the ambient conditions and for all possible source/receiver distances, being impractical to model the sound propagation by applying the same approach used for the transmitter and the receiver. Therefore, two different approach based on sound propagation models can be used: a statistically based or a physically based [36, 31, 37]. The statistically based model, commonly used in RF, is a model that can be applied to benchmark different systems in some common situations that can be statistically defined by setting some parameters [37]. However, it does not model the behavior of a system in a particular room or in a particular situation as the physic based model does. The proposed US simulator, uses a physic based model in spite of being a more time consumption model for both computer and user. For that reason, wherever possible, some approximations and simplifications were made in the sound propagation model, reducing the simulation time and the user intervention without compromising the closeness to reality. On that account, the simulator presents the following characteristics and specifications, that will be detailed later:

1. The ultrasonic wave propagation speed is only a function of the room temperature;
2. The room ambient conditions (temperature, pressure and humidity) can be set but after that they are considered to be constant over time and space;
3. The wave propagation attenuates due dispersion and energy absorption and it is considered to be a function of the signal frequency;
4. The wind effect and the gradients of temperature were not taken into account.

3.3.1 Ultrasonic Wave Speed

The ultrasonic wave speed in air, c, depends on temperature, pressure and density of the air [38]. In [38] it can be found a simple expression to accurately compute the ultrasonic wave speed for a perfect gas:

$$c = \sqrt{\gamma \frac{P}{\rho}}, \tag{3.1}$$

where, γ is the ratio of the specific heats (1.402 for air), P the acoustic air pressure and ρ the medium density for a specific temperature (1.2922 kg/m^3 at 0º C and 1.2041 kg/m^3 at 20º C). Therefore, the speed of an ultrasonic wave that travels in air at 1 atm (1.01325×10^5 Pa) of pressure and 0º C is given by:

$$c_0 = \sqrt{1.402 \times \frac{101325}{1.2922}} = 331.5641 \text{ m/s}, \tag{3.2}$$

and at 20º C:

$$c_{20} = \sqrt{1.402 \times \frac{101325}{1.2041}} = 343.4797 \text{ m/s}. \tag{3.3}$$

On the other hand, the ultrasonic wave speed in air can be written as a function of the absolute temperature, T_K, [38]:

$$c = \sqrt{\gamma \frac{\mathcal{R} T_K}{M}}, \tag{3.4}$$

where, $\mathcal{R}$ is the universal gas constant and M the mass of gas. Therefore the ultrasonic wave speed in air is proportional to the square root of the absolute temperature:

$$c \propto \sqrt{T_K}. \tag{3.5}$$

In addition, at typical room conditions, the ultrasonic wave speed in air is almost constant with pressure variations [38]. Using the result of Equation 3.2 the ultrasonic wave speed in air can be written only as a function of the air temperature:

$$c = c_0 \sqrt{\frac{T_K}{273.15}} = c_0 \sqrt{\frac{T}{273.15} + 1} = 331.5641 \sqrt{\frac{T}{273.15} + 1} \tag{3.6}$$

where T is the air temperature in °C. Figure 3.7 presents the variation of the ultrasonic wave speed in air for typical room temperature range.

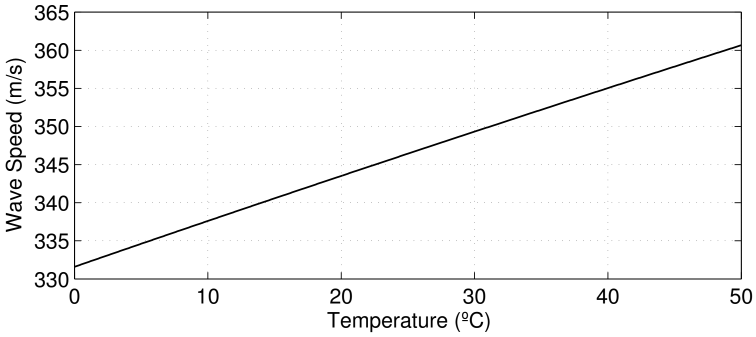


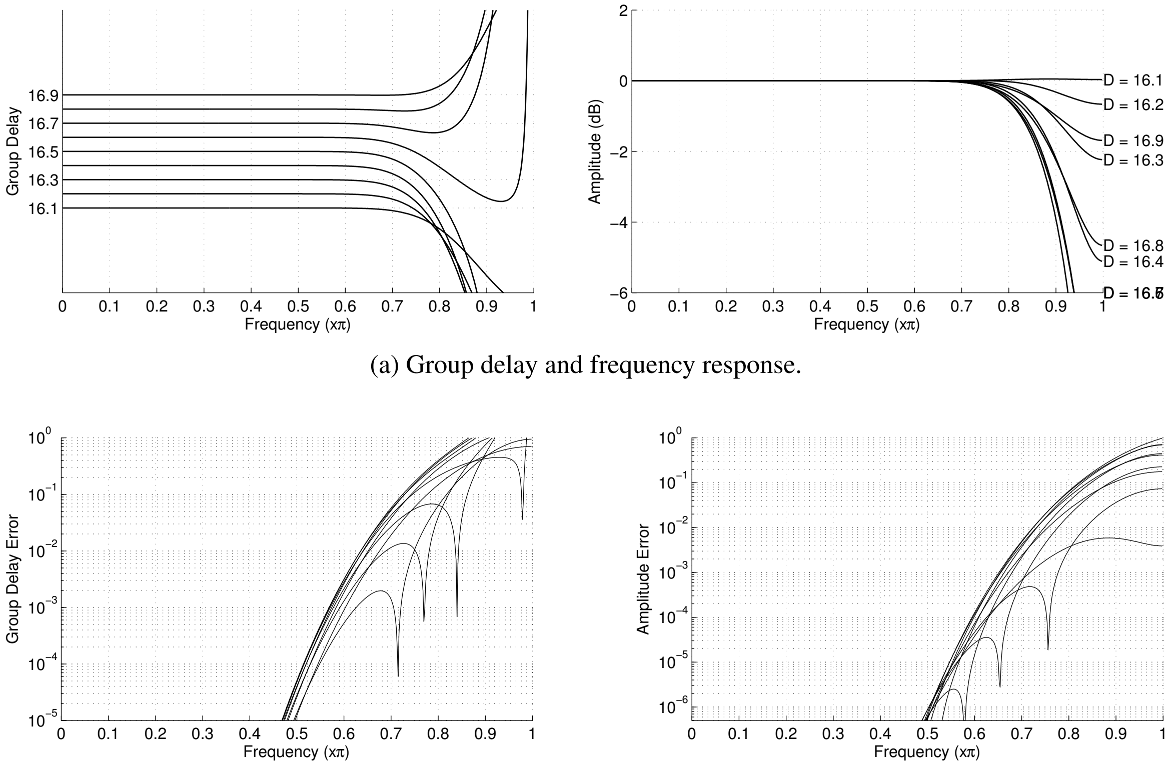
Figure 3.7: Ultrasonic wave speed in air for different room temperatures.

This speed implies that the ultrasonic wave takes some time τ to travel from the source up to the receiver according to their distance, d. Due to the need to accurately simulate this delay, the input signal ($x_m(n)$) may need to be delayed by a non integer value in order to model the real interactions between the received waves. For that reason fractional delay filters must be used for that purpose. This type of filters have many applications in signal processing and communication systems, such as, signal interpolation and changing the signal sampling frequency [39]. Taking into account the simulator properties and constraints, Lagrange fractional delay filter were chosen, see Appendix B. The choice of this particular type of filters relies on the fact that they produce an IR, $d(n)$, that can be easily integrated with the other IRs of the simulator. Furthermore, they were also chosen because their filter coefficients can be readily updated by a close-form equation [39] given by:

$$d(n) = \prod_{\substack{k=0 \\ k\neq n}}^{N} \frac{D-k}{n-k} \tag{3.7}$$

where D is a real number that represents the filter delay. Due to the filter nature the IR is centered around D, therefore, the integer part of D must be near to $N/2$ in order to reduce the causality problem.

Figure 3.8a presents Lagrange fractional delay filters behavior with 32 coefficients. Moreover, in Figure 3.8b it is presented the error associated with the ideal delay. From these figures it can be observed that for a frequency less than $\pi/2$ it presents a group delay error less than 10^{-4} which, for a sampling frequency of 80 kHz, represents 0.43 μm in distance error and less than 0.003% of wave length error for a 20 kHz wave. For frequencies above $\pi/2$ both errors increase considerably therefore it can be considered that a Lagrange fractional delay filter with 32 coefficients only operates properly for frequencies up to a fourth of the sampling frequency which can be pointed out as the main Lagrange fractional delay filter's disadvantage. For that reason, the simulator must use a sampling rate at least four times higher than the maximum frequency of the signal in order to reduce the delay and amplitude errors.



(a) Group delay and frequency response.

(b) Group delay and frequency response errors.

Figure 3.8: Lagrange fractional delay filter with 32 coefficients.

3.3.2 Ultrasonic Wave Attenuation

A reasonable approximation for the ultrasonic wave propagation is to consider it as a spherical wave [38] as shown in Figure 3.9, where the ultrasonic wave propagation from an ideal ultrasonic source point at distance d_0 and d is presented.

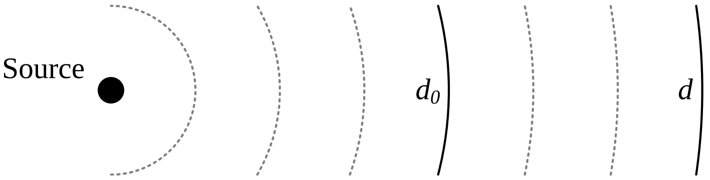


Figure 3.9: Ultrasonic wave propagation.

Considering this approximation, the ultrasonic wave power per unit area is the same for any two points that are at the same distance from the source. Therefore, the acoustic wave intensity I, at distance d from a source with power W is given by:

$$I(d) = \frac{W}{4\pi d^2} \qquad \text{W/m}^2. \tag{3.8}$$

Moreover, the acoustic wave intensity is related with the acoustic air pressure amplitude by [38]:

$$I = \frac{P^2}{2\rho c}. \tag{3.9}$$

From (3.8) and (3.9), the acoustic air pressure amplitude at distance d is:

$$P(d) = \sqrt{\frac{2\rho c W}{4\pi d^2}} = \frac{1}{d} \times \sqrt{\frac{2\rho c W}{4\pi}}. \tag{3.10}$$

On the other hand, the majority of the ultrasonic transducers manufacturers do not give the source power but the acoustic air pressure amplitude, P_0, at a certain known distance d_0, therefore, the acoustic air pressure amplitude at distance d relates with P_0 as:

$$P(d) = \frac{d_0}{d} \times P(d_0) = \frac{d_0}{d} \times P_0. \tag{3.11}$$

This equation only takes into account the wave attenuation due to dispersion. However, as observed in Figure 3.10, the wave attenuation increases significantly with frequency, especially for distances above 3 m.

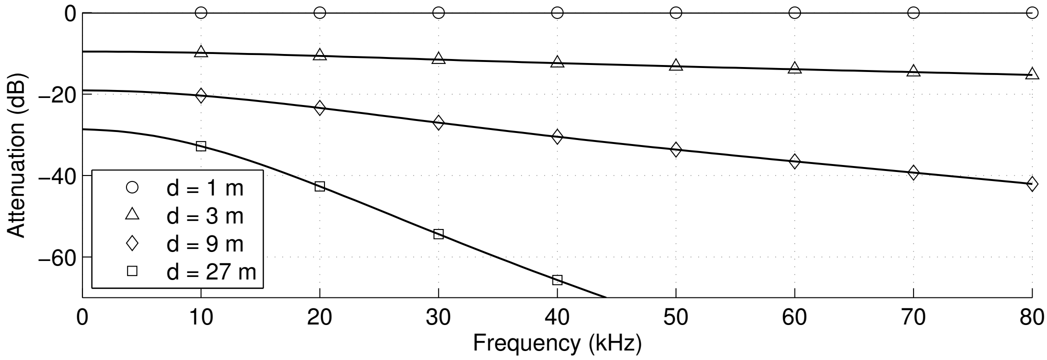


Figure 3.10: Wave attenuation variation over frequency for different distances at typical room conditions (reference distance of 1 m).

The propagation losses are manly due to thermal conductivity, viscosity, diffusion and molecular relaxation [40], under this conditions the acoustic air pressure amplitude taking into account only the propagation losses at distance d from the source can be expressed as [40]:

$$P(d) = P_0 \times 10^{-\frac{\alpha(d-d_0)}{20}} \tag{3.12}$$

where, α is the absorption coefficient in air and it can written as [38]:

$$\alpha = 20\log(e)\left\{\frac{2\pi^2 f^2}{\rho c^3}\left[\frac{4}{3}\nu + (\gamma-1)\frac{k}{C_p}\right] + \sum_{i}^{N_C}\frac{\pi D_i}{c}\left[\frac{f^2/f_i}{1+(f^2/f_i^2)}\right]\right\} \tag{3.13}$$

where f is the sound frequency, f_i is the ith compound relaxation frequency of the air mixture, N_C is the number of the air compounds, ν is the viscosity, k is the thermal conductivity and C_p is the thermal capacity at constant pressure. However the 9613-1 ISO norm [41] give us a much easy way to compute the absorption coefficient in air using only parameters easy to obtain, such as, the air temperature, the acoustic air pressure, frequency and relative humidity.

Figure 3.11 shows the absorption coefficient in air as a function of the temperature, frequency, humidity and pressure using the 9613-1 ISO norm. Moreover it was considered as default values

20°C of temperature, 50% of relative humidity and 1 atm of pressure and a 40 kHz wave.

Figure 3.11a presents the absorption coefficient in air as a function of the acoustic air pressure for different wave frequencies. The acoustic air pressure has only a little influence for frequencies above 60 kHz for the remaining ones α can be considered constant.

Figure 3.11b presents the absorption coefficient in air as a function of the acoustic air pressure for different ambient temperatures. The acoustic air pressure influence increases for very low ambient temperatures. However, for common room temperature the absorption coefficient in air changes only 0.05 dB/m for extreme pressure variation[2]. As a result of this, it can be pointed out that the absorption coefficient in air is not sensible to acoustic air pressure variations.

Figure 3.11c presents the absorption coefficient in air as a function of the relative humidity for different wave frequencies. The relative humidity has a strong impact in the absorption coefficient in air especially for frequencies above 40 kHz where variations greater than 60% could occur for typical room conditions[3] could occur. For example, the absorption coefficient in air for a 60 kHz wave can change from around 1.5 dB/m at 30% of relative humidity to around 2.1 dB/m at 60% of relative humidity. However, for low ultrasonic frequencies, such as a 30 kHz wave, the absorption coefficient in air variation is less that 0.1 dB/m for the same relativity humidity interval.

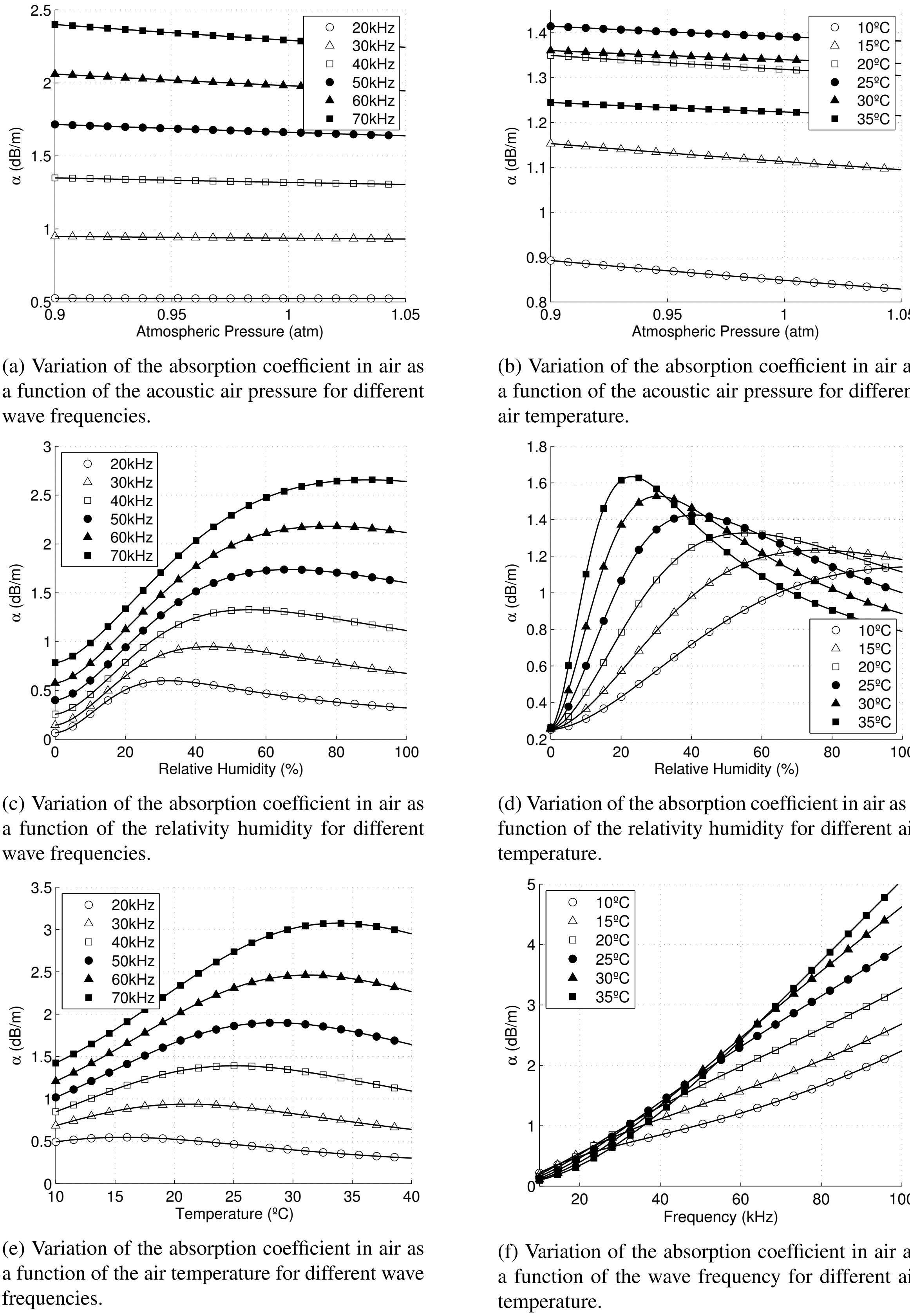
Figure 3.11d presents the absorption coefficient in air as a function of the relative humidity for different ambient temperatures. The absorption coefficient in air, for typical room temperature, increases faster and reaches its maximum around 40% of relative humidity, after this value the absorption coefficient in air starts to decrease slowly. Nevertheless, the maximum absorption coefficient in air depends on the ambient temperatures, for low temperatures the maximum absorption coefficient in air is lower than for high temperatures and it is reached at a very high relative humidity.

Figure 3.11e presents the absorption coefficient in air as a function of the ambient temperature for different wave frequencies. The ambient temperature has a strong impact in α especially for frequencies above 40 kHz similar to the relative humidity. For typically indoor temperatures (up to 25º C) the absorption coefficient in air increases with the temperature for high frequencies, whereas, for low frequencies α decreases with the temperature.

Figure 3.11f presents the absorption coefficient in air as a function of the wave frequency for different ambient temperatures. The frequency presents a strong impact in α, increasing around 0.5 dB/m for each 10 kHz. Starting at around 0.2 dB/m for 20 kHz and ending around 3.5 dB/m for 100 kHz.

[2] The highest acoustic air pressure recorded on United States was 1.05 atm and the lowest was 0.88 atm [42]

[3] It is considered, in typical room conditions, that the relativity humidity is between 30% and 60%.



(a) Variation of the absorption coefficient in air as a function of the acoustic air pressure for different wave frequencies.

(b) Variation of the absorption coefficient in air as a function of the acoustic air pressure for different air temperature.

(c) Variation of the absorption coefficient in air as a function of the relativity humidity for different wave frequencies.

(d) Variation of the absorption coefficient in air as a function of the relativity humidity for different air temperature.

(e) Variation of the absorption coefficient in air as a function of the air temperature for different wave frequencies.

(f) Variation of the absorption coefficient in air as a function of the wave frequency for different air temperature.

Figure 3.11: Impact of the room ambient conditions and the wave frequency into the absorption coefficient in air.

Figure 3.12 presents a comparison between propagation losses and dispersion attenuation in air, none of them can be discarded by the simulator, both have a great impact in the total attenuation.

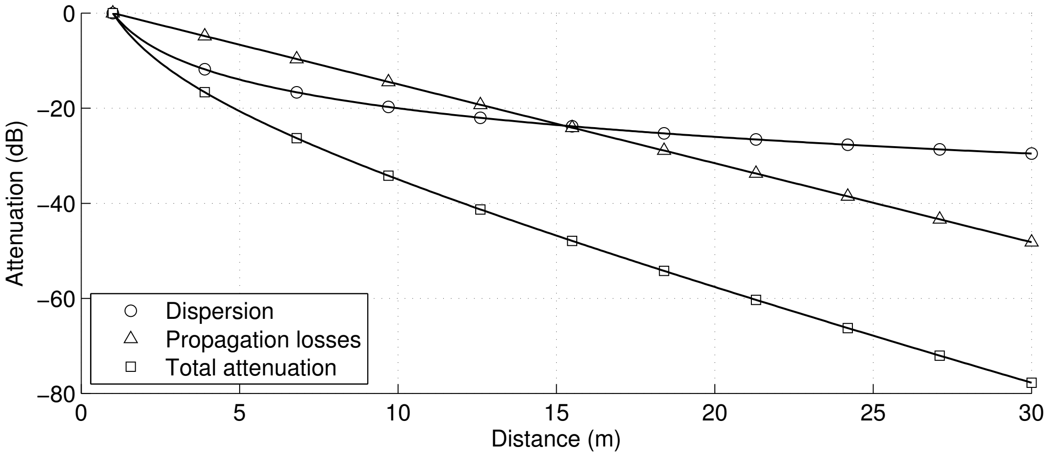


Figure 3.12: Comparison between atmospheric and dispersion attenuation in air, considering 1 m as the reference distance and the following ambient conditions: 20ºC of temperature, 50% of relative humidity, 50 kHz of frequency and 1 atm of pressure.

Combining the attenuation due to dispersion and due to propagation losses, Equations 3.11 and 3.12, the resultant acoustic wave pressure amplitude at distance d is given by:

$$P(d) = P_0 \times \frac{d_0}{d} \times 10^{-\frac{\alpha(d-d_0)}{20}}. \tag{3.14}$$

Taking into account the influence of the atmospheric parameters into the absorption coefficient in air it was considered, for the simulation purpose, that they can be set in the simulator but they remain constant over time. As a consequence, the absorption coefficient in air will be only a function of the wave frequency. Therefore, the atmospheric attenuation faced by an ultrasonic wave at d meters from the source is given by:

$$A_T(d, f) = \frac{d_0}{d} \times 10^{-\frac{\alpha(f)(d-d_0)}{20}} \tag{3.15}$$

which in dBs is given by:

$$A_T(d, f) = 20 \log\left(\frac{d_0}{d}\right) - \alpha(f)\,(d - d_0) \quad \text{dB}. \tag{3.16}$$

As a result of this, the $A_T(d, f)$ will be only a function of the wave frequency for a fixed distance d and therefore can be considered as a linear filter, which can result into an IR, $a(n)$.

To exemplify the process used in the simulator to obtain the IR, $a(n)$, it will be considered a receiver at 4 m from the source and the following ambient conditions: 20ºC of temperature, 30% of relative humidity and 1 atm of acoustic air pressure. Under this constraints the total attenuation in air, $A_T(4, f)$, is depicted in Figure 3.13.

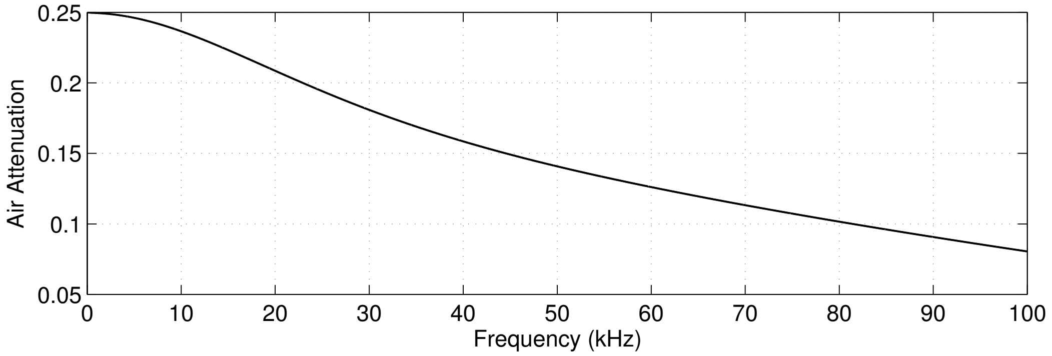
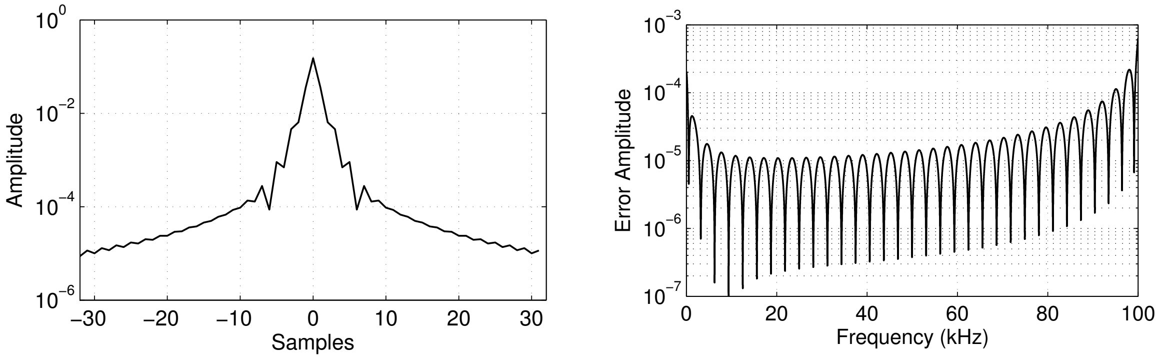


Figure 3.13: Total attenuation in air for a receiver at 4 m in reference to 1 m with the following ambient conditions: 20ºC of temperature, 30% of relative humidity and 1 atm of pressure.

The first approach to obtain the IR is to use the inverse Fourier transform (IFT). Which leads us to an infinite impulse response. Moreover, the resultant IR must be truncated in order to obtain an IR with only a few samples. Figure 3.14 presents the impulse response with 64 samples obtained by this process, it also shows the error produced in the frequency domain. Note that this IR must be delayed due to causality however this delay can be easily subtracted from the total delay resulted from propagation (in this example the IR must be delayed of 32 samples which represents 5.5 cm of distance). We can see in Figure 3.14b that the resultant error is very low and it can be considerably reduced by increasing the impulse response length. However, this approach will increase the minimum distance between the source and the receiver that can be simulated.



(a) Impulse response with 64 samples length.

(b) Error obtained by the truncation process.

Figure 3.14: Impulse response with 64 samples length that represents the total air attenuation and the error obtained by the truncation process.

Nevertheless, the error can be reduced if we discard the frequency response accuracy in some of the frequency components. For example, this simulator will be used for the ultrasonic range then it does not need to present a very good accuracy in the audible range (up to 20 kHz). Moreover, as a result of the fractional delay filter used in Section 3.3.1, which only performs well up to a quarter of the sampling frequency, the accuracy of the total attenuation frequency response between one quarter and one half of the sampling frequency can be reduced. As a result of these two constraints, the total attenuation frequency response must be accurate between 20 kHz and one quarter of the sampling frequency (which for this example represents 50 kHz).

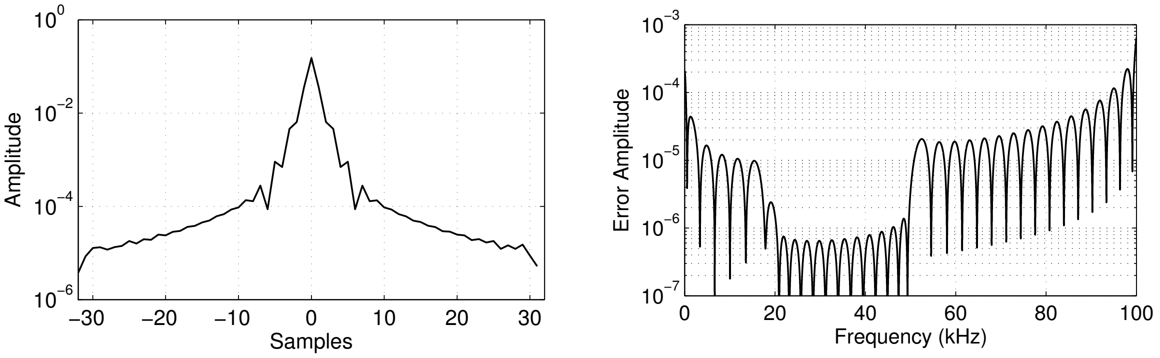
In order to increase the accuracy of one part of the spectrum while giving up elsewhere, we can use the algorithm presented in Figure 3.15 based on the Papoulis-Gerchberg algorithm [43, 44].

```
compute $A_0(k)$ from $A_T(d, f)$ in $K \gg L$ frequency points
$i = 0$
repeat
    $A_i(\mathcal{K}) = A_0(\mathcal{K})$
    $a(n) = \text{IFFT}\{A_i(k)\}$
    $a(n) = a(n) \times w_L(n)$
    $i = i + 1$
    $A_i(k) = \text{FFT}\{a(n)\}$
until $\|A_i(\mathcal{K}) - A_0(\mathcal{K})\|^2 < \epsilon$
```

Figure 3.15: Algorithm for reduction of the error in a specific band.

The algorithm starts by computing the total attenuation frequency response $A_T(d, f)$ using Equation 3.15 in K frequency points, obtaining $A_0(k)$. K must be greater than the desired impulse response length L to ensure proper frequency response interpolation. After that the impulse response, $a(n)$, is computed by the inverse discrete Fourier transform (IDFT) of $A_0(k)$. This impulse response is truncated with a length L rectangular window $w_L(n)$. From the truncated impulse response, the corresponding frequency response is computed using the discrete Fourier transform (DFT) resulting $A_1(k)$. If the error in the desired band is less than a given value ϵ the algorithm stops. If not, the frequency point from the band where the accuracy must be increased $A_1(\mathcal{K})$ are restored to its original amplitude $A_0(\mathcal{K})$ and the process is repeated.

By using the proposed algorithm it is possible to significantly reduce the error for a predetermined band. Using the same example applied to the band from 20 kHz to 50 kHz as discussed previously and running it one hundred times, it is possible to increase the accuracy of the total attenuation frequency response ten times for the same impulse response length. These results are depicted in Figure 3.16 where it can be seen that the total attenuation frequency response error in the band from 20 kHz to 50 kHz is reduced more than ten times see Figure 3.16b and Figure 3.14b.



(a) Impulse response with 64 samples length.

(b) Error obtained by the truncation process.

Figure 3.16: Impulse response with 64 samples that represents the total air attenuation and the error obtained by the truncation process using the proposed algorithm.

3.4 The Direct Path Case

At this point, it is possible to model the transmission, propagation and reception chain using separate IRs and without considering reflections. As a result of this, it is already possible to properly model direct path from the sent signal $x_m(m)$ to the received signal $y(n)$. Therefore the block diagram presented in Figure 3.3 especially the *sound propagation IR* can be expanded to express the different IRs involved in the process. Resulting in the diagram of Figure 3.17 in which the *sound propagation IR* is divided in two: the *air attenuation IR*, $a(n)$, and the *distance delay IR*, $d(n)$. The *air attenuation IR* models the total attenuation suffered by the ultrasonic wave and the *distance delay IR* is used to represent a possible fractional delay that need to be applied to the signal.

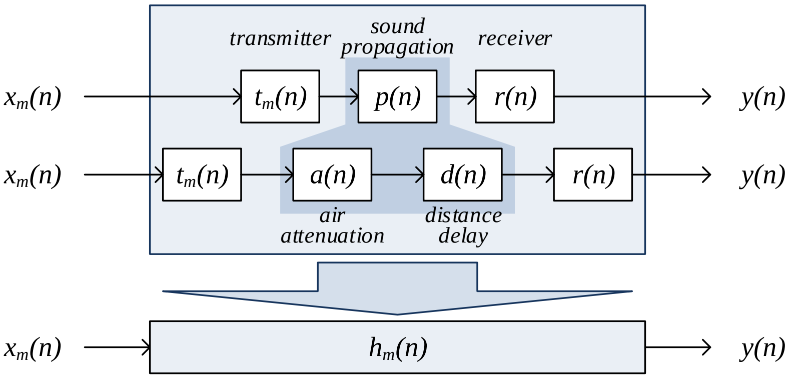


Figure 3.17: The division of $h_m(n)$ in four main IRs.

3.5 Multiple Reflections Model

Due to the number of obstacles to ultrasonic propagation usually present in rooms, this is a very important element of the simulator. However the theory of sound propagation in closed spaces can be very hard to apply and model and it may bring an important drawback associated to it: the simulation time. Nevertheless, ultrasonic waves have a huge advantage when compared to sound, which is that its maximum wave length is about 1.7 cm (for 20 kHz), for that reason a simple model can be used in the multiple reflections model, reducing the simulation time and user intervention without compromising the closeness to reality. On that account, the simulator presents the following characteristics and specifications, that will be detailed later:

1. All the surfaces in the room (e.g. walls) are modeled as quadrilaterals and therefore modeling curved surfaces is limited;
2. It is considered that the sound will propagate along sound rays;
3. All possible reflections up to a predefined order are computed and they are considered to be only specular in a plane surface with a specific IR (previously measured) associated to it. Diffraction and scattering are ignored due to its modeling and computational complexity;
4. All the possible received sound rays are subsequently validated by applying the ray-tracing method.

3.5.1 Surface Modeling

The polygonal based modeling, widely used in 3D computer graphics [45], was the basis to model any surface in the implemented simulator (e.g.: walls, floor and objects). Due to the typical geometric shape of the room surfaces we chose quadrilaterals instead of the conventional triangles as the basic element. Therefore, most of the room surfaces can be represented by making use of only one quadrilateral. However, some of them must be represented has a group of quadrilaterals connected to each other to form a particular geometric shape as depicted in Figure 3.18.

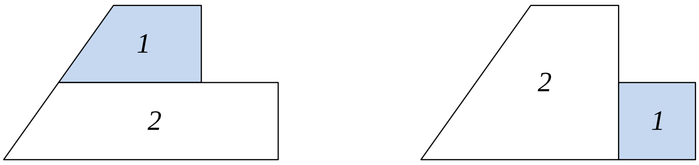


Figure 3.18: Possible divisions in two quadrilaterals of a complex shape.

Furthermore, many of the room models produced by specific software are based in polygonal modeling which may be easily adapted to be used into the simulator. It will be seen later that this type of geometry considerably simplifies the process of computing the surface reflections. However, it presents a huge disadvantage it is necessary to use a huge number of quadrilaterals to properly model curved surfaces. Figure 3.19 presents an example of a simple room with 8 surfaces that can be modeled using only 10 quadrilaterals in the simulator. Furthermore, the image presented is the one produced by the simulator, and allows the user to check if the room geometry is correct.

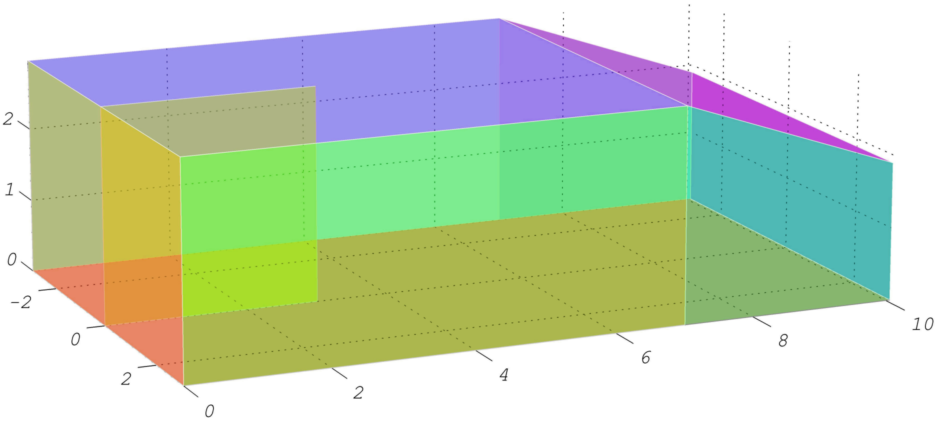


Figure 3.19: Example of a room modeled with 10 quadrilaterals.

3.5.2 Reflection Modeling

When an ultrasonic wave is emitted in a common room, the wave travels radially in all directions. When this wave finds an obstacle (e.g. a wall), the propagation direction changes and the ultrasonic wave reflects on the obstacle [38]. Figure 3.20 shows a possible ultrasonic wave propagation in a common room with two walls.

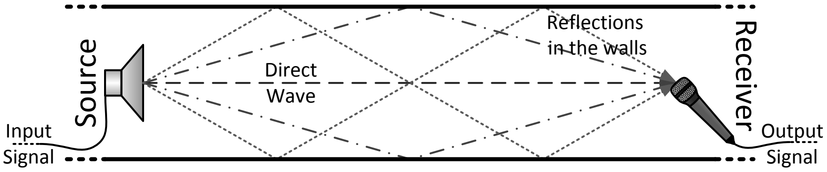


Figure 3.20: Typical ultrasonic wave propagation in a room with two walls.

Figure 3.21 illustrates the reflection of an ultrasonic wave from a punctual source on a plain surface (wall). The wave front from the source (continuous lines) collides into the wall and a new reflected wavefront (dashed line) is generated. As one can see a specular reflection sound wave on a wall can be interpreted as a sound wave produced by a virtual source.

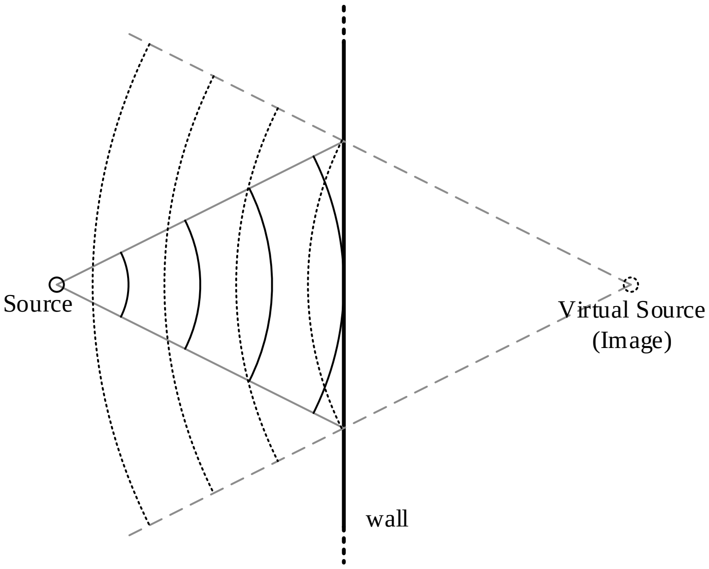


Figure 3.21: Ultrasonic wave reflection in a wall from a punctual source.

The best way to understand what happens with the ultrasonic wave is to consider the wall as a mirror, where the reflected wave front acts as if it comes from a image of the source produced by the mirror. This virtual source is placed at the same distance behind the wall as the distance of the source to the wall.

A reflected wave from a surface can collide with another surface and generates another reflected wave, and so on. In other words, even for a very simple room, it is extremely complex to compute all the possible wave reflections.

In Section 3.3.2 it was shown that the ultrasonic wave attenuates with distance, moreover, for each reflection the traveled distance increases, therefore, the wave energy decreases with the number of reflections until a point that it will be irrelevant to the system (it will be impossible to distinguish it from the noise). Therefore, the best metric to chose the number of reflections that the system must consider is the maximum wave energy after a certain number of reflections.

On the other hand, the reflections, as it was formulated, only occur in objects greater than the wavelength (about 1.7 cm for 20 kHz) [46]. Objects that are smaller than the wavelength are "transparent" to the wave, in other words the ultrasonic wave is not reflected by objects that are smaller than its wave length.

When studying the propagation of ultrasounds it is better to consider that they travel like rays and the wave front is flat [46]. In other words, the ultrasonic wave behaves, in the presence of an object, as a ray of light behaves in the presence of a mirror, as can be seen in Figure 3.22. This means that it is possible to apply the well known Snell Law, from the optics theory, to determine the reflection wave direction

$$\theta_i = \theta_r \tag{3.17}$$

and for the transmitted angle

$$\frac{\sin\theta_i}{c} = \frac{\sin\theta_t}{c_t}, \tag{3.18}$$

where c_t is the ultrasonic wave speed in the medium where the wave was reflected.

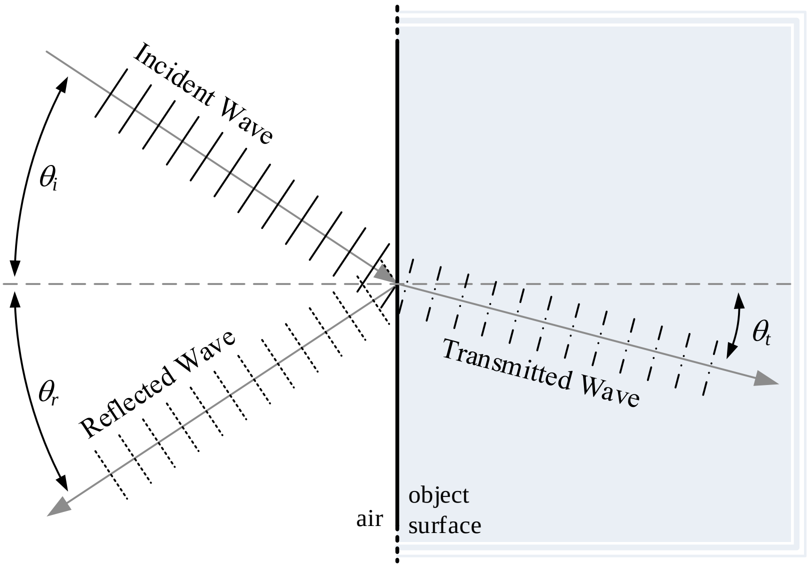


Figure 3.22: Ultrasonic wave reflection in an object (Snell Law).

It is important to notice that in the ultrasonic room simulator, the transmitted wave is discarded due to its low energy, therefore it will be consider that the transmitted wave is completed absorbed by the object [46]. Nevertheless, the reflected wave could have less energy than the incident wave. For most of the solid materials (e.g. wood, metal, plastic, glass, etc) the reflected wave has almost the same energy as the incident wave. Nevertheless, some *soft* materials in a common room can absorb some wave energy and consequently the reflected wave has less energy than the incident wave (e.g. carpets, cork, curtains, etc). In addiction to these, if the energy lost in the reflection depends on the frequency of the incident wave, an IR of the wall reflection must be considered. Nevertheless, similar to what happens with the transmitter and the receiver, this has to be carried out only once and it is considered to be independent of the room conditions.

Moreover, like it was said above, the reflection is considered to be specular, but this is not entirely true in the ultrasonic wave reflection if scattering could occur, although its energy is very low. Due to its complexity, the effect of scattering was not considered in the simulator.

In order to model the reflection in surfaces there are two classical geometric methods to simulate the acoustic reflection in large rooms: the Ray Tracing Method and the Image Source Method [47]. In both methods, the accuracy increases as the frequency increases and they present very good results for high audible frequencies [47]. Therefore, at the ultrasonic range they will produce better results with an excellent simulator performance, avoiding the necessity of using more complex methods.

Ray Tracing Method

The ray tracing method uses a very high number of rays which are emitted by the source in several directions [31]. The path of each ray is calculated and when these paths intersect on a surface a reflection occurs, the wave takes a new direction, according to the Snell law, and the wave is filtered by the IR of the surface.

In order to find which rays are received by the receiver it is necessary to define an intersection volume around the receiver to obtain the received rays. With this method it is difficult and costly to obtain a good accuracy. Figure 3.23 shows a small example in 2D of a transmitter and a receiver in a room with two walls. It is assumed that the source emits 16 rays and the receptor only receives ray 3 directly, rays 4 and 8 are reflected once and ray 7 twice. The resultant received wave is the combination of the four received waves.

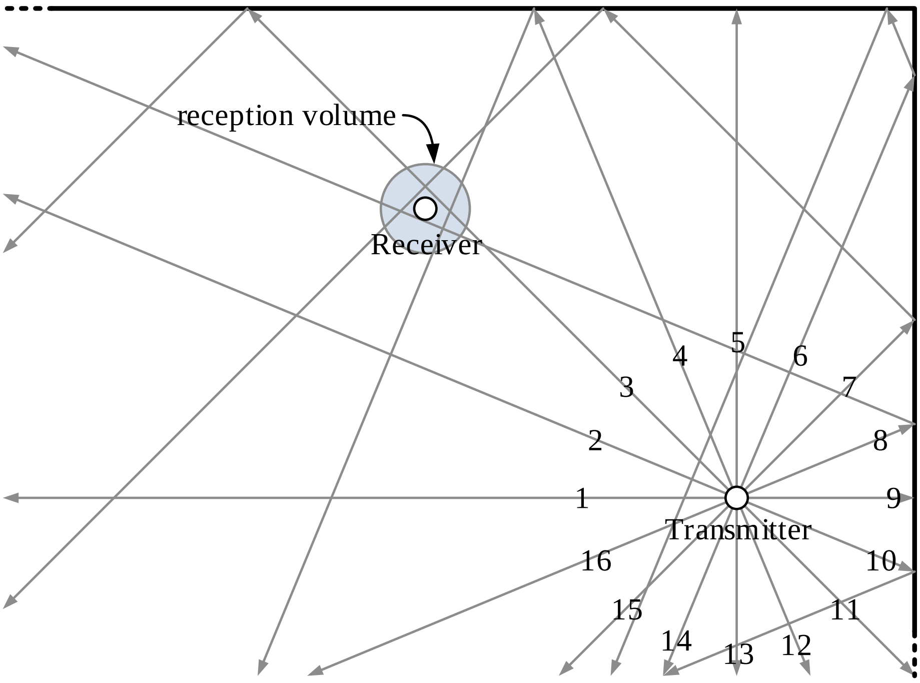


Figure 3.23: Ray tracing method (example with 16 rays).

In some cases, pyramids may be used in place of rays [31], each pyramid has its main vertex on the center of the source and it “grows” as the wave travels in the air as it is shown in Figure 3.24. If the receptor is inside of a given pyramid it means that the receptor receives the energy from that pyramid.

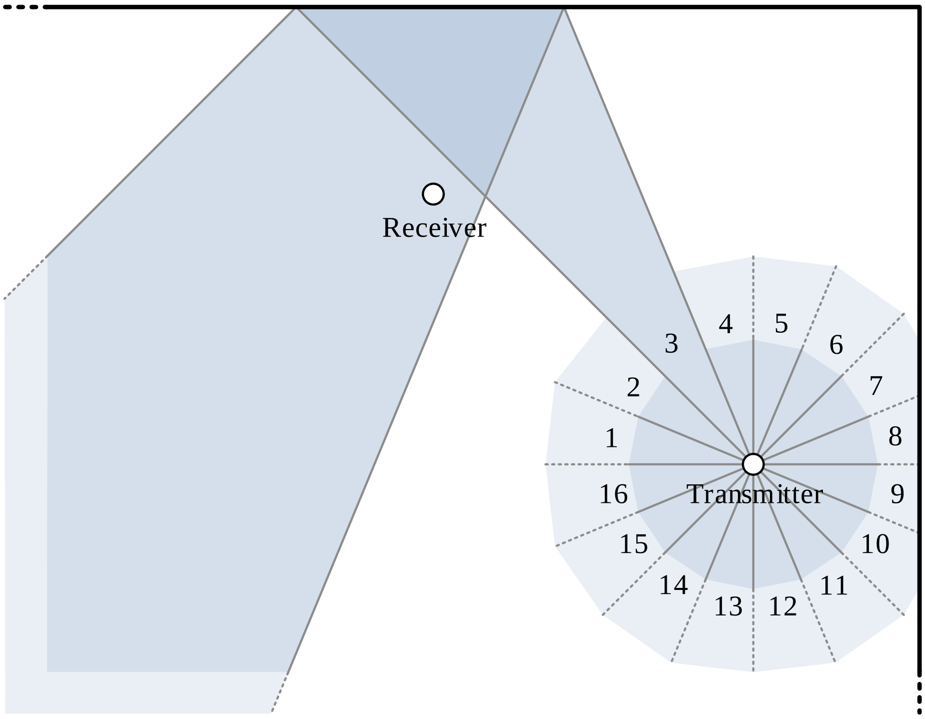


Figure 3.24: Example of ray tracing method using 16 pyramids (only pyramid 3 is shown).

In both cases, the main drawback of theses methods is the risk of receiving false reflections and do not receive some reflections that exist due to an insufficient number of rays.

The probability of a ray to intersect a surface with area A after traveling t seconds with a wave front less than $A/2$ is high. Therefore, the minimum number of rays N_{rt} that the system must consider is [31]:

$$N_{rt} \geq \frac{8\pi c^2}{A} t^2. \tag{3.19}$$

As a result of this the minimum number of rays necessary in a common room is usually very high. For example, in a room where the smallest surface of all objects has 100 cm^2 and considering a maximum wave travel time of 1 second it is necessary to consider at least 300 millions of rays.

Virtual Sources Method

The Virtual Sources Method is based on the principle that an ultrasonic reflection is a specular reflection (as mentioned before), so each reflection can be built by mirroring the original source on the reflected surface. In the daily life it is normal to have the perception that an object's reflection in a mirror is behind it, we are not seeing the source but a virtual image of it which it is created by a reflection in the mirror. This approach can be used on ultrasonic waves where the reflected wave is produced by a virtual source [31]. Moreover, the Snell law can be applied, therefore, the virtual source is placed behind the mirror at the same distance d as the source is in front of the mirror and the line that connects the source to its image is perpendicular to the surface as explained in Figure 3.25.

In a rectangular room with volume V the total number of virtual sources, N_{vs}, confined to a

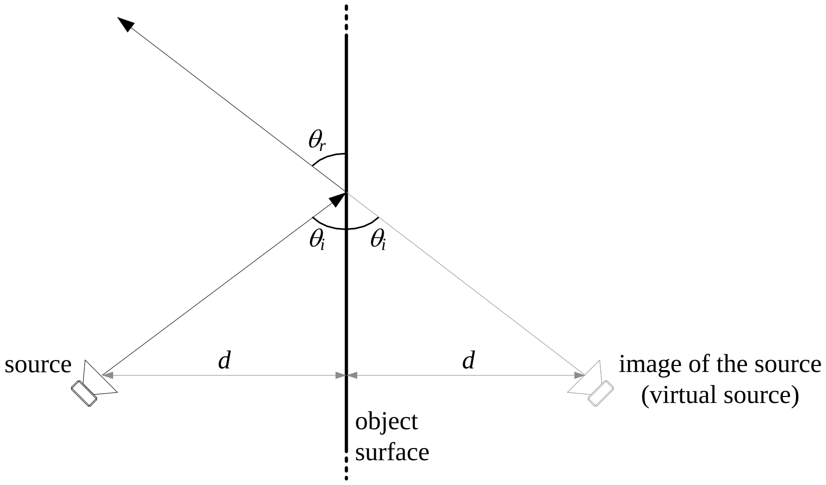


Figure 3.25: Virtual Source Method principle ($\theta_i = \theta_r$).

circumference with radius ct centered in the source can be easily determined [31],

$$N_{vs} = \frac{4\pi c^3}{3V} t^3. \tag{3.20}$$

The best advantage of the Virtual Sources Method is that it is very accurate but the complexity of the method grows with the complexity of the room. For example, for a room with N different surfaces, there are N virtual sources that produce reflections of first order and $N-1$ that produce reflections of second order [31]. Therefore, the total number of virtual sources N_T that create all the reflections up to order r in a room is

$$N_T = 1 + \frac{N(N-1)^r - N}{N-2} \approx (N-1)^r. \tag{3.21}$$

Considering a common room with only 6 surfaces (the simplest one: 4 wall, ceil and floor) and the shortest distance between two surfaces is formed by the ceil and the floor and it is 2.5 m. Considering, also, that the necessary propagation time to have a good simulation is 1 s (which corresponds to a distance of 170 m) it will be needed to compute all reflections up to the 68th reflection, which means that it will be necessary to compute $\approx 10^{47}$ virtual sources. This huge number of sources is explained mainly due to the reflection exponential factor, this is one of the reasons that the virtual source method is only useful to simple room configurations with small reflections order. However, if the position of the receiver is well known, some of the sources can be removed, because they will not contribute to the received signal.

To explain how to compute all the virtual sources up to a certain number of reflections, it is important to establish the next nomenclature: The real source is denominated by $\mathbf{S}$, the receiver by $\mathbf{R}$ and the virtual sources by $\mathbf{S}_{a,b,c,...}$, where a, b and c means that the wave produced by $\mathbf{S}_{a,b,c,...}$ is the same wave produced from the source $\mathbf{S}$ that it is firstly reflected in surface a, the produced wave is then reflected in surface b, then in c and so on.

Firstly, all the virtual sources that produce reflections of first order $\mathbf{S}_a$ are computed, using the example presented in Figure 3.26, where there are 3 different reflective surfaces, the virtual sources

$\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$ can be obtained by mirroring the real source **S** in surfaces 1, 2 and 3 respectively. One can see in Figure 3.26 that the received waves are perceived by **R** as coming from the virtual sources.

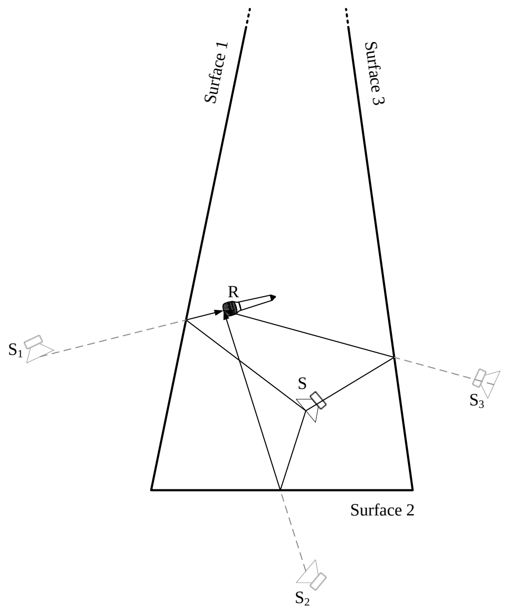


Figure 3.26: Example of Virtual Sources that produce reflections of first order.

In Figure 3.27 all the virtual sources that produce reflections of second order $\mathbf{S}_{a,b}$ are presented, the virtual sources $\mathbf{S}_{a,1}$, $\mathbf{S}_{a,2}$ and $\mathbf{S}_{a,3}$ can be obtained by mirroring the real source **S** in the surfaces 1, 2 and 3 respectively. From Figure 3.26 it can be seen that there are no virtual sources that reflect successively in the same surface $\mathbf{S}_{a,a}$, moreover, the virtual source $\mathbf{S}_{1,2}$ does not produce any valid reflection to the receiver position. This happens manly because to consider a virtual source as a valid source it must respect two conditions:

- The virtual source must be located behind the plane of the surface that produced it from the receiver point of view;
- The line that connects the virtual source and the receiver must intersect the surface and not only the plane of the surface.

The virtual sources that produce reflections of third order in the same previous example are presented in Figure 3.28.

Hybrid Method

The Hybrid Method is a combination of the Ray-Tracing Method and the Virtual Sources Method. Firstly, all the virtual sources are created and then each incoming ray is validated by using the ray-tracing method. This approach is very useful to remove some incoming rays that will not reach the receiver because they collided in other surface that are not included in the supposed reflections.

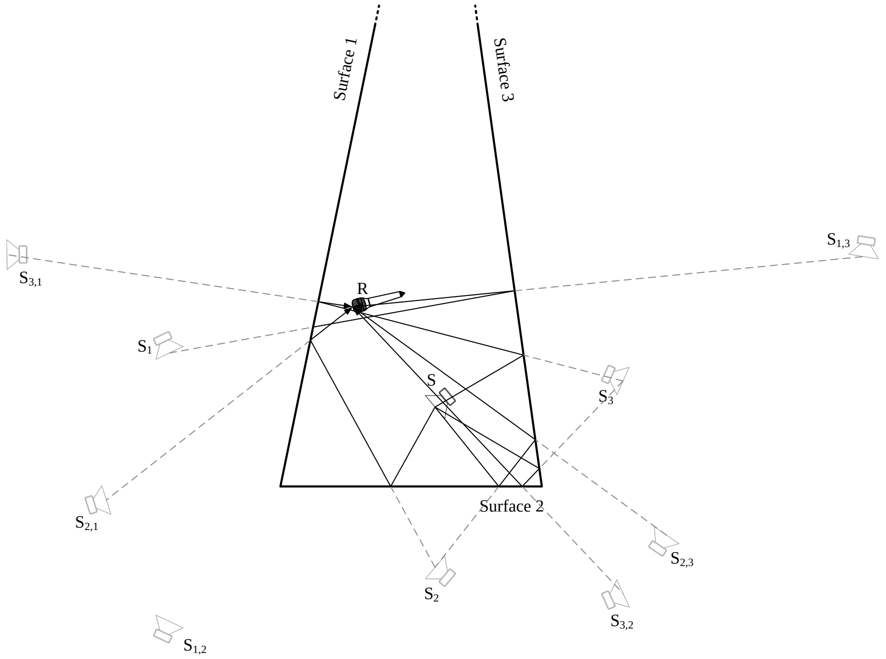


Figure 3.27: Example of Virtual Sources that produce reflections of second order.

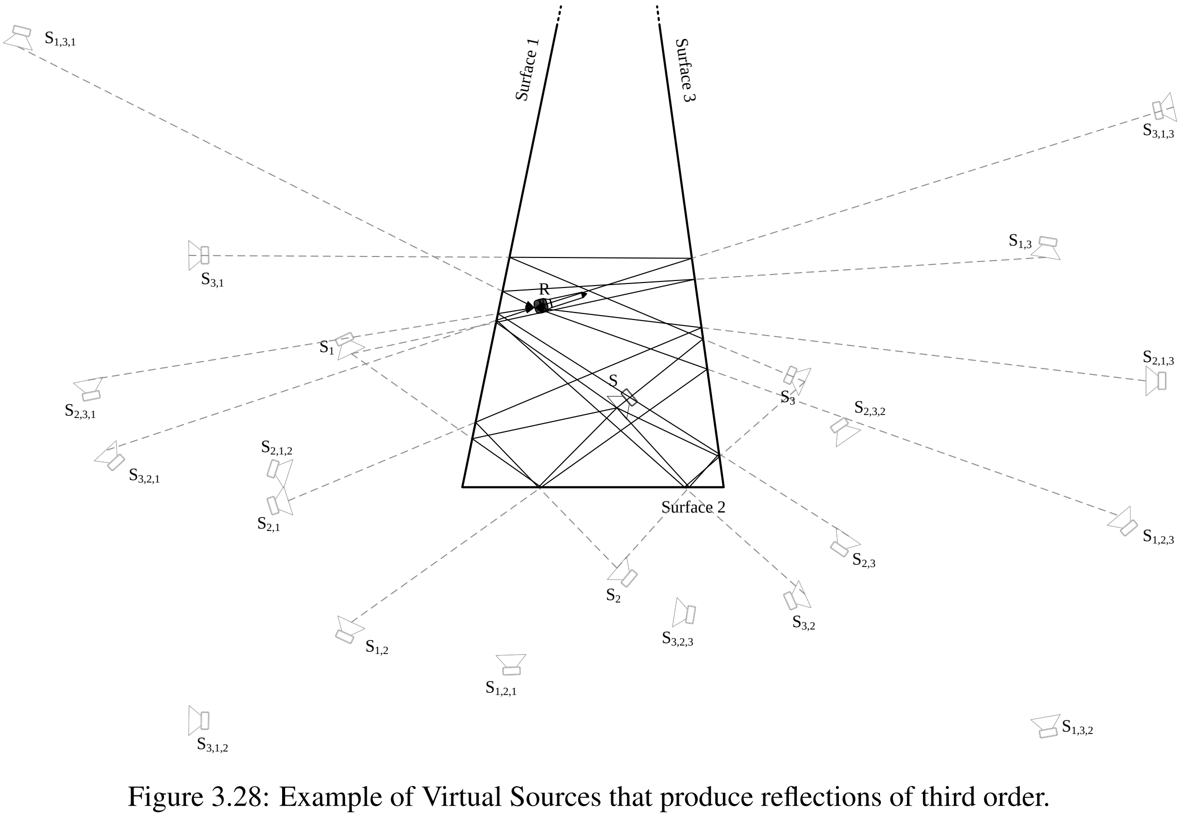


Figure 3.28: Example of Virtual Sources that produce reflections of third order.

3.6 Implementation

To implement the proposed simulator it was decided to use *Matlab*[4]. Since *Matlab* is not freeware software an effort has been made, during the simulator implementation, to use only the typical *Matlab* functions that can be found in several compatible software packages (such as *GNU Octave*[5]) by this way the implemented simulator can be used almost freely. Moreover, the use of this tool will allow the user to have complete control over the simulation and understanding its behavior. Another important choice that was made in the simulator implementation was it modular structure. Which will allow that each module can be substituted by a user developed module and it will facilitate future expansions of the simulator. In addition to this, each module can be easily modified by the user, in order to obtain the desired behavior.

The simulator can be divided into 6 different modules and 7 different methods for modules interaction. A module can be a simple data structure or a complex object depending on its goal. The methods are designed to obtain the information from the modules and/or create new modules. The interaction between the different modules and methods is schematically represented in Figure 3.29.

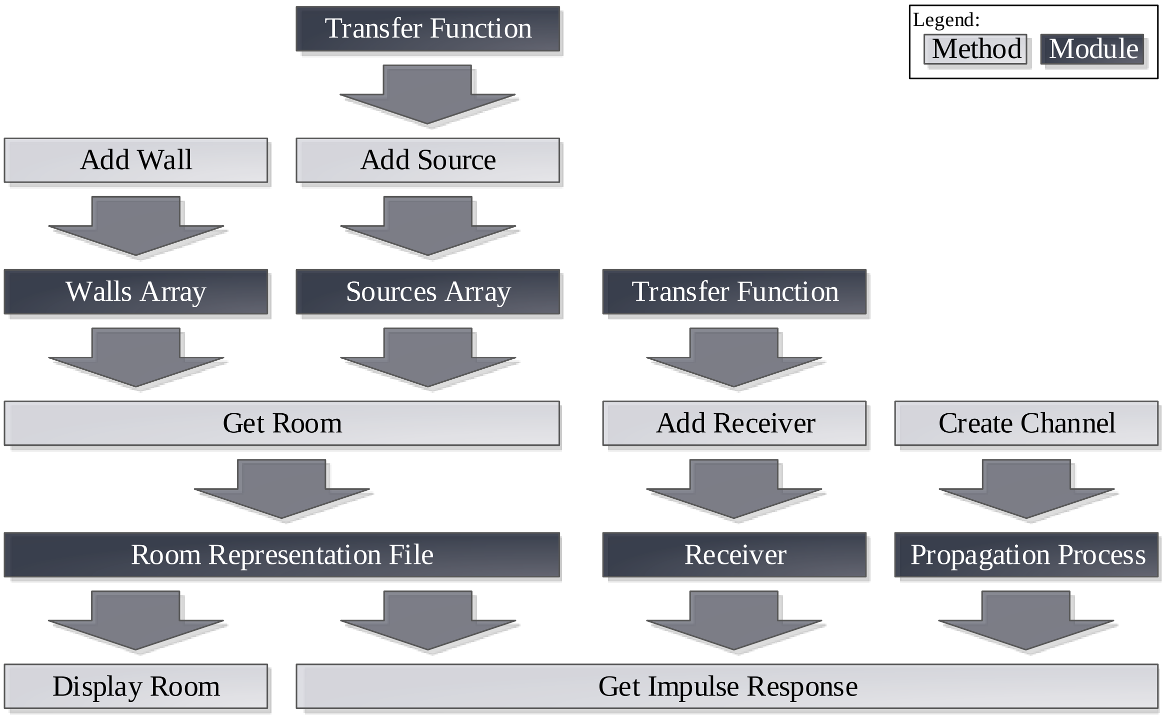


Figure 3.29: Modules and methods interaction in the implemented simulator.

[4] *Matlab* is a numerical computing environment developed by MathWorks, it allows the implementation of complex algorithms providing several tools to reduce the implementation time.

[5] *GNU Octave* is a freeware software very similar to *Matlab* and most of the algorithms implemented in *Matlab* can be run in *GNU Octave* without almost any changes.

3.6.1 Walls Array Module

The *walls array* module is a simple data structure that contains the information of each surface. Each surface in the array is represented by a reflection coefficient and by the four edges of the quadrilateral as discussed previously in Section 3.5.1. This module has a creator method, which creates the array and fills it with the data.

3.6.2 Transfer Function Module

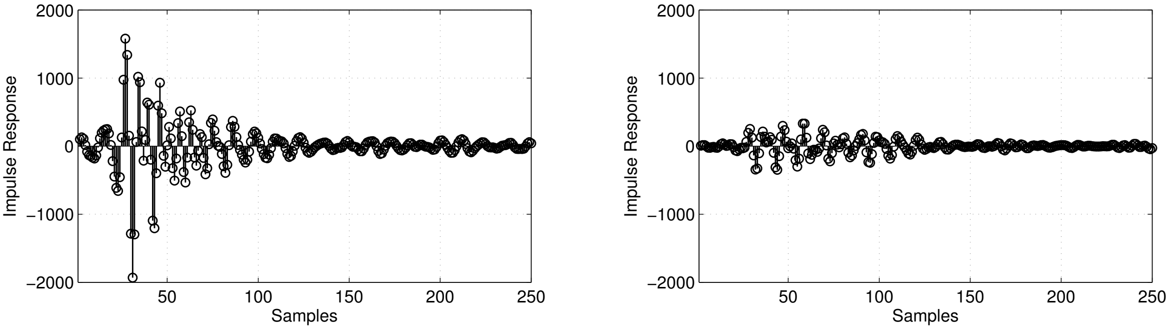
The *transfer function* module is an object that, for a particular source or receiver, provides to the user the IR for a given azimuth and elevation. Depending of the origin of the information this module can have different implementations, however, this must me transparent to the user. In the present simulator implementation, this module needs to have five inputs:

- the direction represented by the **azimuth** and the **elevation**;
- the operation frequency band represented by the **low** and the **high frequency**;
- the **sampling frequency**.

From these inputs the module most provide two outputs:

- the **impulse response**;
- the **index of the first sample**, which can be used for example to avoid the causality problem of the impulse response.

In Figure 3.30 two examples of IR produced by a speaker are depicted, in both cases, the first sample is -20. As can be seen, this speaker produces much more energy at 0º azimuth than at 45º azimuth. This module is used by the the method *add source* to add a source into the simulator and the method *add receiver* to add a receiver into the simulator.



(a) 0º of Azimuth.

(b) 45º of Azimuth.

Figure 3.30: Impulse response given by the transfer function module for different azimuths and 0º of elevation, sampling at 250 kHz. In both cases the the operation bandwidth was set from 20 kHz to 50 kHz.

3.6.3 Sources Array Module

The *sources array* module is a simple data structure that contains the information of each source placed in the room. Each source in the array is represented by:

- the position in the room (x, y, z **coordinates**);
- the coordinates of the internal **source reference** in respect to the room reference (e_x, e_y, e_y);
- the **source transfer function module**.

This module has a creator method, which creates the array and fills it with the data.

3.6.4 Room Representation File Module

The *room representation file* module is a simple data structure store in a file that contains;

- the information of all the room surfaces;
- all the sources in the room (real and virtual up to a certain reflection order).

This module has a creator method, which by processing the walls array and the sources array can create the file and fill it with the data for a maximum reflection order given by the user. In addition to this method the *room representation file* module has a specific method to show graphically the information stored in the file: the *display room* method. Using the same example of Section 3.5.1 Figure 3.31 presents four possible outputs of this method. Figure 3.31a presents the room without reflections and therefore only the real source marked with a yellow circle is shown. Figure 3.31b presents the same example but considering just one successive reflection. The virtual sources, that produce the reflections are marked with a yellow circle and the real source with a red circle. In Figures 3.31c, 3.31d, 3.31e and 3.31f it is considered a maximum of two, three, four and five successive reflection respectively.

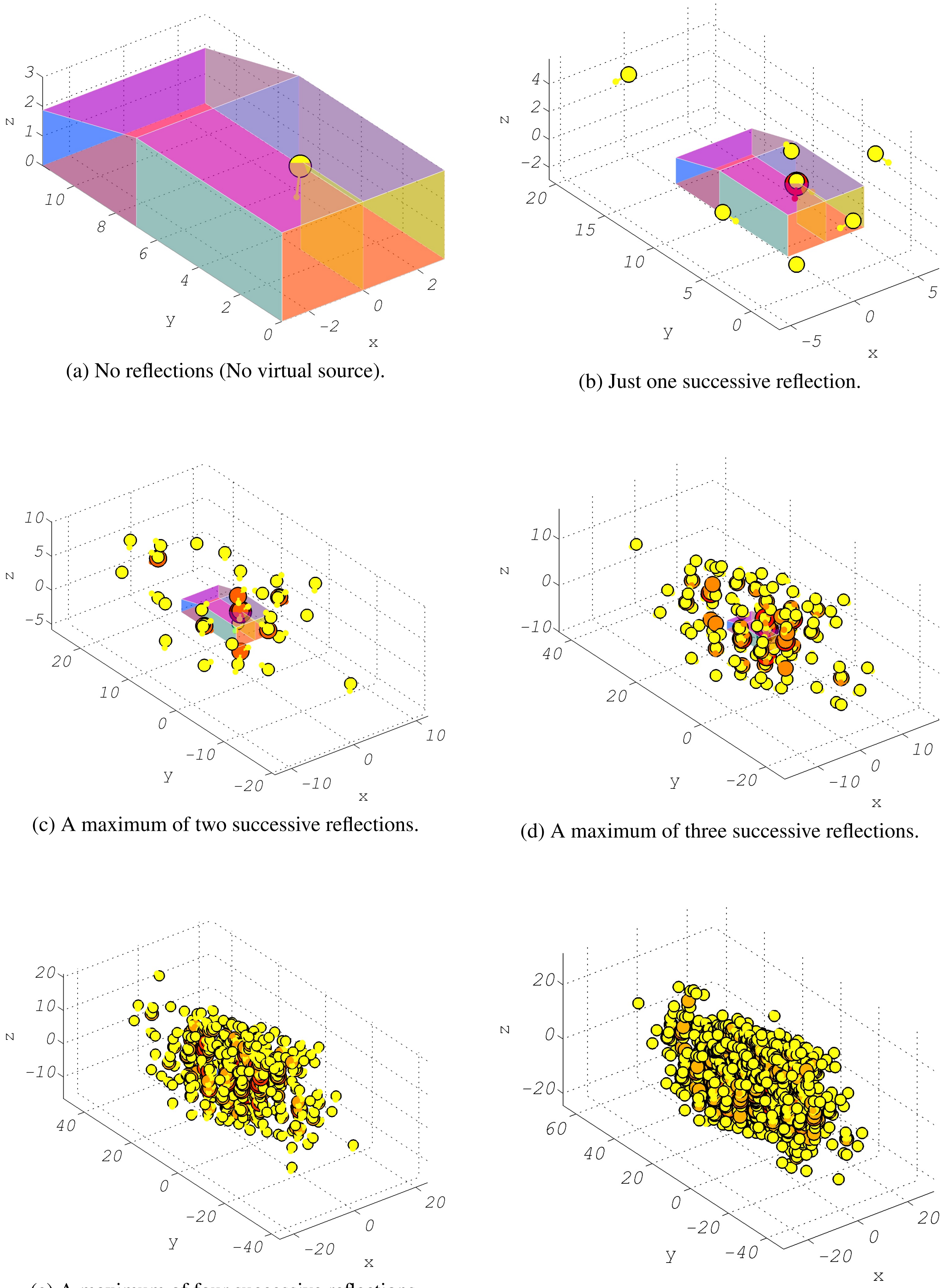
Note that, this file is independent of the room ambient conditions and the receiver position. Therefore, the resultant file can be used for any room ambient conditions and for any receiver position without the need to create it again.

3.6.5 Receiver

The *receiver* module is similar to the *sources array* module. The biggest difference is that it only has one receiver instead of having an array of them. Therefore, the data structure contains:

- the receiver position in the room (x, y, z **coordinates**);
- the coordinates of the internal **receiver reference** in respect to the room reference (e_x, e_y, e_y);
- the **receiver transfer function module**.

This module has a creator method, *add receiver*, which creates it and fills it with the data.



(a) No reflections (No virtual source).

(b) Just one successive reflection.

(c) A maximum of two successive reflections.

(d) A maximum of three successive reflections.

(e) A maximum of four successive reflections.

(f) A maximum of five successive reflections.

Figure 3.31: Output example of the display room method using the room representation file.

3.6.6 Propagation Process Module

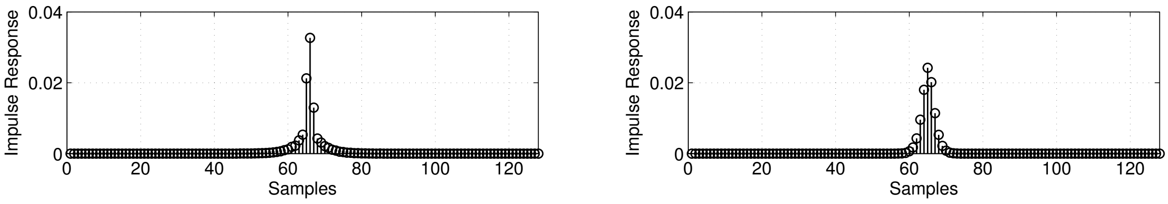
The *propagation process* module is an object that for particular ambient conditions provides the user with the IR for a given distance to the source. This IR only takes into account the ultrasonic wave attenuation presented in Section 3.3. In the present simulator implementation this module needs to have seven inputs:

- the **distance** to the source;
- the ambient conditions: **temperature**, **relative humidity** and **pressure**;
- the operation frequency band represented by the **low** and the **high frequency**;
- the **sampling frequency**.

From these inputs the module most provide two outputs:

- the **impulse response**;
- the sample **index of the first sample**, which can be used for example to avoid extensive IR due to a huge distance to the source.

Figure 3.32 depicts two examples of IRs produced by the atmospheric attenuation, in both cases the first sample is index 7095. One can see that the atmospheric attenuation is different and it attenuates more at 30ºC than at 10ºC, as expected.



(a) For 10ºC of temperature.

(b) For 30ºC of temperature.

Figure 3.32: Impulse response given by the propagation process module at 10 m from the source for two different room temperatures, 50% of relative humidity and 1 atm of atmospheric pressure and a sampling rate at 250 kHz. In both cases the operation bandwidth was set from 20 kHz to 50 kHz.

3.6.7 Get Impulse Response Method

This is the most important method in the simulator due to the fact that it collects and combines all the information in other modules to compute the IR from each source. In the first stage, by using the *receiver* module, it validates all the sources (real and virtual) in the *room representation file* module and it obtains the source output direction for each valid source. The total attenuation due the reflection coefficient of each surface and the total traveling distance and the receiver input direction. In the second stage, it computes the IR from the *transfer function* module of the source, the IR from the *transfer function* module of the receiver and the IR from the *propagation process* module. At the end it groups together all the sources that represent the same real source and computes the resultant impulse response for each real source.

Figure 3.33 presents an example of an impulse response obtained in the center of the room of

Figure 3.31 using a maximum of five successive reflections.

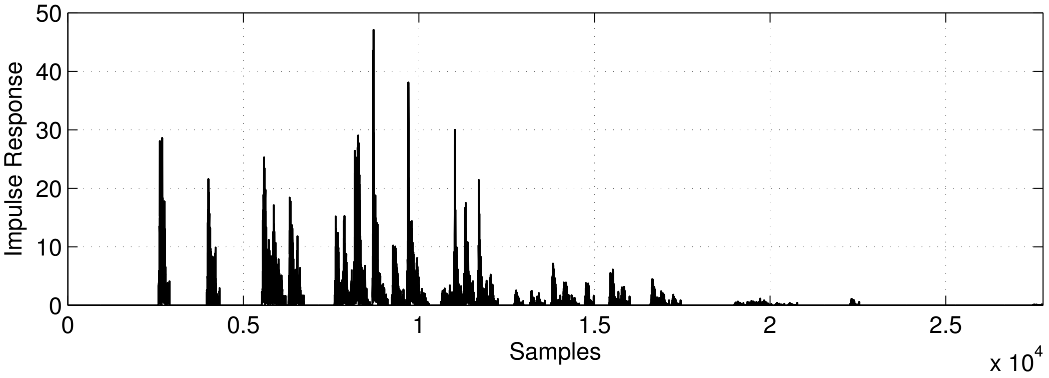


Figure 3.33: Example of the absolute value of an impulse response given by the *get impulse response* method considering a maximum of 5 reflections.

3.6.8 Performance

To evaluate the simulator performance several simulations were carried out with different parameters in the same machine under the same circumstances. In these tests a PC with the characteristics presented in Table 3.1 was used. In order to avoid external interference from other programs running in the computer the Matlab process priority was set to *High*. In addition to this, each test was carried out at least 10 trials.

Description	Value
Matlab Version	2012b (64 bits)
Operating System	Microsoft Windows 7 SP1 (64 bits)
Processor	Dual Core 2.5 GHz (Intel T9300)
Memory	4 GBytes
Graphic Card	Nvidea Geforce 8600M GT

Table 3.1: Characteristics of the computer and software used in the simulator performance tests.

Room Complexity Impact

The first test measures the impact of the room complexity and the maximum number of successive reflections to consider on the simulation time. The simulation time was measured using one to twelve surfaces and considering a maximum successive reflection number of one to five. The simulation times are presented in Figure 3.34. From this figure it can be seen that the simulation time increases exponentially with the number of surfaces (>3) in the room for any number of maximum successive reflections. Considering three maximum successive reflections the simulation time starts at about 2 ms for three surfaces and goes up to 50 ms. Another important point is that the simulation time increases considerably with the maximum number of successive reflections. It starts around 3 ms for one reflection and ends at 20 s for five reflections for a room with twelve surfaces. Therefore, if the number of successive reflections is low, the number of surfaces in the room can be extremely high and the simulation time will be acceptable. On the other hand, if the number of successive reflections is high (more than four) in a room with several surfaces, the user is going to have to wait hours to get the results. Nevertheless, for the majority of the ultrasonic applications, a maximum of three or four successive reflections is more than enough to present a very good test bed.

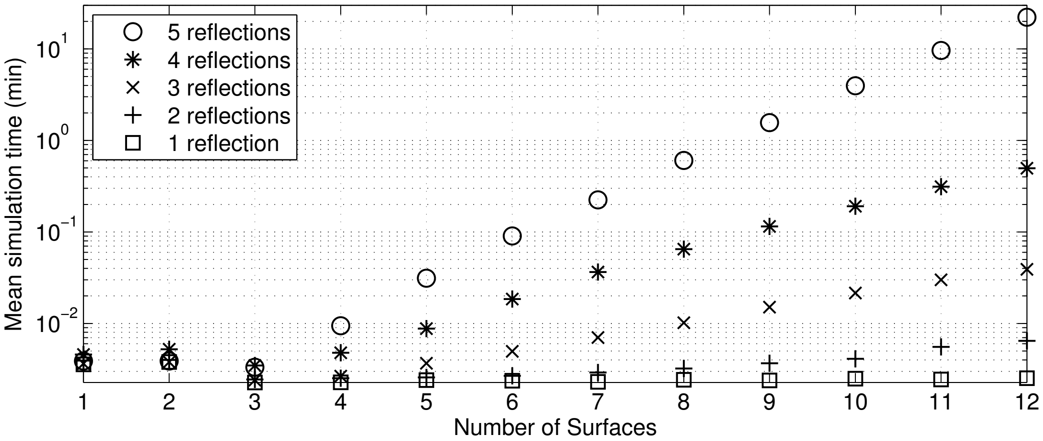


Figure 3.34: Simulation time as a function of the number of room surfaces and the maximum number of successive reflections to consider.

Number of Sources Impact

The second test measures the impact of the number of sources and the maximum number of successive reflections in the simulation time. The simulation time was measured using one to three sources and considering a maximum successive reflection number of one to six. The results from this test are presented in Figure 3.35. From this figure it can be seen that the simulation time increases exponential with the number of reflection and increases linearly with the number of sources. These results are consistent with the previous test and the simulator implementation, where each source is treated separately and independently.

3.7 Simulating the Doppler Effect

The Doppler effect is produced by the receiver and/or source movement, as shown in Figure 3.36. Moreover, the Doppler effect can be characterized by a compression or expansion of the acoustic signal, in the time domain, proportional to the source/receiver relative speed [37].

Nevertheless, the Doppler effect has a stronger impact on ultrasonic communications than in radio frequency communications due to the lower speed of sound. For example, a person traveling at 1 m/s produces a maximum Doppler shift [37] of 118 Hz on a 40 kHz carrier. For radio communications at 1 GHz, to have an equivalent Doppler shift, the person needs to travel at more than 35000 m/s (126000 Km/h).

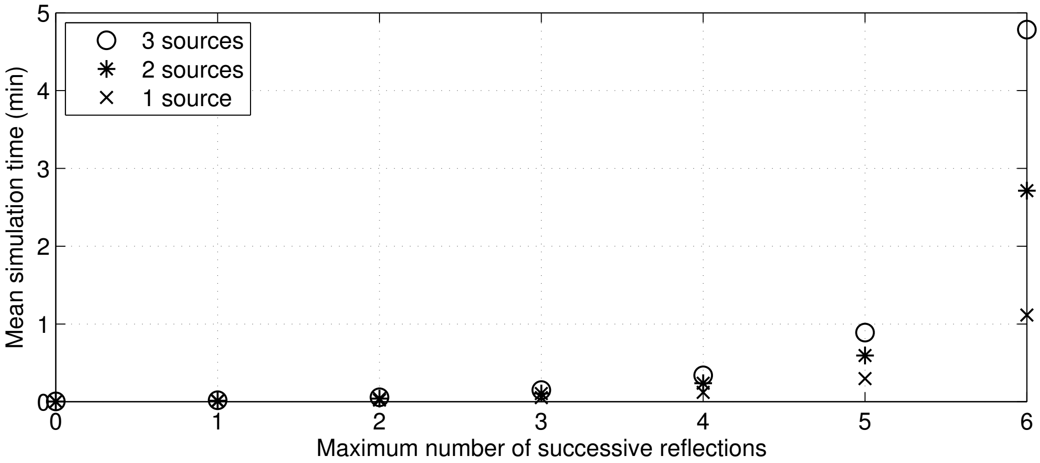


Figure 3.35: Simulation time in function of the maximum number of successive reflections to consider for different number of sources in the room.

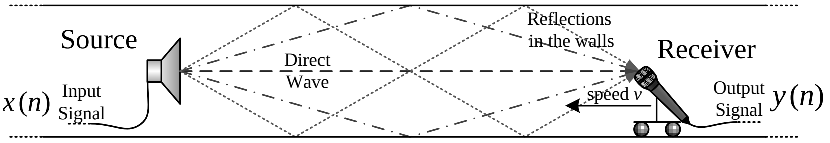


Figure 3.36: Doppler effect produced by the receiver movement.

When the receiver and/or source moves, the impulse response will change over time. Therefore, the impulse response will be a time function:

$$h(n, m) = h_n(m) \tag{3.22}$$

where the $h_n(m)$ is the impulse response for the instant n.

Figure 3.37 presents the block diagram of Doppler effect implementation. For each instant n the source and the receiver position must be determined and the impulse response must be computed using the simulator. After that the impulse response can be applied to the input signal $x(n)$ to obtain the output sample $y(n)$. For instant $n + 1$ a new impulse response must computed and applied to $x(n + 1)$ to obtain the output sample $y(n + 1)$. As a result of this, for each output sample an impulse response must be computed which leads to a extremely high computation time. However, it produces a very accurate result and it can be applied to any kind of movement.

3.7.1 Uniform and Rectilinear Movement

The simulation environment shown in Figure 3.38 was used to test the Doppler effect into the received wave, when the receiver is moving with constant speed. In this environment there is no wall and therefore no reflections. As a consequence of this, the only received signal is the direct signal from the source. The source was placed in the coordinate system origin (0; 0) and 0° beam was set to the y direction. It was used the model of a Kemo L10[6] speaker presented in Figure 3.39 which

[6] Kemo L10 is an ultrasonic speaker produced by Kemo Electronics for Animal Repellers. Its frequency response was measured using a Brüel & Kjær 4954-A Microphone.

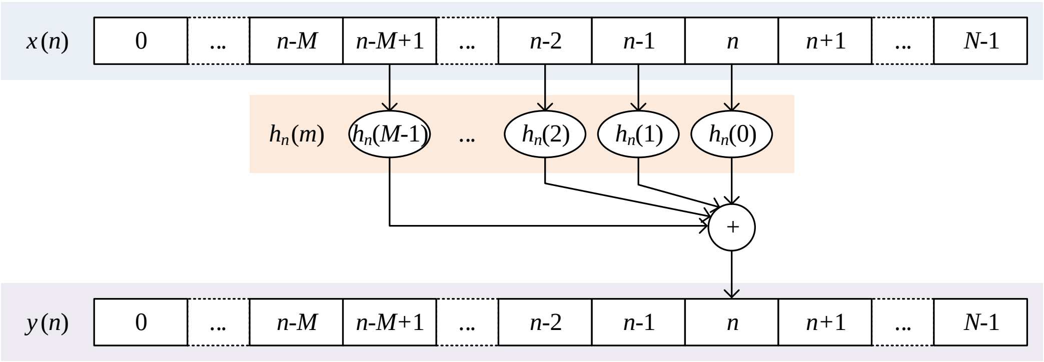


Figure 3.37: Block diagram for Doppler effect processing.

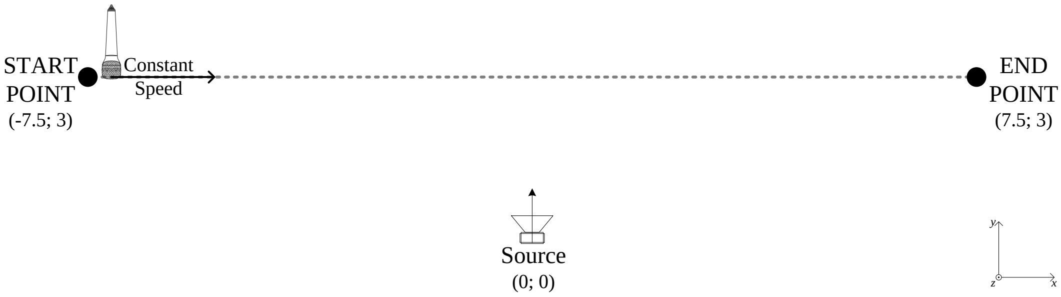
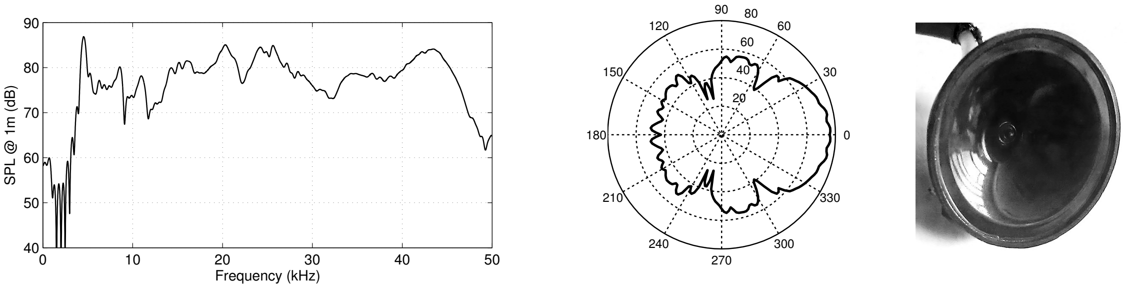


Figure 3.38: Simulation environment to test the Doppler effect for a receiver with uniform and rectilinear movement.

presents a beam around 40º for a frequency of 30 kHz as shown in Figure 3.39b. In addition to this, the receiver (an omnidirectional microphone) starts its movement at position (−7.5; 3) and it travels 15 m in the *x* direction with a constant speed of 1, 5 or 10 m/s.



(a) Frequency response at 0º.

(b) Radiation pattern at 30 kHz.

(c) Front view of the real speaker.

Figure 3.39: Ultrasonic speaker model used in the simulator.

In spite of the receiver constant speed the relative speed between the source and the receiver changes over time. Figure 3.40 presents the relative speed between the receiver and the source for a receiver with constant speed of 1, 5 and 10 m/s. From this figure it can be observed that the relative speed starts at around the receiver speed, v, and ends at around $-v$. At the extreme positions the relative speed decreases slowly. However, near to the source position, the relative speed decreases quite fast, it changes from $0.75v$ to $-0.75v$ in only 7 m.

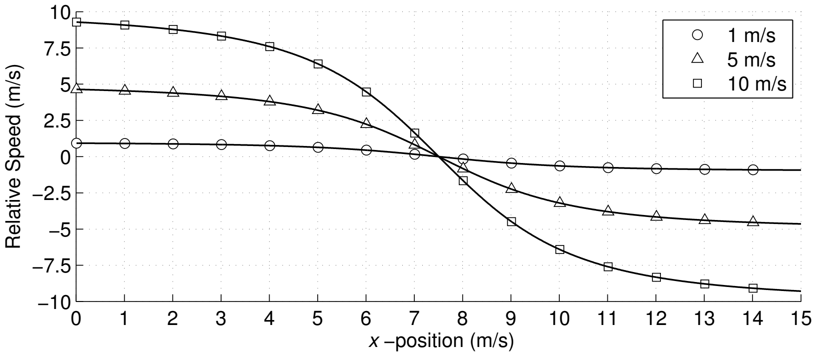


Figure 3.40: Relative speed between receiver and source for a receiver with uniform and rectilinear movement (moving at 1, 5 or 10 m/s).

From the literature [48] the Doppler frequency shift of a sinusoidal signal with frequency f by a receiver moving at v_r m/s relatively to the source is given by:

$$f_D = \frac{v_r}{c} f. \tag{3.23}$$

Figure 3.41 presents the Doppler frequency shift that a 30 kHz sinusoidal signal will suffer for the relative speed presented in Figure 3.40.

From (3.23), the maximum Doppler frequency shift that a sinusoidal signal can experience when only the source or the receiver is moving is:

$$f_{D\text{Max}} = \pm \frac{v}{c} f \tag{3.24}$$

where, v is the receiver/source speed. Therefore for a 30 kHz sinusoidal signal the maximum Doppler frequency shift is around ±87 Hz for a receiver moving at 1 m/s relatively to the source and around ±870 Hz for a receiver moving at 10 m/s. These results corroborate the Doppler frequency shift presented in Figure 3.41.

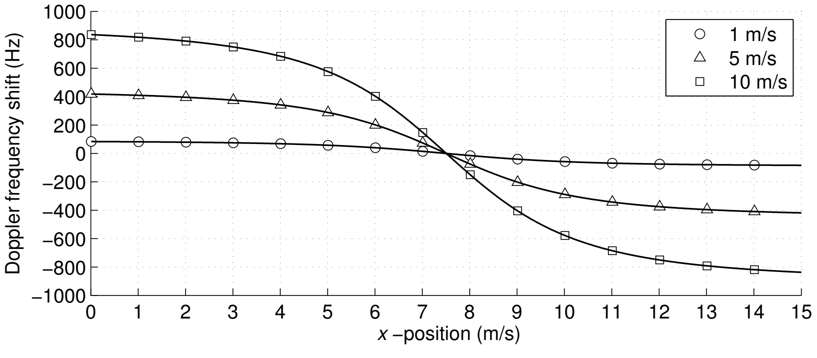


Figure 3.41: Doppler frequency shift experienced in receiver with uniform and rectilinear movement (moving at 1, 5 or 10 m/s).

Simulator Results

Using the room acoustic simulator it is possible to obtain the IR for all the receiver positions and applying the algorithm from Figure 3.37 it is possible to compute the received signal.

The first test considers that the source signal is only composed by a sinusoidal signal with 30 kHz. The spectrogram of the received signal is presented in Figure 3.43 where a receiver speed of 1, 5 and 10 m/s was considered. As can be seen the received signal's frequency changes as predicted in Figure 3.41.

The second test (Figure 3.44) uses the same simulation parameters of the first test but instead of considering a single sinusoidal signal it considers that the source signal is composed by multiple sinusoidal signals with frequencies from 29 to 31 kHz in steps of 500 Hz. At first glance, the Doppler effect in the five sinusoidal signals seems to be the same. However, a more detailed analysis, especially in 10 m/s case, reveals that the Doppler frequency shift is different for the five frequencies. Regarding Figure 3.44c the 31 kHz sinusoidal signal is near to 32 kHz (31.84 kHz) and the 29 kHz sinusoidal signal is near to 30 kHz (29.78 kHz). These results arise from the fact that the Doppler frequency shift depends on the frequency value and, consequently, a sinusoidal signal with a higher frequency is going to suffer a bigger frequency shift than a sinusoidal signal with a lower frequency.

The third and last test (Figure 3.45), uses the same simulation parameters with the source signal composed by a BPSK modulated signal with 1 kbps and 30 kHz of central frequency. For communications under this conditions, it will be difficult to demodulate the received signal. Moreover, if the receiver uses a band-pass filter centered at 30 kHz in order to reduce the noise [49, 50], it will be important for the filter to accommodate those frequency shift ranges, therefore, more noise will be received. Another important aspect to point out is the two dark curves that appear in the spectrogram which are due to the fact that the radiation pattern presents a minimum around 55º for 30 kHz (see Figure 3.39b) and this minimum approaches to 0 if the frequency increases and deviates from it if the frequency decreases. Figure 3.42 presents the sound pressure level (SPL) produced at 1 m from the source in the direction of the receiver position. As one can see the SPL spectrogram presents the same dark curves as in Figure 3.45 apart of the Doppler frequency shift.

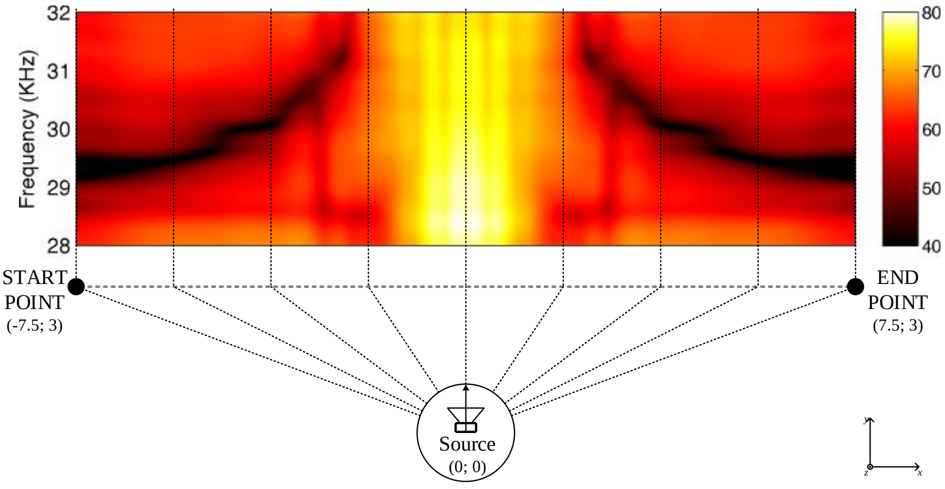
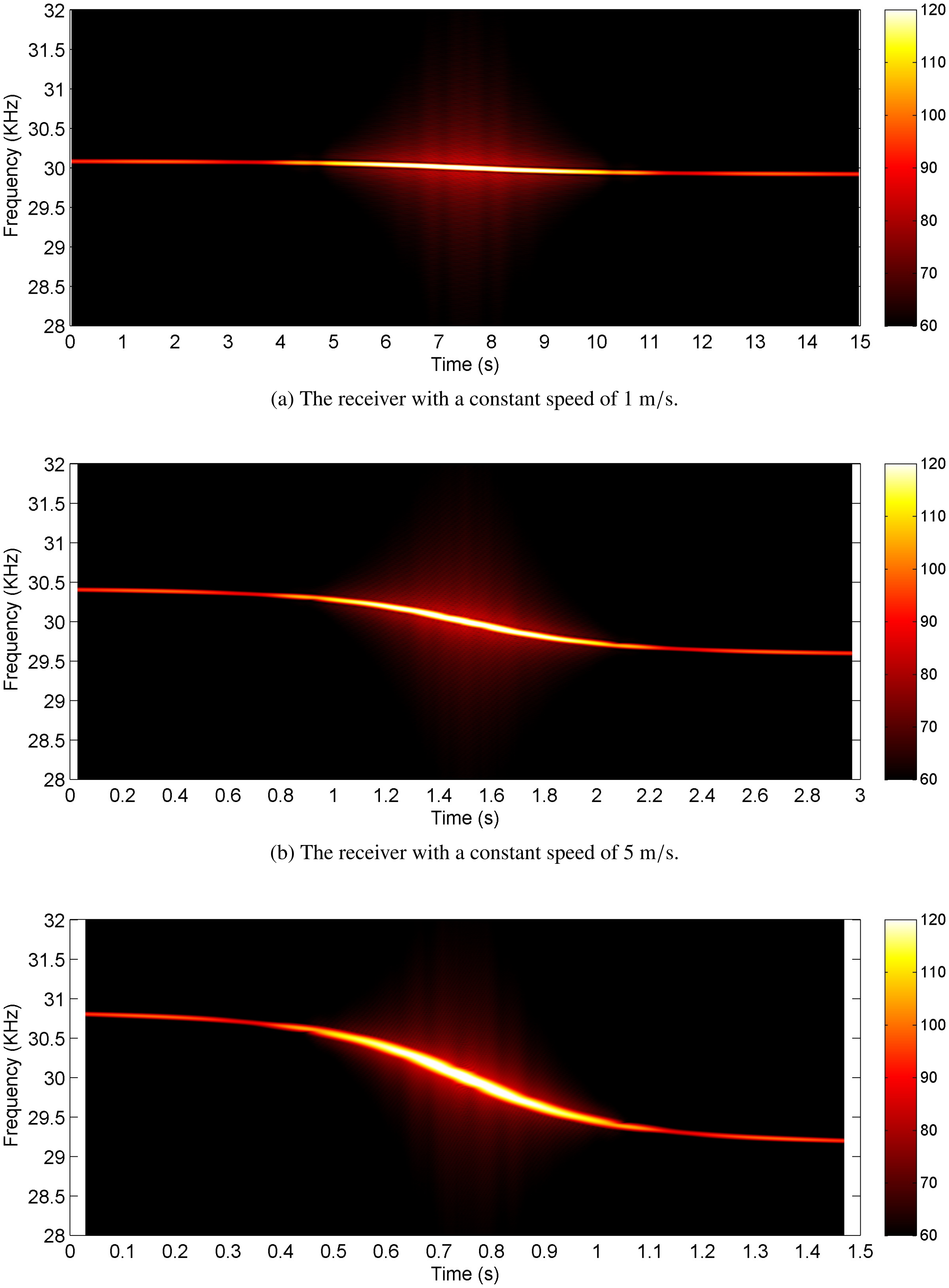


Figure 3.42: SPL produced by the source as a function of the frequency in the receiver direction.

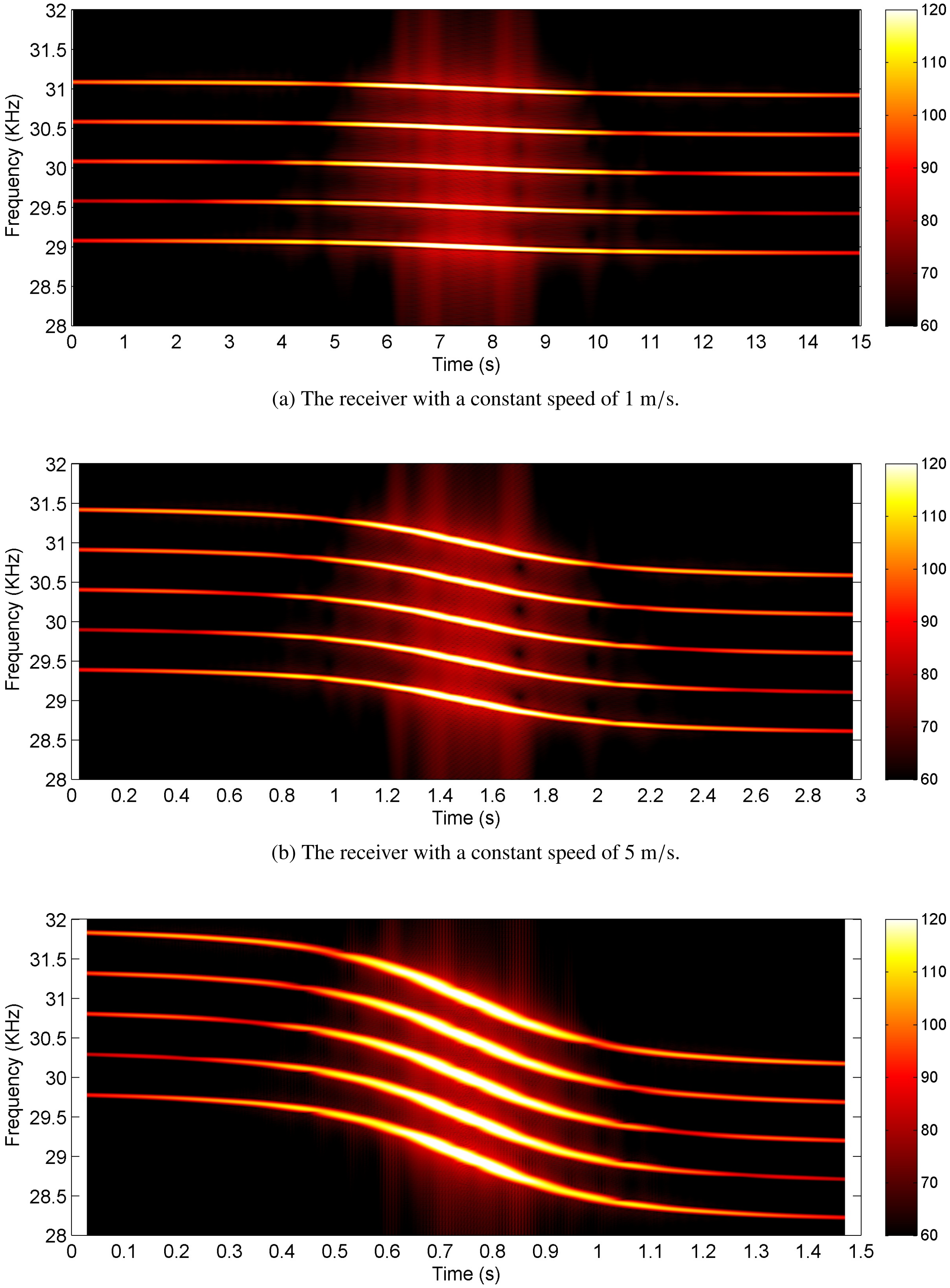


(a) The receiver with a constant speed of 1 m/s.

(b) The receiver with a constant speed of 5 m/s.

(c) The receiver with a constant speed of 10 m/s.

Figure 3.43: Test 1: Doppler effect in a sinusoidal signal with 30 kHz for a receiver with uniform and rectilinear movement for three different speeds.

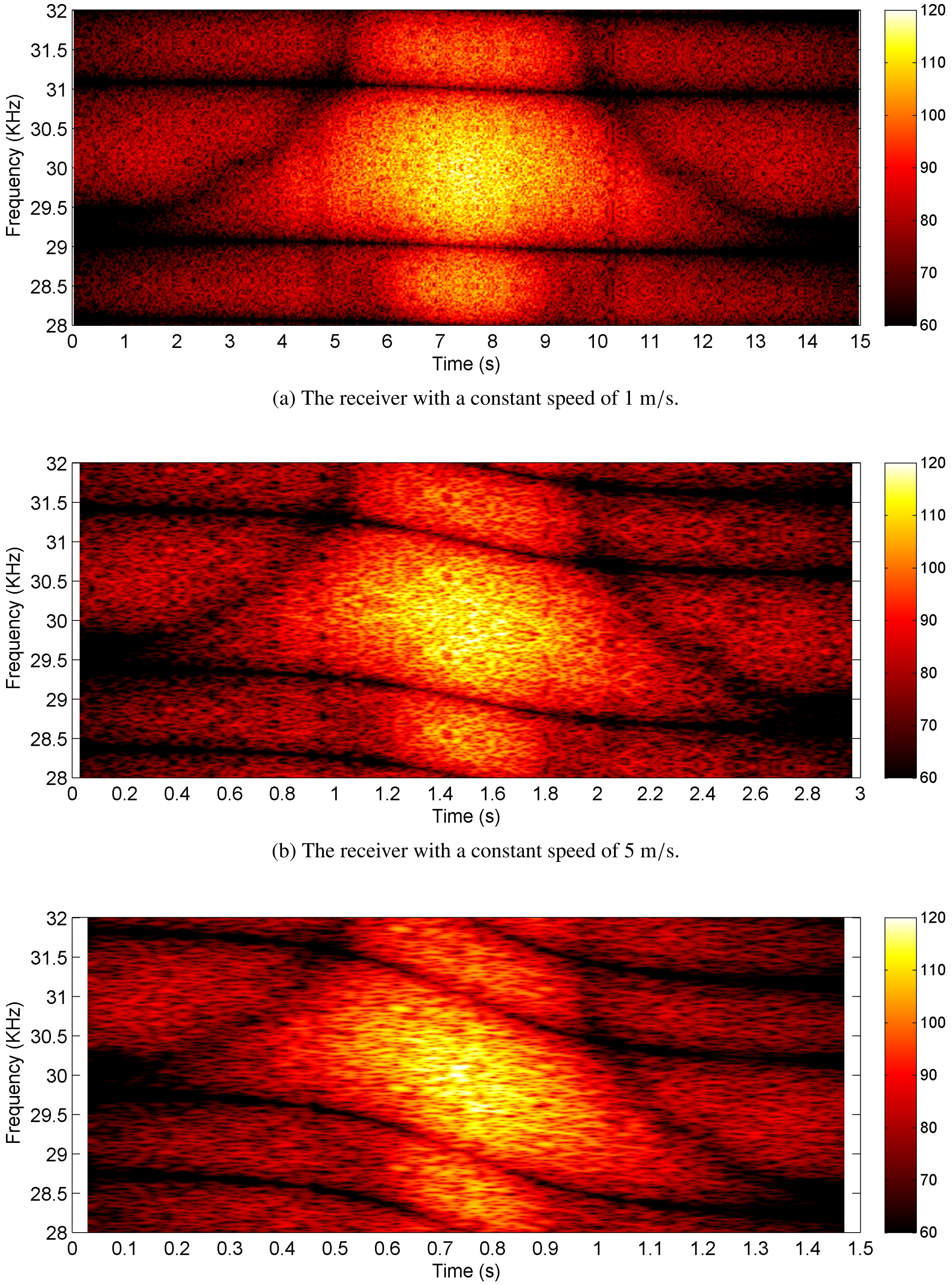


(a) The receiver with a constant speed of 1 m/s.

(b) The receiver with a constant speed of 5 m/s.

(c) The receiver with a constant speed of 10 m/s.

Figure 3.44: Test 2: Doppler effect in a signal composed by 5 sinusoidal signals with frequencies 29, 29.5, 30, 30.5 and 31 kHz for a receiver with uniform and rectilinear movement for three different speeds.



(a) The receiver with a constant speed of 1 m/s.

(b) The receiver with a constant speed of 5 m/s.

(c) The receiver with a constant speed of 10 m/s.

Figure 3.45: Test 3: Doppler effect in 1 Kbps BPSK modulated signal with 30 kHz of central frequency for a receiver with uniform and rectilinear movement for three different speeds.

3.7.2 Uniform and Circular Movement

To test the Doppler effect when the receiver is moving with an uniform and circular movement the simulation environment shown in Figure 3.46 was used. In this environment there are six surfaces with a reflection coefficient of 0.5 (four walls, floor and ceiling), the floor was placed at $z = -1$ and the ceiling at $z = 2$. The receiver performs a circular movement with constant speed of 10 m/s and 2 m of radius. Moreover, to evaluate and understand the reflections impact, the receiver makes two complete turns, which takes around 2.51 s, in the first turn the simulator was configured to consider no reflection and in the second turn the simulator considers a maximum of three successive reflections.

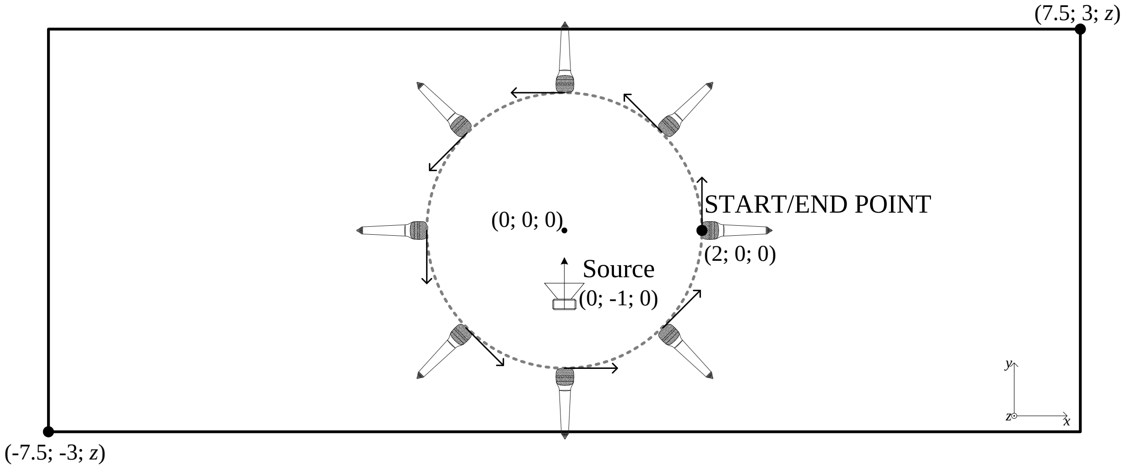


Figure 3.46: Simulation environment to test the Doppler effect for a receiver with uniform and circular movement.

The source was placed at 1 m from the coordinate system origin in (0; -1; 0) and 0º Beam was set to the y direction. The same speaker of the last tests (Figure 3.39) was used. As a consequence of the source position the relative speed will change with time according to Figure 3.47 which presents a speed variation of ±5 m/s.

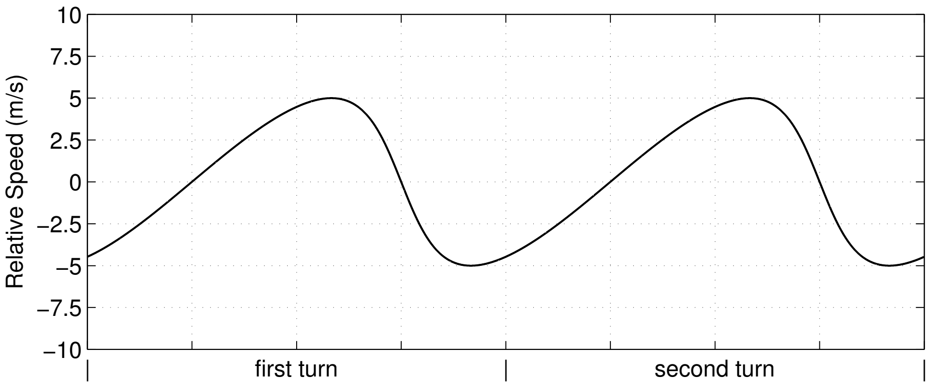


Figure 3.47: Relative speed between the receiver and source for a receiver with uniform and circular movement (moving at 10 m/s and a radius of 2 m).

Figure 3.48 presents the Doppler effect on a single sinusoidal signal, into a signal composed by 5 sinusoidal signals and on BPSK modulated signal. When multiple reflections are introduced, the resultant signal will be a mixture of different sinusoids with different frequencies due to reflected waves arriving to the receiver with different relative speeds. At 10 m/s this mixture can have sinusoids with frequencies shifts of around ±840 Hz for 29 kHz, ±870 Hz for 30 kHz and ±900 Hz for 31 kHz, of maximum Doppler frequency shift, $f_{D\text{Max}}$ (Equation 3.24). This effect can be seen in Figure 3.48a where the signal frequency changes between 29.13 kHz and 30.87 kHz and Figure 3.48b where the signal frequency changes between 28.16 kHz and 31.90 kHz. For communications over this conditions (see Figure 3.48c), due to the extreme mixture it will be difficult to demodulate the received signal.

3.8 Future Work

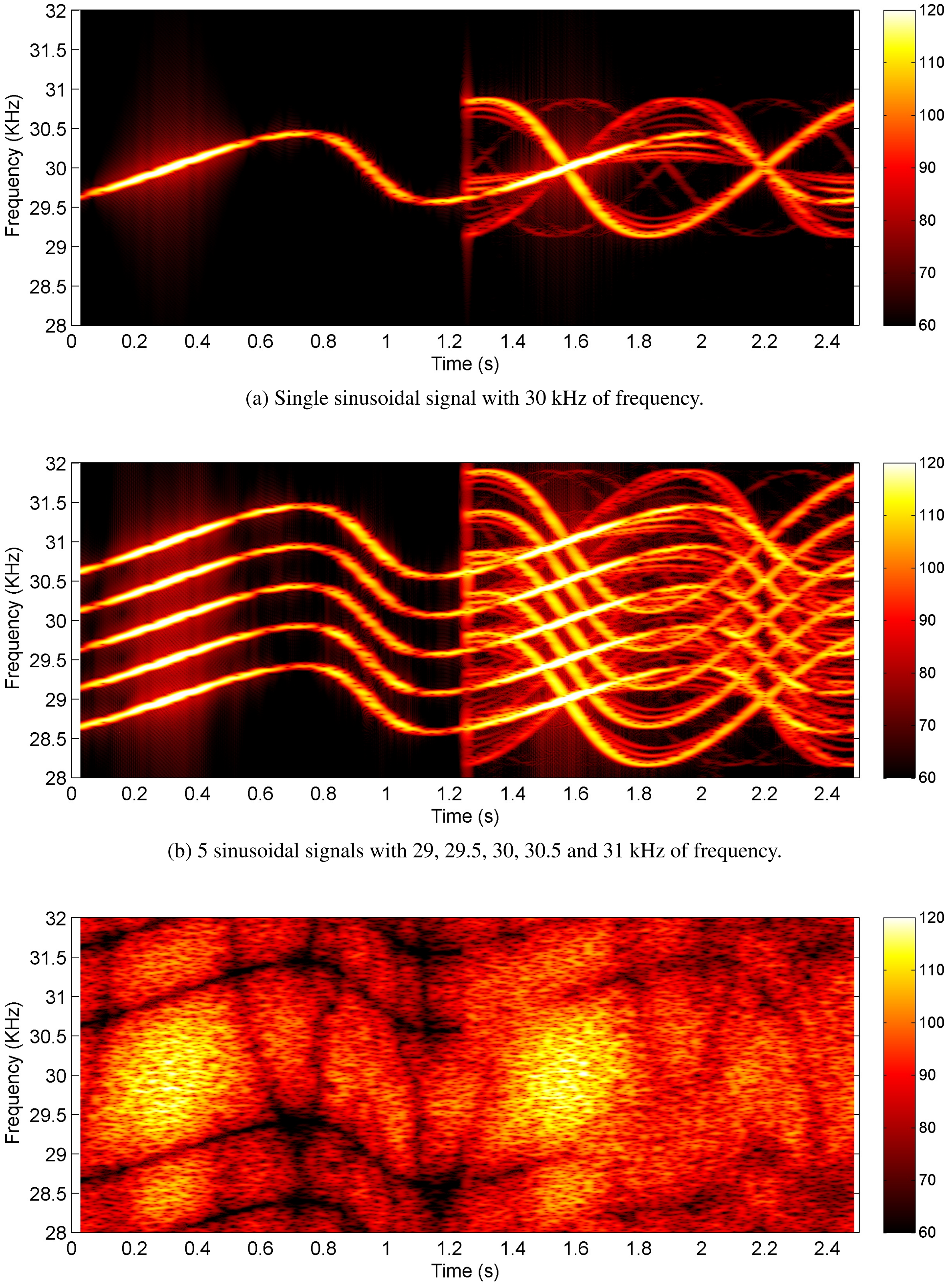
The implemented simulator is already in a mature state with acceptable performance and good accuracy (as will be seen in Chapter 6). However, there are several points that can be improved both in terms of performance and accuracy. In the present simulator the virtual source validation by ray tracing takes more than half of the total simulation time in some simulations and therefore this is a key point to improve the performance. Another important point is the user interface, which forces the user to know how to use Matlab. Therefore some work should be done in this field turning the simulator more user friendly.

In the simulator, the room is implemented by setting all the surfaces coordinates, which is a huge time consumer. Consequently, the simulator integration with other simple and intuitive room design (like SketchUp[7]) will simplify the room implementation process.

From the accuracy point of view, it will be important to take into account the time and space variation of some of the ambient variables, especially room temperature gradients and the wind that can be easily produced by a opened window or air conditioning equipment, another improvement would be to consider the reflection spread due to the surface roughness. Another important point in the accuracy field, especially for low US frequencies is the transmission waves in the room surfaces.

Finally, another very important point for future work is the possibility of porting the implemented simulator to the propagation of sounds in other fluids, with especially attention to water allowing the use of it in underwater applications.

[7] SketchUp from Trimble is one of the most popular and used 3D modeling tools, it was created to be used not only by professionals but also by amateurs with no knowledge in 3D modeling.



(a) Single sinusoidal signal with 30 kHz of frequency.

(b) 5 sinusoidal signals with 29, 29.5, 30, 30.5 and 31 kHz of frequency.

(c) A BPSK modulated signal with 1 Kbps rate and a central frequency of 30 kHz.

Figure 3.48: Doppler effect and reflection impact (in the second half of the time) into three different signals for a receiver with uniform and circular movement.

This page intentionally contains only this sentence.

CHAPTER 4

PULSE DESIGN

This chapter will describe the design of the US pulse that will be used in the proposed location system. This pulse is based on a well-known modulation technique used in telecommunications, the OFDM [51, 52, 48]. Therefore, in Section 4.1 the OFDM communication technique and its advantages and disadvantages are briefly presented. After that, the proposed pulse for TOF measurement and for data communication based on OFDM will be presented in Section 4.2. Thereafter, the pulse detection problem and the data communication problem will be addressed in Sections 4.3 and 4.4 respectively. In both cases their performance will be compared to other usual techniques.

4.1 OFDM Communication

The orthogonal frequency division multiplexing is a method of data transmission that uses multiple carriers at a very low rate. The OFDM is used, nowadays, for data transmission in several digital communication systems such as, wireless networks, mobile communications and digital television broadcasting, among others [51, 52, 48]. To achieve an extremely low rate a large number of overlapped orthogonal carriers closely spaced are used. Moreover, the data is divided among these carriers and modulated with a conventional modulation, such as, phase shift keying (PSK) and quadrature amplitude modulation (QAM), keeping the total bit rate [48]. The main advantage of using OFDM is its robustness to some adverse indoor US channel conditions (see Chapter 3), such as strong multipath and different equalization along the frequency especially for high frequencies, without the need of a complex equalization filter [51, 48]. In addition to this, the low rate allows the use of a guard interval that will prevent the inter-symbolic-interference (ISI) reducing the bit error rate (BER) [51, 48].

As a consequence of its advantages OFDM will be used as the basis for both, data transmission and pulse synchronization. This technique has already being used in ultrasonic underwater communications [53, 54]. This choice was based on the fact that the OFDM is a very flexible modulation technique, very robust to multipath and, as it will be seen, channel equalization could be avoided. Moreover, it is very sensitive to synchronization, which may be an advantage when the application requires the measurement of the TOF [55].

In OFDM communication, the information is divided in N blocks and each value of this block is sent using a different carrier. To avoid ISI the OFDM uses Fourier transform carriers that are orthogonal to each other [48]. Therefore, the OFDM communication global system can be viewed as a modulation with multiple carriers [48]. In order to speed up the process of modulation and demodulation the fast Fourier transform (FFT) and the inverse fast Fourier transform (IFFT) can be used, resulting in considerable efficiency improvement on the implementation of the system. Figure 4.1 presents the block diagram of the OFDM communication.

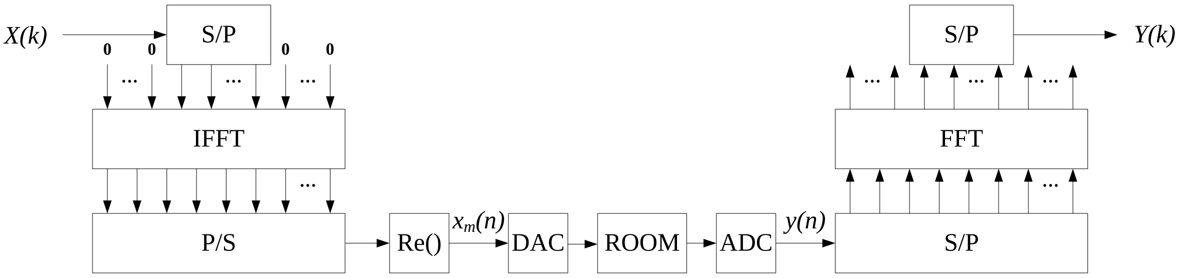
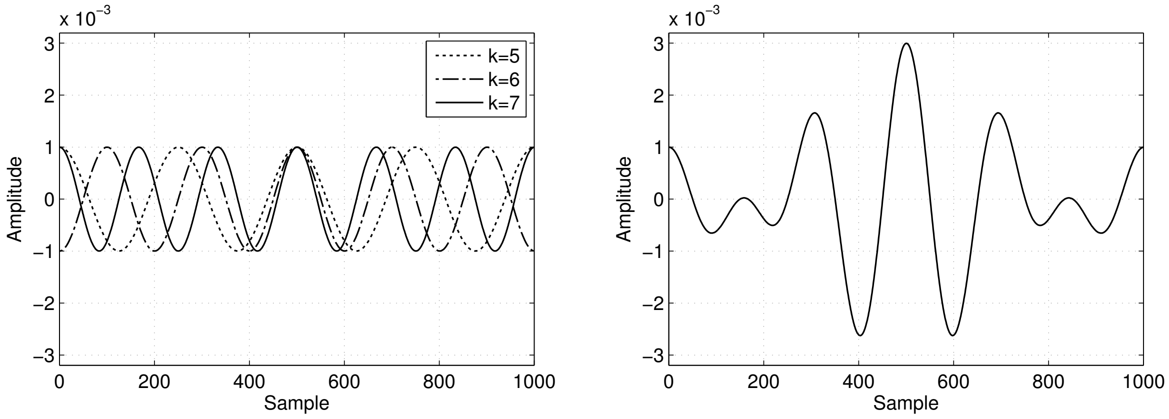


Figure 4.1: Block diagram of the OFDM communication system.

The system modulates each carrier of the IFFT directly with some classic single carrier modulation like PSK or QAM. The system only modulates the carriers of the transmission system band (e.g. from 25 kHz to 50 kHz). Moreover, the system does not use the Nyquist frequency to keep some energy outside the carriers because of the poor spectral decay [48].

To exemplify the OFDM system operations we are going to consider the transmission of an OFDM pulse with three bits (0 1 0) of information, and we chose BPSK to modulate the information. Therefore, the resultant symbols from that particular information will be (1 -1 1). Moreover, for these three symbols three carriers must be used, we choose carrier 5, 6 and 7 of an IFFT block with 1001 carriers. Figure 4.2 presents the three chosen carriers modulated with BPSK and the resultant OFDM pulse. Nevertheless, Figure 4.3 shows the signal for the same example but the symbols information is different (1 -1 -1). As can be seen from Figures 4.2b and 4.3b different combination of carriers information creates different pulse shapes and pulse amplitudes [48]. Which could result into a pulse with a huge amplitude compared to others with the same energy, reducing the transmission efficiency [48].



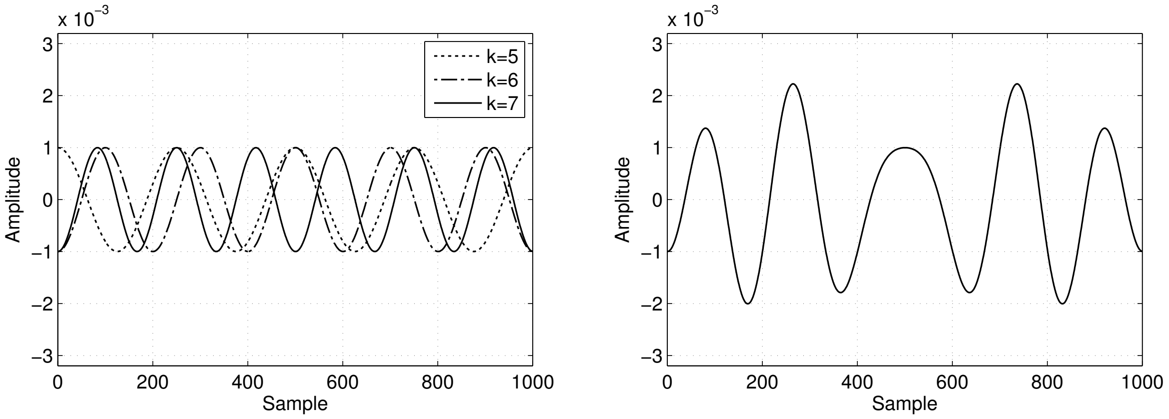
(a) The 3 carriers modulated with BPSK.

(b) The resultant pulse.

Figure 4.2: OFDM example 1: carriers 5, 6 and 7 modulated with the data: (1 -1 1) for an IFFT with 1001 carriers.

4.2 Frame Prototype

For asynchronous data transmission using OFDM pulses it is proposed one architecture based on two concatenated OFDM pulses, one for time synchronization (e.g. TOF measurement) and an-



(a) The 3 carriers modulated with BPSK.

(b) The resultant pulse.

Figure 4.3: OFDM example 2: carriers 5, 6 and 7 modulated with the data: (1 -1 -1) for an IFFT with 1001 carriers.

other for some data information transmission (e.g. source identification/position). The group of the synchronization pulse and the data information pulse will be called frame. Figure 4.4 presents the proposed frame prototype, as mentioned before, there are two different main pulses in the frame: the *OFDM Sync.* and the *OFDM Data*. The first pulse is detected by the receiver by a matched filter and used for synchronization purposes. It will also be used to demodulate the second pulse, the *OFDM Data*, by a differential demodulation scheme [49, 50]. This method was chosen manly because the OFDM pulse is robust to environments with multipath and the differential demodulation is robust to some possible error in pulse synchronization, as will be seen later on.

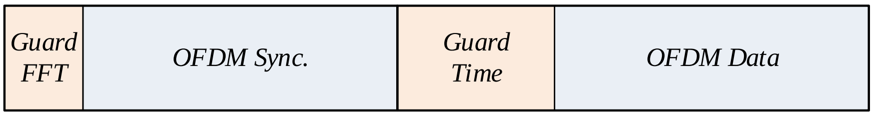


Figure 4.4: Block diagram of the frame prototype.

The *Guard FFT*, presented in Figure 4.4 is a cyclic extension to protect the demodulation process due to some time synchronization error. The *Guard Time* is a cyclic extension of the OFDM data pulse to avoid the ISI caused by the room impulse response [48].

4.2.1 OFDM Frame parameters

To choose the best frame parameters it is necessary to know the limitations of the system that will be used to transmit the information, the limitations come mainly from room conditions and the US wave propagation.

From a detection point of view as longer the pulse is the better will be the detection (for the same amplitude the energy increases with the length of the signal) [55]. On the other hand, the length of the signal is inversely proportional to the distance of two adjacent carriers and consequently to the bandwidth of the resultant signal. So in the first analysis a huge pulse will be the best, however it will introduce a problem, the carriers will be very near and a little relative speed between the source and the receiver produce a catastrophic change in the carriers position due to the Doppler effect [48]. Therefore an up and a lower bounds to the pulse size are presented:

$$\frac{N_c}{B} \leq T_p \leq \frac{c + v_r}{2 v_r f_{max}}, \tag{4.1}$$

where T_p is the time length of the pulse, B is the maximum bandwidth for the resultant pulse, v_r is the maximum allowed relativity speed between the source and the receiver and f_{max} is the maximum frequency of the resultant desired pulse. Note that, in this equation the maximum Doppler shift allowed corresponds to half of the distance between the adjacent carriers.

The *Guard Time* must be greater or equal than the *Guard FFT* and greater than the minimum time to avoid inter-symbolic interference T_{ISI}. Normally, the minimum time to avoid the inter-symbolic interference is bigger than the *Guard FFT*. Mathematically the lower bound to *Guard Time* is:

$$T_G \geq \max \left\{ T_{G_{FFT}}, T_{ISI} \right\}. \tag{4.2}$$

Considering that a wave that arrives to the receiver some time after the direct wave Δt, needs to travel more $\Delta d = {}^{\Delta t}/_{c}$ distance, and that for the omni-directional transducers it will be attenuated according to Equation 3.16:

$$A_r(f) = A_T(d_{sr} + \Delta d, f) - A_T(d_{sr}, f) = 20 \log \left(\frac{d_{sr}}{d_{sr} + \Delta d} \right) - \Delta d \times \alpha(f), \tag{4.3}$$

where d_r is the distance between the source and the receiver. The minimum attenuation results from the maximum distance between the source and the receiver d_{max} and the minimum absorption coefficient in air α_{min} for the signal bandwidth:

$$A_{min} = 20 \log \left(\frac{d_{max}}{d_{max} + \Delta d} \right) - \Delta d \times \alpha_{min}. \tag{4.4}$$

If β is the maximum ratio between the reflection wave amplitude and the direct wave amplitude that does not produce interference, then, the minimum to avoid the inter-symbolic interference can be approximated by the solution of the equation:

$$20 \log \left(d_{max} + c T_{ISI} \right) + \alpha_{min} c T_{ISI} = 20 \log \left(\frac{d_{max}}{\beta} \right). \tag{4.5}$$

To give an example we chose a common situation, when the maximum relative speed between the source and the receiver is less than 0.01 m/s. The system operates in the range from 20 kHz to 40 kHz and we consider that 1000 carriers will be needed for data communication. From Equation 4.1 we can estimate the limits for the length of the pulses:

$$100 \text{ ms} \leq T_p \leq 428 \text{ ms}. \tag{4.6}$$

Thus, we choose T_p of 100 ms, the lower bound. With $T_p = 100$ ms the distance between two adjacent carriers is 10 Hz. Allowing the needed throughput and a maximum relative speed between the source and the receiver of 0.04 m/s.

Knowing that the maximum error in the synchronization process is 5 ms. Therefore 10 ms was chosen for the $T_{G_{FFT}}$ to allow a low accuracy in the received instant.

Finally, to compute the *Guard Time* we consider that the maximum distance between the source and receiver is 7 m, the absorption coefficient in air is 1.2 dB/m and a reflected wave with amplitude of one tenth ($\beta = 0.1$) of the direct wave does not produce a relevant inter-symbolic-interference. With these values and using (4.5) the T_{ISI} is 30 ms so we are going to use 30 ms of *Guard Time*.

Under these conditions and using a sampling frequency of 100 kHz Figure 4.5 presents one possible frame.

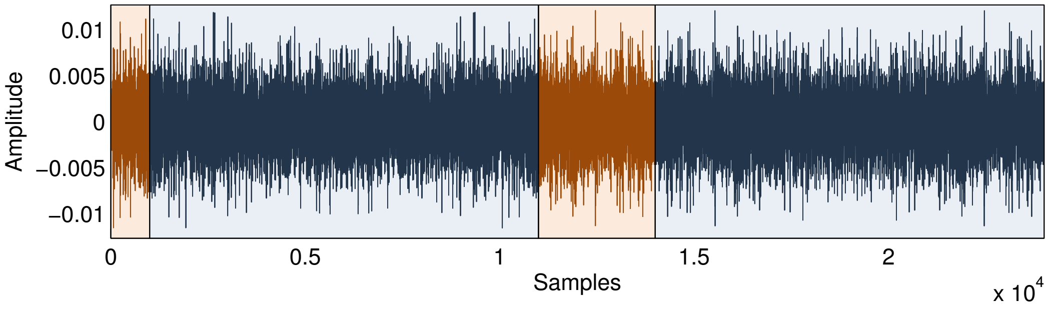


Figure 4.5: Example of one possible frame.

4.3 Pulse Detection

The detection theory was well developed in the radar field, however, nowadays it is present in several fields such as, telecommunications, image processing and medicine, among others [56].

To measure the TOF in an US location system it is important to precisely detect when the sent pulse is received in order to estimate the distance between the transmitter and the receiver. Therefore, the challenge will be to decide if the received signal corresponds to the sent pulse mixed with noise or only to noise. Moreover, the decision process depends on the amount of background noise. It will be easier to determine if the pulse is received when the noise energy is much smaller than the signal energy. Otherwise, if the noise energy increases considerable the resultant decision could be wrong.

Before analyzing in detail the detection process, it is important to define the received signal and make some assumptions. Therefore, the received signal $y(n)$ can be written as:

$$y(n) = w(n) + A \times s(n - n_\tau), \tag{4.7}$$

where $w(n)$ is white Gaussian noise with zero mean and σ^2 variance, $w(n) \in \mathcal{N}(0, \sigma^2)$. The amplitude attenuation due to propagation losses and signal amplification is A and n_τ represents the first delay due to TOF. Moreover, without loss of generalization we are going to consider this delay, n_τ, 0, in order to reduce the mathematical analysis complexity. Nevertheless, it is important to keep in mind that the analysis should always be valid and the only difference is the time reference that must be changed from zero to n_τ.

4.3.1 Pulse Detection with Matched Filter

In order to detect the presence and position of a known signal in an incoming signal a matched filter (MF) can be used. The MF correlates the incoming signal with the known signal, which is equivalent to filter the signal with a finite impulse response (FIR) filter where the coefficients are the conjugated time-inverted version of the known signal. Moreover, it was proved that the MF is the optimal filter to maximize the signal to noise ratio (SNR) over white Gaussian noise [55]. Therefore, it will be the best way to detect a known pulse in an incoming signal.

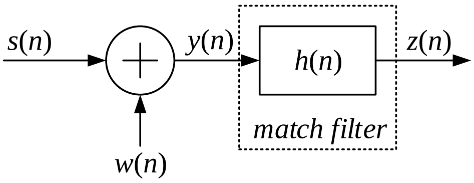


Figure 4.6: Matched Filter.

The matched filter is schematically represented by a filter $h(n)$, Figure 4.6, where the coefficients $h(n)$ are given by:

$$h(n) = s^*(N - n - 1), \tag{4.8}$$

where $()^*$ represents the complex conjugate. Therefore, the output $z(n)$ is given by:

$$z(n) = \sum_{m=0}^{N-1} h(m)y(n - m). \tag{4.9}$$

Using (4.8) the output of the MF becomes:

$$z(n) = \sum_{k=0}^{N-1} s^*(N - k - 1)y(n - k). \tag{4.10}$$

4.3.2 Pulse Detection with DFT

Apart the conventional time domain MF implementation, the DFT followed by a small scalar product can also be used to implement the MF. Figure 4.7 presents the technique, where the incoming signal, $y(n)$ (pulse to detect plus noise) enters into the serial to parallel converter which acts as a Buffer Delay with the same size of the pulse $s(n)$. Therefore, the information in the buffer is converted to the frequency domain. Note that, until now the receiver is the same as the typical OFDM receiver depicted in Figure 4.1. Moreover, due to the OFDM properties $S(k)$ (Fourier transform of $s(n)$) is only different from zero in the carriers with information so the system only needs to perform the dot product in that carriers.

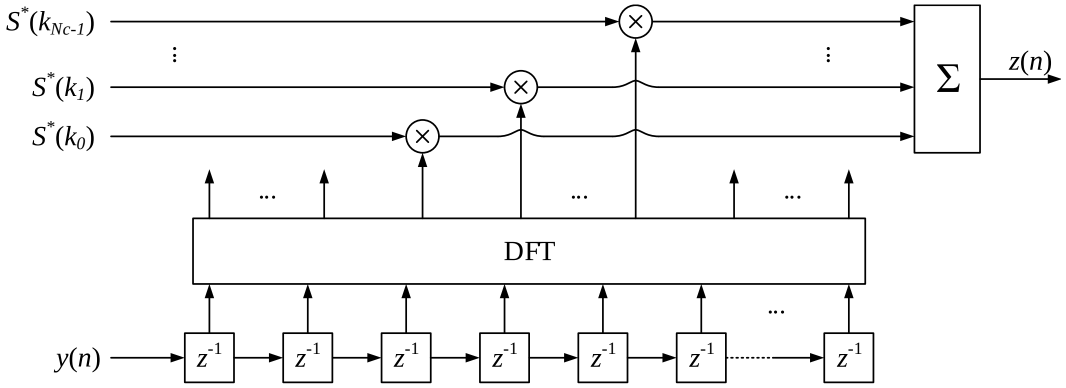


Figure 4.7: Pulse detection with DFT.

Before proceeding with the proof it is important to mathematically define the DFT and IDFT:

$$Y(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} y(n) W^{-nk} \tag{4.11}$$

$$y(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} Y(k) W^{nk} \tag{4.12}$$

where $W = e^{j\frac{2\pi}{N}}$. Therefore, the output $z(n)$ of the detector using the DFT can be written as:

$$z(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S^*(k) \sum_{m=0}^{N-1} y(m+n-N+1) W^{-mk}. \tag{4.13}$$

By changing the terms order it is possible to obtain the following equation:

$$z(n) = \sum_{m=0}^{N-1} y(m+n-N+1) \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S(k) W^{mk} \right)^*, \tag{4.14}$$

where the term between brackets is IDFT of $s(n)$:

$$z(n) = \sum_{m=0}^{N-1} y(m+n-N+1) s(m)^*. \tag{4.15}$$

Which is equivalent to:

$$z(n) = \sum_{k=0}^{N-1} s^*(N-k-1) y(n-k). \tag{4.16}$$

As can be seen, Equation 4.16 is equal to Equation 4.10. Therefore the OFDM receiver can be used to implement part of the matched filter. Moreover, in the proposed technique the scalar product needs to compute N_c products. On the other hand, the MF technique needs to have a filter with pulse size samples (N products). So the choice between one of these techniques depends on the pulse size and the number of carriers. Nevertheless, the number of carriers tends to be much smaller than the pulse size therefore the proposed technique proves to be faster, nevertheless, the traditional technique to implement MF filter can also be used.

4.3.3 Pulse Detection with Noise

One possibility to check if the pulse is present is to compare the MF output, $z(n)$, with a threshold [56], where the system only considers that it is in the presence of the pulse if the MF output exceeds that threshold. Therefore, the threshold must be chosen in a way that a very weak pulse must be detected and the noise does not produce false alarms. In this way, it is very important to keep the probability of false alarm, which is the probability of detecting the pulse when there is only noise in the input, very small and the probability of detection very high. To evaluate these probabilities we must consider two competing hypotheses [56]:

$$\begin{cases} \mathcal{H}_0 & : \quad y(n) = w(n) \\ \mathcal{H}_1 & : \quad y(n) = w(n) + As(n) \end{cases} \tag{4.17}$$

which is equivalent to:

$$\begin{cases} \mathcal{H}_0 & : \quad y(n) = w(n) + As(n) \quad , A = 0 \\ \mathcal{H}_1 & : \quad y(n) = w(n) + As(n) \quad , A \neq 0 \end{cases} \tag{4.18}$$

where $\mathcal{H}_0$ is called the *null hypothesis*, $\mathcal{H}_1$ the *alternative hypothesis*. The system chooses the hypothesis by the magnitude of the signal $z(n)$. Therefore, if the magnitude of $z(n)$ is greater than a given threshold γ the system decides $\mathcal{H}_1$ otherwise the system decides $\mathcal{H}_0$. The block diagram for this procedure is shown in Figure 4.8.

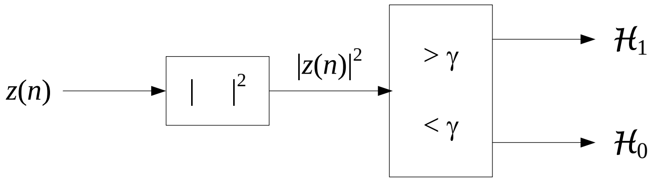


Figure 4.8: Detection decision block.

The output of the first block, before the comparators, is given by:

$$|z(n)|^2 = \frac{1}{N}\left|\sum_{k=0}^{N-1} S^*(k) \sum_{m=0}^{N-1} y(m+n-N+1)W^{-mk}\right|^2. \tag{4.19}$$

Using (4.18):

$$|z(n)|^2 = \frac{1}{N}\left|\sum_{k=0}^{N-1} S^*(k)\left(\sum_{m=0}^{N-1} w(m+n-N+1)W^{-mk} + A\sum_{m=0}^{N-1} s(m+n-N+1)W^{-mk}\right)\right|^2, \tag{4.20}$$

The signal $S(k)$ is only different from zero in a set of carriers of the OFDM pulse, $k \in (k_0, k_1, ..., k_{N_c-1})$, due to the differential demodulation the carriers amplitude must be the same. And without loss of generalization we will consider that $|S(k)| = 1$ for $k \in (k_0, k_1, ..., k_{N_c-1})$.

So we can simplify the Equation 4.20 to:

$$|z(n)|^2 = \frac{1}{N}\left|\sum_{k=k_0}^{k_{N_c-1}}\sum_{m=0}^{N-1} w(m+n-N+1)W^{-mk-\theta_k} + A\sum_{k=k_0}^{k_{N_c-1}} S^*(k)\sum_{m=0}^{N-1} s(m+n-N+1)W^{-mk}\right|^2. \quad (4.21)$$

Part of the first term can be seen as the DFT of the delayed noise:

$$|z(n)|^2 = \frac{1}{N}\left|\sum_{k=k_0}^{k_{N_c-1}} \sqrt{\frac{N\sigma^2}{2}}\Big(U(k) + jV(k)\Big) + A\sum_{k=k_0}^{k_{N_c-1}} S^*(k)\sum_{m=0}^{N-1} s(m+n-N+1)W^{-mk}\right|^2, \quad (4.22)$$

where $U(k)$ and $V(k)$ for $k = 1, 2, ..., N/2 - 1$ are random independent and Gaussian variables with zero mean and unity variance [57]. The equation can be simplified by summing the N_c random independent variables:

$$|z(n)|^2 = \frac{1}{N}\left|\sqrt{\frac{NN_c\sigma^2}{2}}\Big(U(k) + jV(k)\Big) + A\sum_{k=k_0}^{k_{N_c-1}} S^*(k)\sum_{m=0}^{N-1} s(m+n-N+1)W^{-mk}\right|^2. \quad (4.23)$$

In order to evaluate the probabilities of detection and false alarm it is important to separate the equation in real and imaginary part:

$$|z(n)|^2 = \frac{N_c\sigma^2}{2}\left[\Big(U(k) + \mu_U(n)\Big)^2 + \Big(V(k) + \mu_V(n)\Big)^2\right], \quad (4.24)$$

where $\mu_U(n)$ and $\mu_V(n)$ can be seen as the mean of these two random variables, moreover, they depend on the pulse and they are only different from zero when the pulse is present, and they are mathematically expressed as:

$$\mu_U(n) = \frac{A}{\sigma}\sqrt{\frac{2}{N_c}}\mathbf{Re}\left\{\sum_{k=k_0}^{k_{N_c-1}} S^*(k)\frac{1}{\sqrt{N}}\sum_{m=0}^{N-1} s(m+n-N+1)W^{-mk}\right\}; \quad (4.25)$$

$$\mu_V(n) = \frac{A}{\sigma}\sqrt{\frac{2}{N_c}}\mathbf{Im}\left\{\sum_{k=k_0}^{k_{N_c-1}} S^*(k)\frac{1}{\sqrt{N}}\sum_{m=0}^{N-1} s(m+n-N+1)W^{-mk}\right\}; \quad (4.26)$$

where, **Re**{**.**} and **Im**{**.**} represents the real and imaginary part of the argument respectively.

Analyzing Equation 4.24 it can be seen that the two quadratic terms are the sum of two quadratic random independent and Gaussian variables, $U(k)$ and $V(k)$, with unity variance and mean $\mu_U(n)$ and $\mu_V(n)$ respectively, which results into a Chi-Squared distribution with two degrees of freedom (χ_2^2) [57], where the noncentrality parameter, $\lambda(n)$, is given by:

$$\lambda(n) = \mu_U(n)^2 + \mu_V(n)^2 \quad (4.27)$$

From (4.25), (4.26) and (4.27) we can get two important conclusions: first, the value of N does not change the magnitude of λ, second, λ is proportional to A/σ consequently λ is proportional to the SNR. Note that the variable A is not the amplitude of the pulse, A is a scale factor. The amplitude of the pulse s changes with the size and the number of carriers in the pulse. So for the same A/σ we get different SNRs with different pulse lengths or carriers. For the particular case when $n = N - 1$, (last sample of the pulse $s(n)$), $\mu_U(N-1)$ and $\mu_V(N-1)$ becomes:

$$\mu_U(N-1) = \frac{A}{\sigma}\sqrt{\frac{2}{N_c}}\mathbf{Re}\left\{\sum_{k=k_0}^{k_{N_c-1}} S^*(k)S(k)\right\} = \frac{A}{\sigma}\sqrt{2N_c}; \tag{4.28}$$

$$\mu_V(N-1) = \frac{A}{\sigma}\sqrt{\frac{2}{N_c}}\mathbf{Im}\left\{\sum_{k=k_0}^{k_{N_c-1}} S^*(k)S(k)\right\} = 0. \tag{4.29}$$

As a consequence of this $\lambda(N-1)$ takes the value:

$$\lambda(N-1) = \frac{2N_cA^2}{\sigma^2}. \tag{4.30}$$

The signal energy is given by A^2N_c and the noise energy by $\sigma^2 N$, therefore, $\lambda(N-1)$ can also be written as function of the SNR and N:

$$\lambda(N-1) = 2N \times SNR. \tag{4.31}$$

With these results it would be easy to compute the probability of false alarm (P_{fa}) and the probability of detection (P_d) for any instant n.

Probability of False Alarm

The P_{fa} can be seen as the probability of choosing $\mathcal{H}_1$ when the system is in the presence of $\mathcal{H}_0$ i.e. the system believes to be in the presence of the pulse but it only has noise in the input ($\mu_U(n)$ and $\mu_V(n)$ are equal to zero). The probability of false alarm can be defined as:

$$P_{fa} = Pr\left\{|z(n)|^2 > \gamma; \mathcal{H}_0\right\}. \tag{4.32}$$

Replacing $|z(n)|^2$ we obtain:

$$P_{fa} = Pr\left\{\frac{N_c\sigma^2}{2}\left[(U(k)+\mu_U(n))^2 + (V(k)+\mu_V(n))^2\right] > \gamma; \mathcal{H}_0\right\}, \tag{4.33}$$

which is the same as:

$$P_{fa} = Pr\left\{U(k)^2 + V(k)^2 > \frac{2\gamma}{N_c\sigma^2}\right\}. \tag{4.34}$$

Moreover the right-tail probability for a χ_2^2 with a null central parameter ($\lambda = 0$) is given by [57]:

$$Q_{\chi_2'^2}(x) = e^{-0.5x}. \tag{4.35}$$

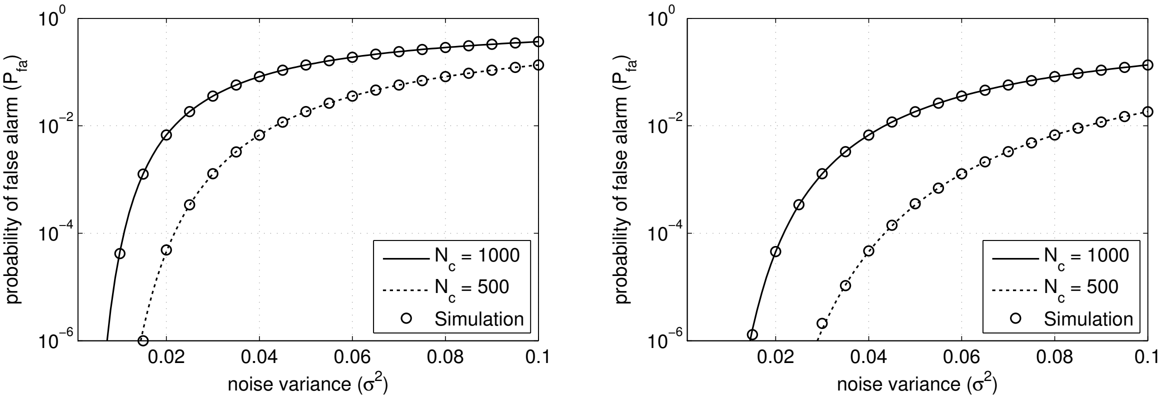
Therefore the probability of false alarm can be expressed as:

$$P_{fa}Q_{\chi_2^2}\left(\frac{2\gamma}{N_c\sigma^2}\right) = e^{-\frac{\gamma}{N_c\sigma^2}}. \tag{4.36}$$

The most important part is how to choose the threshold to ensure certain probability of false alarm:

$$\gamma = -N_c \sigma^2 \ln(P_{fa}), \tag{4.37}$$

from this equation it can be seen that the threshold value is proportional to the number of carriers and the noise variance. For example: if a system that uses a thousand carriers is in the presence of white noise with variance $\sigma^2 = 0.02$ then, to ensure a probability of false alarm of 1×10^{-4}, the threshold must be at least 184.2. Figure 4.9 presents the probability of false alarm as a function of the noise variance for a system with 1000 carriers and another one with 500 carriers, using a threshold of 100 and 200 in both systems. Moreover, the same graph presents also the simulation results for 10 million trials for white noise with variance from 0.005 to 0.1 with steps of 0.005. As can be seen the simulation results match very well with the expected values.



(a) Using a threshold of $\gamma = 100$.

(b) Using a threshold of $\gamma = 200$.

Figure 4.9: Probability of false alarm for 500 and 1000 carriers and a threshold value of 100 and 200. The lines are the mathematical expression and the marks are the probability evaluated for 10 million of trials.

Probability of Detection

The probability of detection can be seen as the probability of choosing $\mathcal{H}_1$ when the system is in the presence of the pulse i.e. the system considers that it is receiving a pulse and it really is. The probability of detection can be defined as:

$$P_d = Pr\Big\{|y(n)|^2 > \gamma; \mathcal{H}_1\Big\}. \tag{4.38}$$

Doing the same for P_d that it was done for P_{fa}:

$$P_d = Pr\left\{(U(k) + \mu_U(n))^2 + (V(k) + \mu_V(n))^2 > \frac{2\gamma}{N_c\sigma^2}\right\}. \tag{4.39}$$

Replacing the γ by the expression in (4.37) it is possible to write the probability of detection as a function of the probability of false alarm:

$$P_d = Pr\left\{(U(k) + \mu_U(n))^2 + (V(k) + \mu_V(n))^2 > -2\ln(P_{fa})\right\}, \tag{4.40}$$

which is given by:

$$P_d = Q_{\chi_2^2(\lambda(n))}(-2\ln(P_{fa})). \tag{4.41}$$

Moreover the right-tail probability for a χ_2^2 with a central parameter given by Equation 4.27 is more complex than the null central parameter however it can be easily computed numerically with the algorithm proposed in [56].

To exemplify what happens with the probability of detection over time, consider the follow example: a 5000 samples pulse with 1000 carriers randomly chosen between 1 and -1 and at the receiver the threshold was set to ensure a probability of false alarm of 10^{-6}. Two different signal to noise ratios (0 and -20 dB) were considered for this example.

Figure 4.10 presents the output of the matched filter without the presence of noise and the threshold considering signal to noise ratio of 0. On the other hand, Figure 4.11 presents the same example but now with a threshold considering a signal to noise ratio of -20 dB.

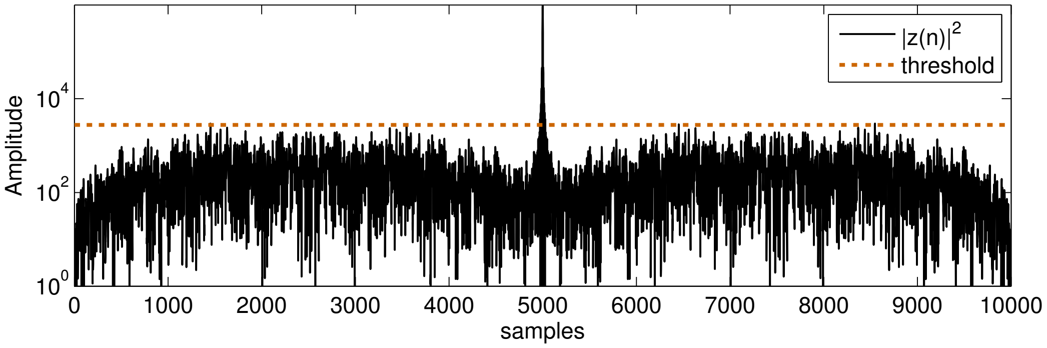


Figure 4.10: Matched filter output for a SNR of 0 dB and the threshold value correspondent to a probability of false alarm of 10^{-6}.

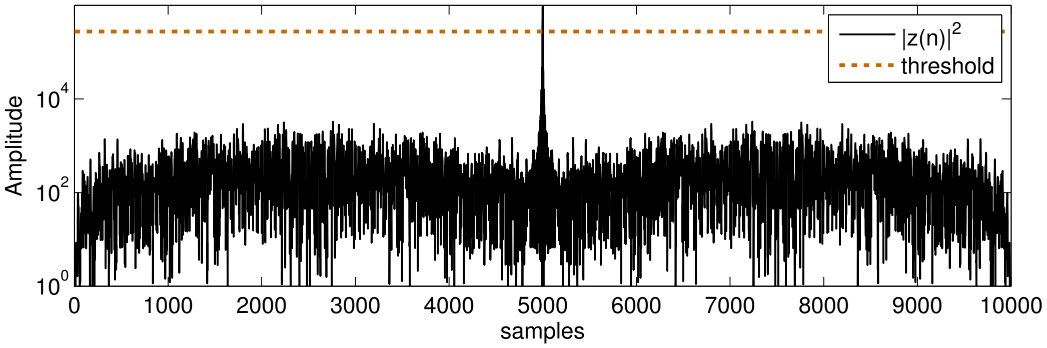
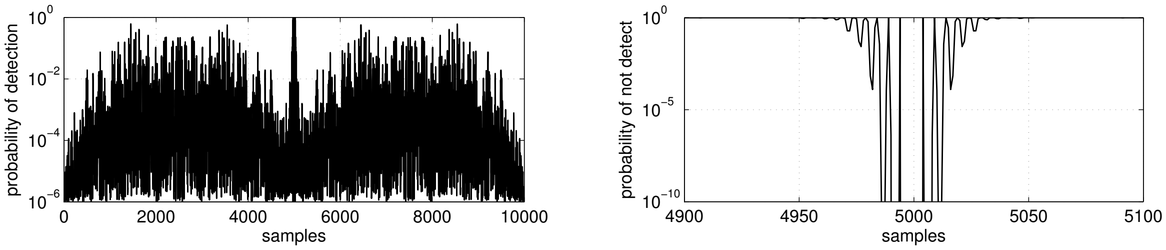


Figure 4.11: Matched filter output for a SNR of -20 dB and the threshold value corresponding to a probability of false alarm of 10^{-6}.

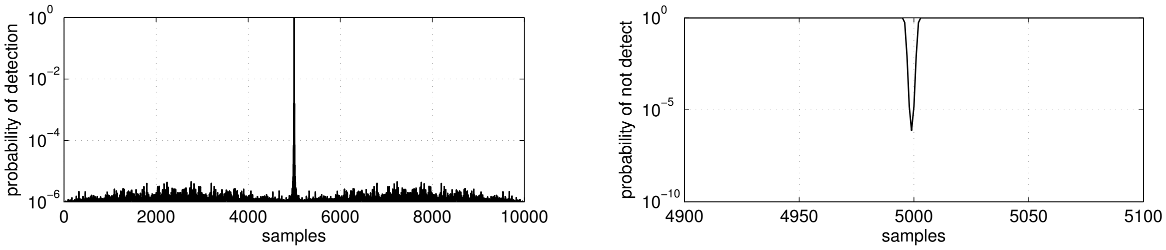
As can be seen, in both cases, the pulse can be easily detected with this technique. However, for the 0 dB SNR case it is almost impossible not to detect the pulse compared to the -20 dB case. Moreover, for the 0 dB case there is some probability of detect the pulse before the matched filter output reach its maximum due to the fact that the threshold is very near to the amplitude of the matched filter output. Therefore, a small amount of noise can make the matched filter output surpass the threshold leading to the pulse detection. On the other hand, for the -20 dB case it is almost impossible to detect the pulse before it is completely received due to the difference between the matched filter output amplitude and the threshold value. These statements can be observed in the probability of detection and the probability of not detecting the pulse presented in Figures 4.12 and 4.13 for a signal to noise ratio of 0 dB and -20 dB respectively, where the probability of not detecting the pulse decreases from almost 0 to 7.5×10^{-7}.



(a) Probability of pulse detection.

(b) Probability of not detecting the pulse.

Figure 4.12: Probability of detecting and not detecting the pulse for a signal to noise ratio of 0 dB.



(a) Probability of pulse detection.

(b) Probability of not detect the pulse.

Figure 4.13: Probability of detect and not detect the pulse for a signal to noise ratio of -20 dB.

From (4.27) and (4.41) it can be seen that for a predetermined probability of false alarm the probability of detection only depends on the signal to noise ratio and the pulse size. Therefore, the probability of detection can be improved by increasing the pulse size or the signal to noise ratio. As a direct consequence of this and as we saw in Chapter 3 the signal energy decreases with the distance to the ultrasonic source, therefore the probability of detection decreases with the distance to the source.

Figure 4.14 presents the probability of detection for the last pulse sample ($N - 1$) as a function of the distance to the source for a pulse around 50 kHz with the same characteristics of the previous example ($N = 5000$, $N_c = 1000$ and $P_{fa} = 10^{-6}$) at typical room conditions (20ºC of temperature, 50% of relative humidity and 1 atm of pressure). Moreover in this example it was considered that the source produce at 1 m from it three different signal to noise ratios of: 20, 40 and 60 dB. As we can see the source range increases as the signal to noise ratio increases. Moreover, at the limit range of the source the probability of detection decreases considerable in a few centimeters. Using the 20 dB case as an example the probability of detection changes from 0.99 at 13.1 m to 0.1 at 15.7 m, which means that at 13 m the pulse is detected with high probability but after 2.5 m it is unlikely to detect the pulse.

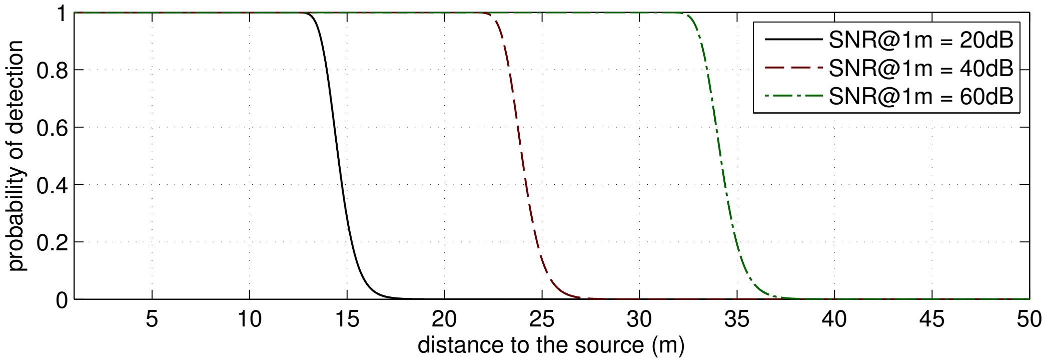


Figure 4.14: Probability of detection for the last pulse sample for three different signal to noise ratios produced by the source at 1 m from it.

Figure 4.15 presents, for the same example considered in Figure 4.14, all the distances that produce a probability greater than 0.99. From that figure it can be observed that for a source that produces only a SNR of 20 dB the maximum range of it is 13.12 m. Furthermore, its range increases up to 32.46 m for 60 dB and to reach the 100 m, the source needs to produce more than 200 dB of signal to noise ratio at one meter which is quit huge, it is greater than the sound ratio between a Saturn rocket and the threshold of human hearing[1].

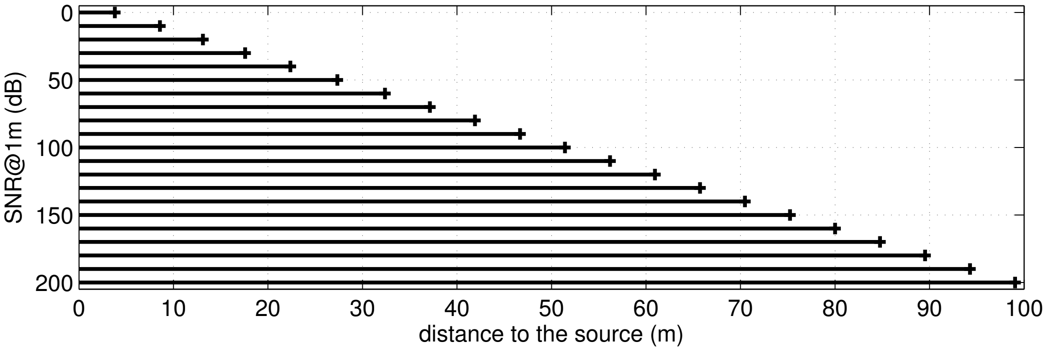


Figure 4.15: Distance to the source that has a probability of detection greater than 0.99 for different signal to noise ratios produced by the source at 1 m from it.

4.3.4 Main-to-Side Lobe Ratio Problem

As seen in the previous sections the probability of detection of the signal is directly related with the signal energy. However, this probability, only, tell us if it is possible to detect the pulse with high probability or not. Nevertheless, the position where the pulse is detected is very important due to fact that if the pulse is detected too early instead of been detected in the last sample of the pulse it will introduce an error into the TOF measure and therefore it will introduce an error in the position estimation. From Figures 4.12 and 4.13 it can be seen that, if the threshold was set to low, the probability of detecting the pulse earlier is greater than for a bigger threshold.

[1] A Saturn rocket produces a sound 194 dB louder than the minimum sound that a human can hear [46].

The probability of detecting the pulse too early is directly related with the MF output shape, which regarding the Equation 4.15 can be seen as the auto-correlation function (ACF) of the pulse delayed by $N-1$ samples [56, 55]. Therefore, instead of studying the MF output shape it is sufficient to study the ACF of the proposed pulse. Regarding this, if the MF output is almost zero for all samples except for the last pulse sample ($N-1$), which is the same of saying that if the ACF is almost zero for all samples except for lag 0, the probability of detecting the pulse for any sample different from ($N-1$) will be the same as the probability of false alarm [56]. As a consequence of this, the ideal pulse ACF is the function that has a single sample with a huge amplitude at lag 0 and it is zero elsewhere [55].

Figure 4.16 presents the comparison between the ideal ACF and two real ACF produced by two OFDM pulse with 50 carriers. We can observe that pulse 1 is nearer to the ideal ACF than pulse 2 which leads to an important conclusion: it is necessary to have a metric to measure how close to the ideal a pulse is. Therefore the main metric used is the main-to-side lobe ratio (MSR). The MSR measures the relation between the amplitude of the main lobe of the ACF and the biggest amplitude of the other lobes, usually known as side lobes [58, 55].

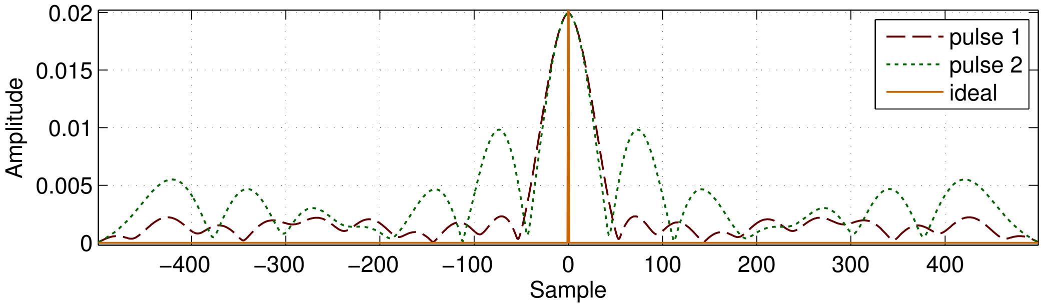


Figure 4.16: Comparison between an ideal auto-correlation function and the auto-correlation function produced by two pulses with 50 carriers.

Figure 4.17 shows how the MSR is measured using pulse 1 from the example depicted in Figure 4.16. Therefore, the MSR value can be between ∞, the best case since the side lobes are 0 (nearest to ideal), and 1, the worst case since the side lobes have the same amplitude of the main lobe and therefore the probability of detecting the pulse early is the same as that of detecting it in the correct position [56, 55].

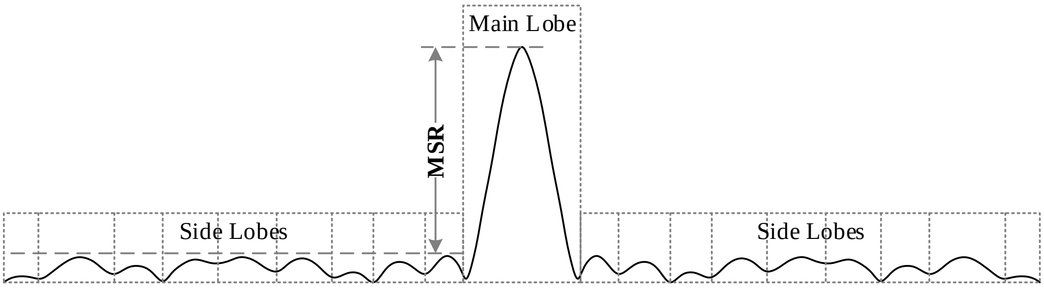
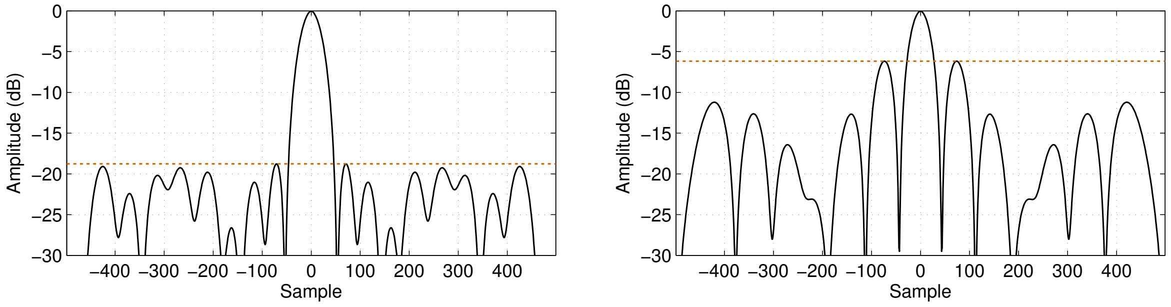


Figure 4.17: Representation of the Main-to-Side lobe Ratio.

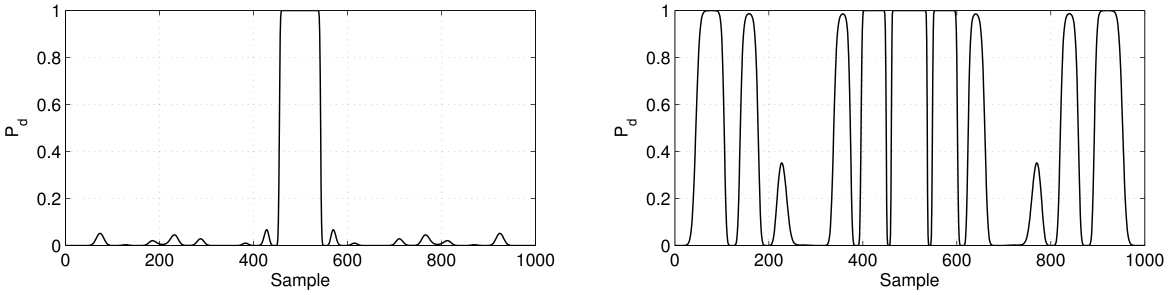
Figure 4.18 depicts the same example of Figure 4.16 but now the ACFs are normalized to the main lobe amplitude and the scale is in dBs. As can be seen the two examples present an huge difference in the MSR value. Pulse 1 is clearly better than pulse 2, pulse 1 presents a MSR of 18.8 dB, whereas the pulse 2 presents 6.2 dB. The probability of detection for this two pulses is presented in Figure 4.19 where can be seen that for the pulse 1 if the probability of detecting the pulse before the last sample is less than 0.05 however for pulse 2 it is very likely to detect the pulse a quite early (almost at the pulse beginning). This uncertainty will increase the detection algorithms complexity and does not bring any advantage to the detection process and TOF measurement [56, 55]. Therefore pulses with a small MSR must be avoided.



(a) Auto-correlation function of pulse 1.

(b) Auto-correlation function of pulse 2.

Figure 4.18: Comparison between the auto-correlation function of two pulses with 50 carriers normalized to the main lobe and the highest side lobe amplitude marked with dark gray dotted line.



(a) Pulse 1 - With a MSR of 18.8 dB.

(b) Pulse 2 - With a MSR of 6.2 dB.

Figure 4.19: Probability of detection for two pulses with different MSR values, considering a probability of false alarm of 1×10^{-6} and a SNR of 0 dB.

Nevertheless, the MSR value variance for an OFDM pulse with all the carriers with the same amplitude decreases with the number of carriers and converges to a fixed value. This statement is confirmed by the plot presented in Figure 4.20 where one can see that the MSR value for 500 or more carriers does not change to much, it changes from a minimum of 4.44 to a maximum of 4.73 for 500 carriers and from 4.56 to 4.69 for 1000 carriers.

From the statistical signal processing theory [57, 59] where the expected value of the ACF for an OFDM pulse, with N_c carriers and carrier's phases uniformly distributed between 0 and 2π, is given by:

$$E[ACF(n)] = \left(1 - \frac{|n|}{N}\right) \sum_{k=k_0}^{k_{N_c-1}} |S(k)|^2 W^{\frac{kn}{N}} \tag{4.42}$$

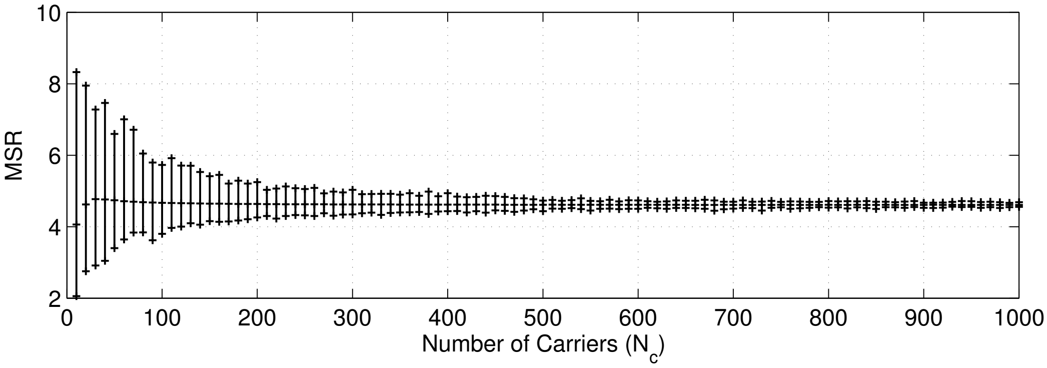


Figure 4.20: MSR minimum, mean and maximum value of an OFDM pulse as a function of the number of carriers. Convergence to 4.64 for large number of carriers. For each number of carriers 10000 trial were carried out.

In the proposed pulse all the carrier's amplitudes are considered to be 1 and therefore, the ACF is given by:

$$E[ACF(n)] = \left(1 - \frac{|n|}{N}\right) \sum_{k=k_0}^{k_{N_c-1}} W^{\frac{kn}{N}} \tag{4.43}$$

Figure 4.21 presents the expected value of the auto-correlation function for an OFDM pulse with 3 different number of carriers. From the figure it can be seen that for a large number of carriers (greater than 500) the MSR could be considered independent of the number of carriers and it will be always equal to 4.64. Moreover, this result is in consonance with the results presented in Figure 4.20.

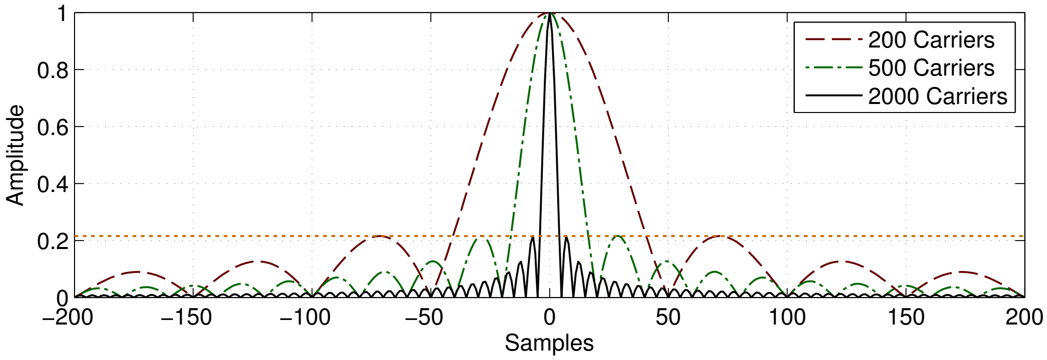


Figure 4.21: Normalized auto-correlation function expected value for an OFDM pulse size of 5000 samples with 200, 500 or 2000 carriers.

As a consequence of this for a considerable number of carriers (more than 500) the MSR value will be always the same and it will be around 4.64. Therefore, under this constraint the chosen carrier phases are not important, considering the MSR problem.

However, the side lobes can be reduced by introducing windowing in time or frequency domain at the transmitter and at the receiver [55]. Nevertheless, due to the demodulation process, time domain windowing will distort the frequency domain information therefore it must be discard. Moreover, the frequency domain windowing at the transmitter will change the carriers amplitudes reducing the effectiveness information differential demodulation process [49, 50] therefore it must be avoided. As a consequence of this, the only way is to use the windowing at the receiver which will degrade the SNR of the detection process [55]. However, this degradation will be only by few a dBs [55]. Moreover, the detection process can be carried out with and without the window. Under these conditions the receiver block diagram is presented in Figure 4.22. Note that, instead of applying the window into the received signal, it can be applied direct to the local $S(k)$ signal, avoiding increasing the receiver complexity. Therefore, instead of using the $S(k)$ signal in the MF the $S(k) \times W(k)$ signal can be used, where $W(k)$ is the frequency window.

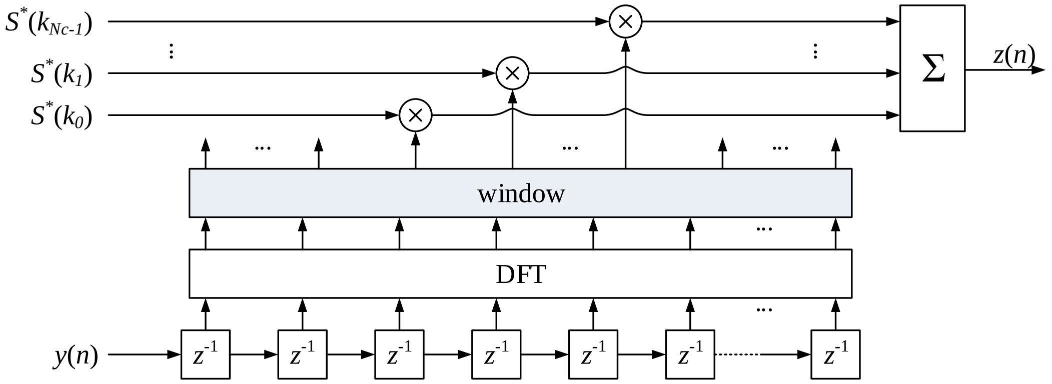
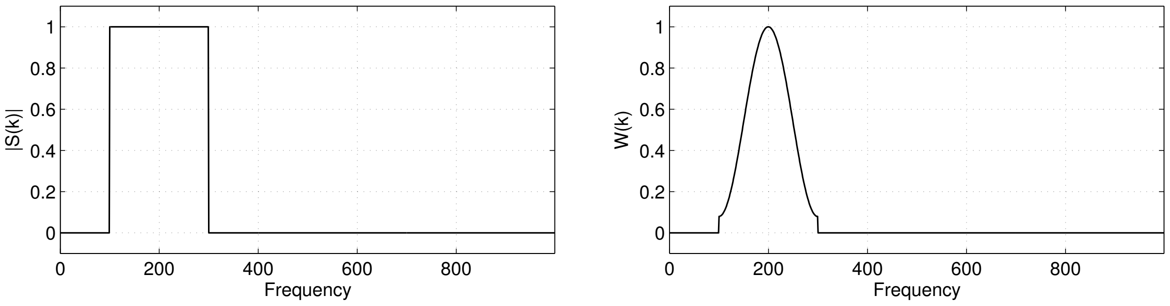


Figure 4.22: Pulse detection with a window at the receiver side.

Figure 4.23 presents the absolute value of an OFDM pulse carriers with 1000 samples and 200 carriers where the first carrier was set to 100 ($k_0 = 100$), it also shows an Hamming window which could be used to reduce the side lobes in the MF output [55, 60, 58].



(a) Absolute value of the carriers.

(b) Hamming window.

Figure 4.23: Absolute value of the carriers and an Hamming windows for a OFDM pulse with 1000 samples size and 200 carriers.

The impact of using the Hamming window in the ACF is depicted in Figure 4.24, where it can be seen that the side lobes are reduced considerably. Nevertheless, this reduction introduces a disadvantage, the main lobe width increases a little bit which can result in a small degradation in the TOF measurement precision.

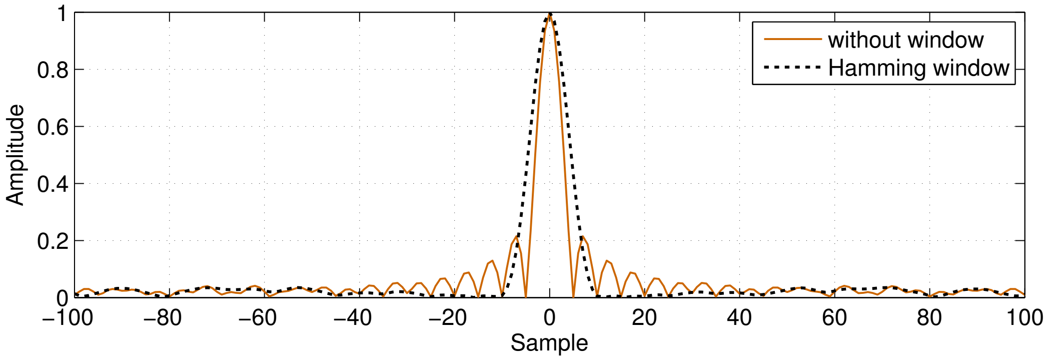


Figure 4.24: The normalized auto-correlation function with and without using a Hamming windows.

4.3.5 Peak-to-Average Power Ratio Problem

One of the major drawbacks of using OFDM pulses for pulse detection purposes is its energy variation for the same amplitude. As can be seen in Figure 4.25 the pulse amplitude presents some variations, but worse than that, at some samples it presents peaks much higher than the signal mean. Moreover, as presented in Section 4.3.3 the probability of detecting the pulse increases with signal to noise ratio, in addition to this, if the transmission system uses a power amplifier it is important to increases the signal energy and reduce the signal amplitude peak in order to increase the power amplifier efficiency [48].

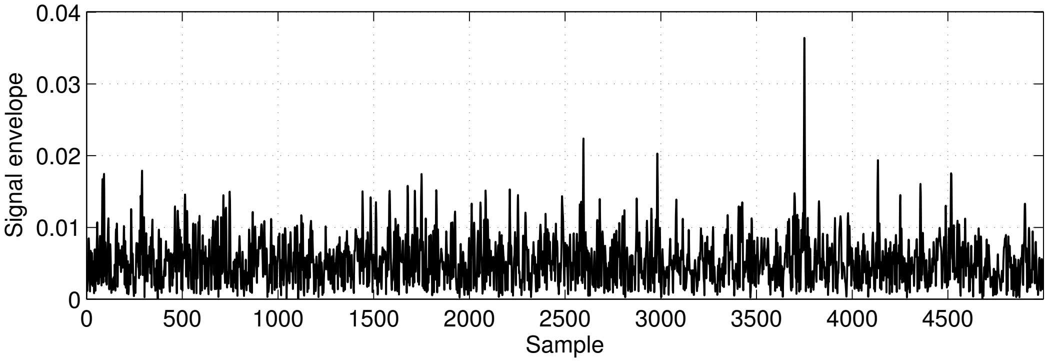


Figure 4.25: Example of an OFDM 5000 samples pulse with 1000 carriers with amplitude 1 and random phase.

The pulse used for detection purposes needs to present a peak-to-average power ratio (PAPR) as low as possible [48]. The PAPR is defined as the ratio between the peak power and the mean power of the OFDM pulse which means that the goal is a PAPR of 1, the peak power is the same as the mean power (like what happens in DC).

Nevertheless, the literature usually covers the PAPR problem for communication purposes [61, 62] but for pulse detection, where the quest for the best pulse is the goal, the typical solutions are not for the PAPR problem but for a similar one, the peak-to-mean envelope power ratio (PMEPR) [55]. The PMEPR instead of measuring the ratio between the peak power and mean power of the real sent signal it computes the ratio using the signal envelope. For narrow-bandwidth signals[2], the PMEPR provides a good approximation of the PAPR value (typical radar case) [55, 62]:

$$\text{PAPR} \approx 2 \times \text{PMEPR}. \tag{4.44}$$

This result comes from the fact that for low-bandwidth signals the instantaneous peak power is almost the same for the real signal and for the envelope signal (see Figure 4.26). As the mean power of the real signal is half of the envelope power, the PAPR will approximate the double of the PMEPR [62]. In Figure 4.26 the peak power is 1.5073 for the envelope and 1.5065 for the real signal, moreover the mean power is 0.4 for the envelope and 0.2 for the real signal. Which result in a PMEPR of 3.768 and a PAPR of 7.533. As foreseen the PAPR is approximately the double of the PMEPR.

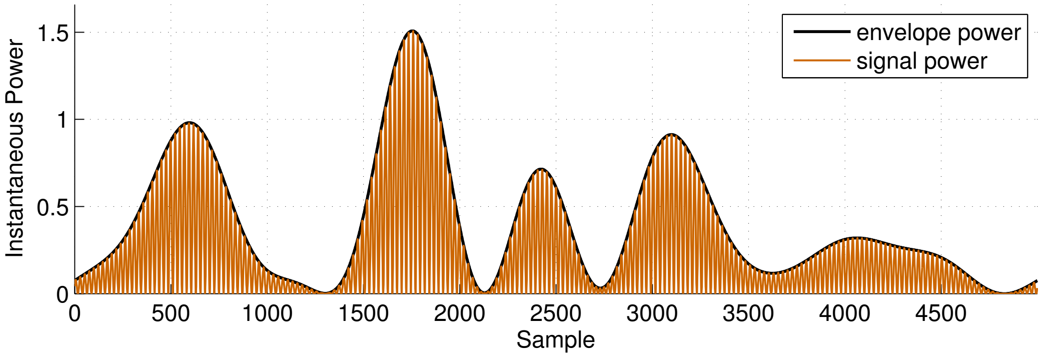


Figure 4.26: Envelope instantaneous power and real signal instantaneous power for a pulse with narrow-bandwidth.

However, for typical US signals (up to 100 kHz) narrow-bandwidth signals are not very useful due to the fact that they must present a bandwidth of a few Hertz (increasing the pulse size and/or reducing the Doppler resistance due to the carriers distance). Nevertheless, there is a relation between the PAPR and the PMEPR which is:

$$\text{PAPR} \leq 2 \times \text{PMEPR}. \tag{4.45}$$

As a consequence of this, the PMEPR value can differ considerably from the PAPR value for US pulses. This statement is illustrated in the example of Figure 4.27. Moreover, Figure 4.28 presents in detail what happens between the sample 2200 and 2400 samples. From both figures it can be seen that the main difference to the low-bandwidth case is that the power peaks for the real signal can be smaller than those for the envelope signal. In this example the peak power is 187.7 for the envelope and 71.135 for the real signal, moreover the mean power is 40 for the envelope and 20 for the real signal. This results in a PMEPR of 4.693 and a PAPR of 3.557. As can be seen the PAPR is much smaller than the double of the PMEPR. According to these results the PMEPR can only be used to give a upper-bound of the PAPR and, therefore, techniques that reduce the PMEPR may also reduce the PAPR.

[2] Narrow-bandwidth signals are signals whose carriers' frequency is much greater that the signal bandwidth.

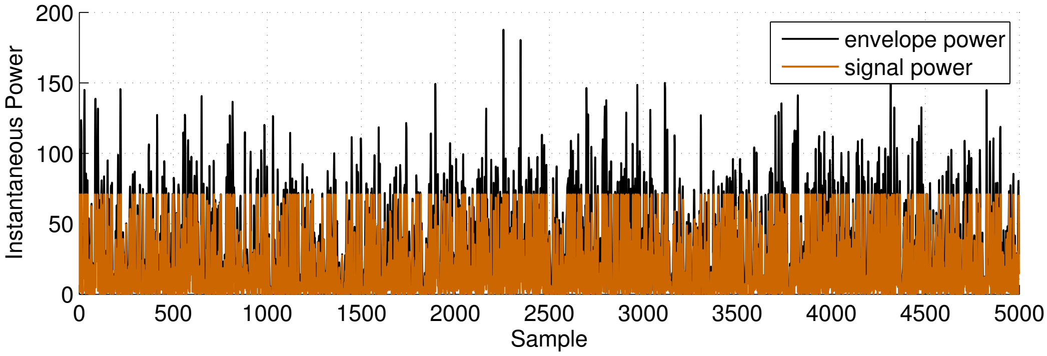


Figure 4.27: Envelope instantaneous power and real signal instantaneous power for a pulse with high-bandwidth.

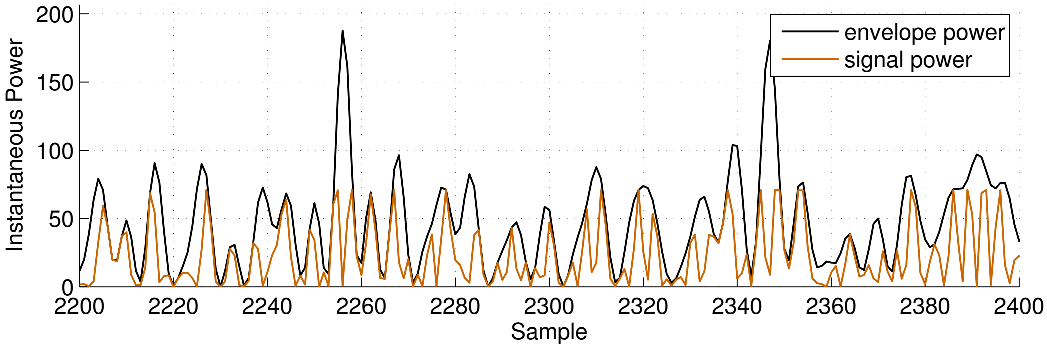


Figure 4.28: Impact of high-bandwidth into the PAPR and PMEPR difference.

OFDM Pulses with low PMEPR.

In the proposed OFDM pulse the carriers have the same amplitude and the only free parameter is the initial phase. Under those constraints, there are several closed form expressions to determine a set of initial phase that will produce a low PMEPR [61, 62, 55]. From the several methods studied there are two with a very simple closed forms that can be applied directly to the proposed pulse, namely the Newman method [63] and the Narahashi and Nojima method [64].

In the Newman method the carrier's phase is given by:

$$\theta(k) = \frac{(k-1)^2}{N_c}\pi, \tag{4.46}$$

whereas, in Narahashi and Nojima method is given by:

$$\theta(k) = \frac{(k-1)(k-2)}{N_c - 1}\pi. \tag{4.47}$$

In spite of the Narahashi and Nojima method being a more recent one, it only presents some advantages for OFDM pulses with less than 30 carriers (see Figure 4.29), therefore, the Newman method will be used to compute the carrier phase of the OFDM pulse. As we can verify from Figure 4.29 the Newman method produces a PMEPR around 1.82. This is a very good value but not really close to the optimum value that some other methods can achieve [61, 62, 55]. However, as it will be seen, this will not be a problem and can be easily overcame.

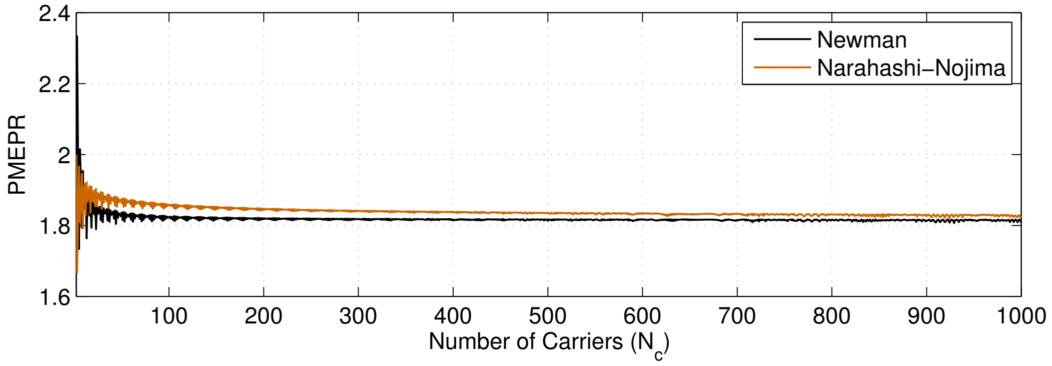


Figure 4.29: PMEPR value for Newman and Narahashi-Nojima methods as a function of the number of carriers.

OFDM Pulses with low PAPR.

Using the Newman method it is possible to achieve a PMEPR of 1.82 which tells us that the PAPR of the OFDM pulse will be smaller than 3.64, which is rather large than the desired PAPR.

Nevertheless, the resultant PAPR from the Newman method can be reduced by an algorithm adapted from the algorithm proposed in [65] which is based on the Papoulis-Gerchberg algorithm [43, 44].

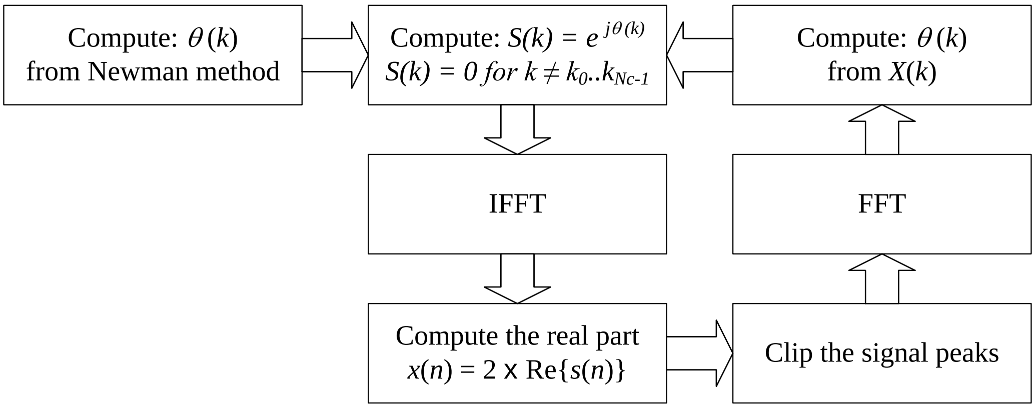
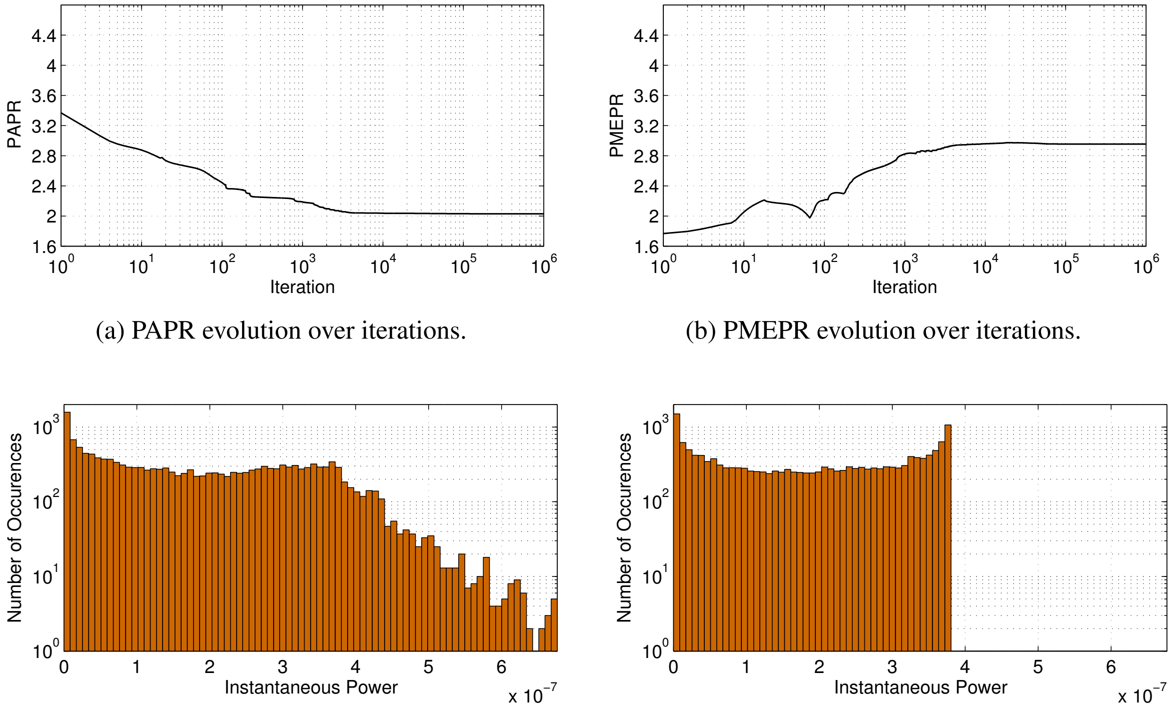

Compute: $\theta(k)$ from Newman method
Compute: $S(k) = e^{j\theta(k)}$ $S(k) = 0$ for $k \neq k_0..k_{Nc-1}$
Compute: $\theta(k)$ from $X(k)$
IFFT
FFT
Compute the real part $x(n) = 2 \times \mathrm{Re}\{s(n)\}$
Clip the signal peaks

Figure 4.30: The iterative algorithm to decrease the PAPR.

The algorithm is presented in Figure 4.30, it starts by computing the carrier phases, $\theta(k)$, using the Newman method. After that it computes the frequency carriers information, $S(k)$, which have amplitude one. The resultant signal is then passed to the time-domain and it is taken the double from its real part. Note that the double is only important to keep the carriers amplitude equal to one. Therefore, from the resultant signal, $x(n)$, the peaks are removed by clipping the maximum and the minimum of the signal. The clipping process must be between 75% to 95% from the maximum amplitude of the signal to ensure that the algorithm converges and that the PAPR is reduced as fast as possible. After passing the clipped signal to the frequency-domain the new carriers phases are obtained and the first iterations are completed. For each iteration, the carriers phase from the last iteration must be used. Note that in order to increase the PAPR reduction, the clipping process can starts at 75% and must be reduced in each iteration in order to keep the algorithm convergence [65]. To test the algorithm a 10000 samples OFDM pulse with two different number of carriers (100 and 1000) was used. In both situations the algorithm was ran one million times and the clipping process started at 80% and if the PAPR during one iteration did not reduce the clipping changed to 0.8 of the previous value plus 20%. The result for this test is presented in Figures 4.31 and 4.32 for 100 and 1000 carriers respectively. As can be seen the PAPR reduces considerably in both cases (see Figures 4.31a and 4.32a), it starts at 3.63 and ends at 2.03 for the 100 carriers case and it starts at 3.61 and ends at 1.95 for the 1000 carriers case. From both figures it is easy to get a PAPR around 2 the problem is to reduce the PAPR bellow 2. Another interesting situation is what happens to the PMEPR value. In spite of the fact that the PAPR is reduced over each iteration the PMEPR goes the other way around, it increases considerably, it starts at 1.82 and ends at 4.56 for the 100 carriers case and it starts at 1.81 and ends at 2.95 for the 1000 carriers case. This reinforces the idea that the PMEPR is not enough, by itself, to find a pulse with a very good PAPR. Figures 4.31c and 4.31d present the instantaneous power[3] distribution before and after the algorithm for 100 carriers case and Figures 4.32c and 4.32d for the 1000 carriers case. As a result of the algorithm the instantaneous power distribution changes considerably. It can be said that after applying the algorithm the instantaneous peak power is completed reduced to around 60% of its previous value.



(a) PAPR evolution over iterations.

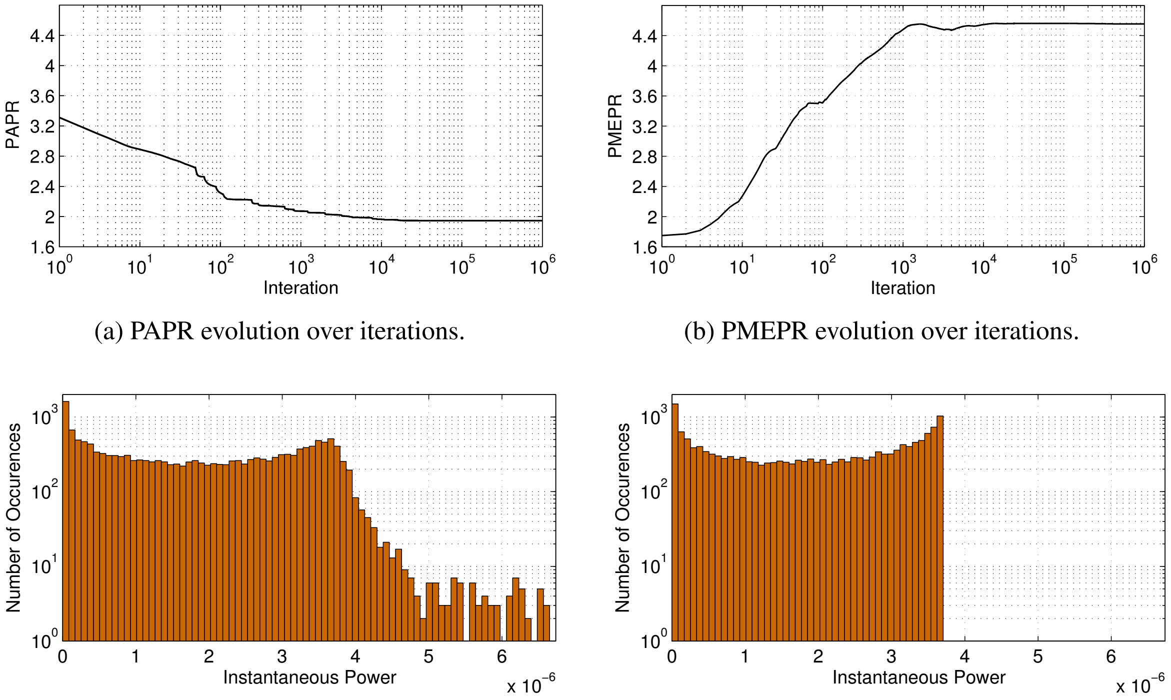
(b) PMEPR evolution over iterations.

(c) Instantaneous Power Distribution before starting the iterative algorithm.

(d) Instantaneous Power Distribution after 10000 iterations.

Figure 4.31: Algorithm results after 1 million iterations for an 10000 samples OFDM pulse with 100 carriers.

[3] The instantaneous power is considered to be the square of the signal absolute value.



(a) PAPR evolution over iterations.

(b) PMEPR evolution over iterations.

(c) Instantaneous Power Distribution before starting the iterative algorithm.

(d) Instantaneous Power Distribution after 10000 iterations.

Figure 4.32: Algorithm results after 1 million iterations for an 10000 samples OFDM pulse with 1000 carriers.

4.3.6 Comparison with Other Usual Pulses

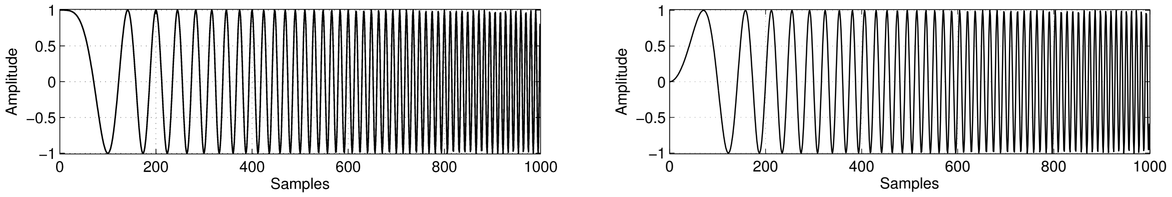
To compare the pulse detection characteristics a chirp signal[4] with the same characteristics will be used. The chirp signal is one of the first signals used in applications where a pulse must be detected with some precision in the presence of noise (like radar and sonar applications) and it probably is still being used in several similar applications nowadays [58, 55]. The chirp signal is a signal whose instantaneous frequency changes linearly inside a limited band during the pulse duration. A complex chirp pulse can be written as:

$$s_c(n) = W^{\left(k_0+\frac{N_c}{2N}n\right)\frac{n}{N}}. \tag{4.48}$$

Figure 4.33 presents am example of a chirp with 1000 samples and, from the signal variation, its instant frequency increases linearly over time.

For a fair comparison, it will be used an OFDM pulse and a chirp pulse with the same main characteristics (size, energy and bandwidth). Moreover, for the OFDM pulse, we apply the PAPR reduction technique presented in 4.3.5. In particular, we will use a 10000 samples pulse with 1000 carriers between the frequency 2000 and 3000, see Figure 4.34.

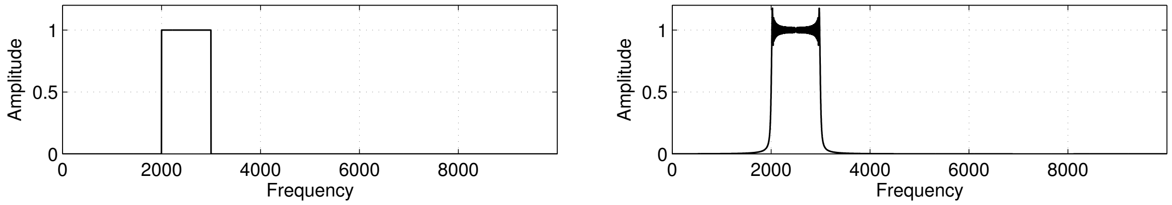
[4] The term chirp is sometimes used interchangeably with sweep signal and linear frequency modulation signal.



(a) Real Part.

(b) Imaginary Part.

Figure 4.33: Chirp pulse example with 1000 samples and its instantaneous frequency starts at 0 ($k_0 = 0$) and ends at 100 ($N_c = 100$).



(a) OFDM pulse spectrum.

(b) Chirp pulse spectrum.

Figure 4.34: Comparison between the spectrum of the OFDM and the Chirp pulse.

For the resultant pulses the ACF is presented in Figure 4.35, where it can be observed that they are very similar, which results in the same MSR value and therefore, the probability of detection will be very similar for both pulses. Moreover, the probability of early detection will also be very similar due to the side lobes amplitude similarity. As a consequence of this, the proposed pulse, will be very similar to the Chirp pulse from the detection point of view.

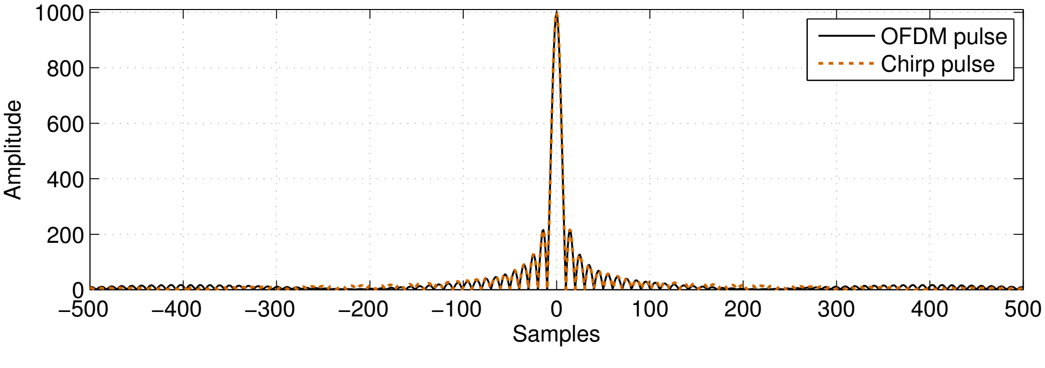


Figure 4.35: The OFDM and the Chirp pulse auto-correlation functions.

This statement is reinforced by the probability of detection as a function of the signal's SNR depicted in Figure 4.36, where it is shown the probability of detection for the last pulse sample $(N-1)$ using a MF and the threshold technique from Figure 4.8 and considering a probability of false alarm of 10^{-6}. The OFDM pulse detection behaves similarly to the Chirp which reveals, once again, that the OFDM pulse presents the same detection proprieties of the Chirp pulse.

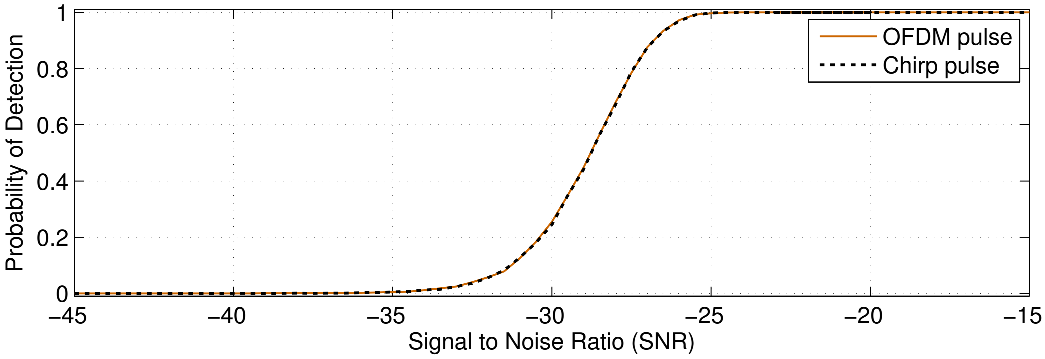
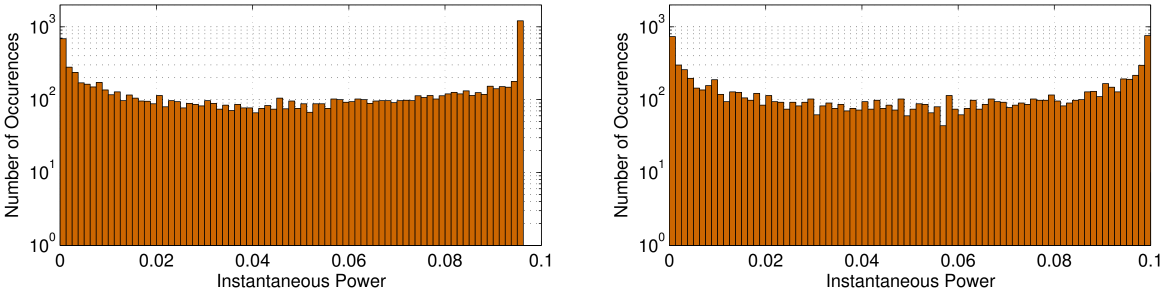


Figure 4.36: Probability of detection for an OFDM and a Chirp pulse as a function of the Signal to Noise Ratio for a probability of false alarm of 10^{-6}.

Regarding the PMEPR problem, the OFDM pulse presents almost the triple value, 2.95 against 1.00 for the Chirp. Nevertheless, the PMEPR is not a important factor for the proposed system, where only the real part of the pulse is used. Figure 4.37 presents the instantaneous power for both pulses, which shows almost the same profile. Furthermore, taking into account the PAPR value, the PAPR reducing technique shows up its value, it presents a PAPR of 1.95 against 2.00 for the Chirp. As a result of this, the OFDM pulse has the same transmission efficiency of the Chirp pulse.



(a) OFDM Instantaneous Power Distribution.

(b) Chirp Instantaneous Power Distribution.

Figure 4.37: Instantaneous Power Distribution for an OFDM and a Chirp pulse with the same characteristics.

4.4 Pulse for Data Communication

There are two types of data communications, the synchronous and the asynchronous communication [66, 49]. The main difference between them relies in the fact that in asynchronous communication the data is sent intermittently instead of continuously with a fixed rate. There are, three main challenges that the asynchronous communication's receiver needs to face that a synchronous does not need to address [66, 49]. First, the receiver must distinguish if the signal that is receiving contains any data or if it is only noise. In addition to this, the receiver needs to unveil the best time instant to sample the data in order to reduce the errors. Finally, the receiver needs to know, when it must stop the demodulation process.

In the proposed US location system, the same acoustic channel will be used for TOF measurement and communication. As a consequence of this, it was decided to use asynchronous data communication instead of synchronous data communication.

In order to deal with asynchronous communication problems, the system will rely on the information provided by the pulse detection process. This information will be used to determine the data position in time and to estimate the best instant to sample that data. To know when the demodulation process should stop it, a fix data length will be used, which means that the receiver must know the pulse size.

4.4.1 Asynchronous Receiver

The block diagram for the proposed asynchronous receiver is shown in Figure 4.38. The *Synchronization and TOF-Measurement* part includes the MF and the OFDM pulse detection process. Only after the detection of the pulse, The *Data Extraction* starts, the first stage is to store the received synchronization pulse, that will be used in a second stage to demodulate the data pulse to extract the data.

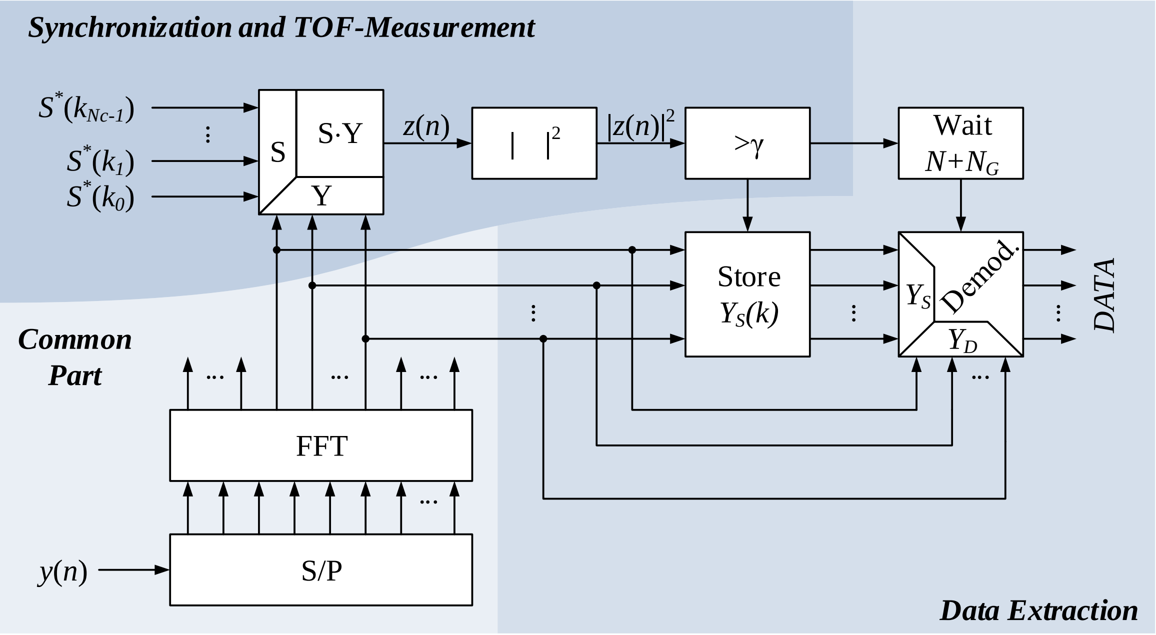


Figure 4.38: The block diagram of the asynchronous OFDM receiver.

As presented in Section 4.3.4, the OFDM pulse can be detected in a early stage instead of the last sample of the OFDM synchronization pulse. This event is presented in Figure 4.39, where, due to the low threshold value, the pulse is detected earlier. Nevertheless, this will not be a problem to obtain the synchronization pulse, $Y_S(k)$, and data pulse, $Y_D(k)$, because the *Guard FFT* and the *Guard Time* will accommodate this type of events (see Figure 4.39). However, this mismatch will introduce phase shifts in the OFDM carriers of both OFDM pulses.

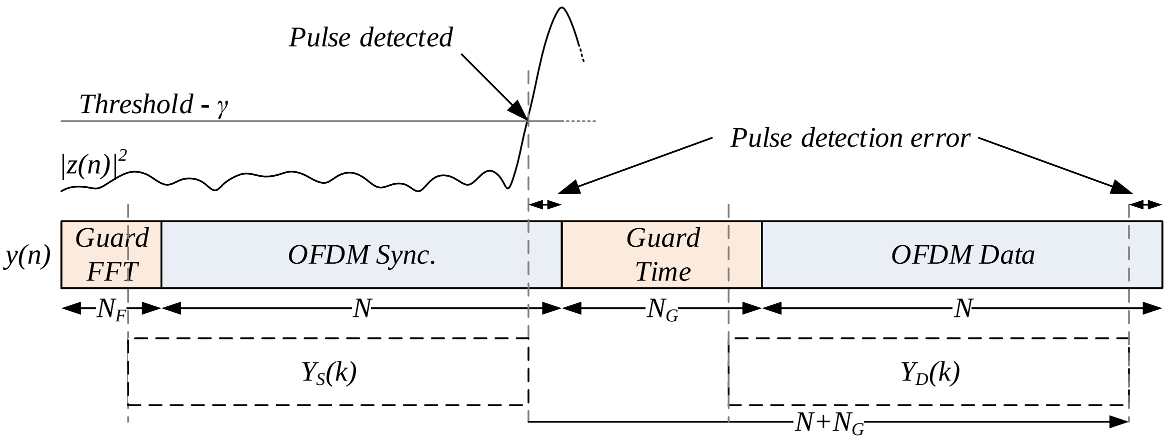


Figure 4.39: Exemplification of a possible pulse detection.

The resultant phase shifts in the OFDM data pulses can produce catastrophic results, it is possible that a symbol with phase 0 can be converted into phase π. For example, in a 1000 samples pulse and for carrier 125 ($k = 125$), 2 samples delay is enough to change its phase from 0 to $\pi/2$ and for the carrier 250 it is only needed 1 sample to produce the same effect, which in BPSK can easily result in an error in the demodulation process. Moreover, as seen in Figure 4.21 the received pulse presents a considerable width and therefore it is impossible to obtain sub-sampling resolution. As a result of this the use of typical modulation schemes, such as, PSK and QAM, is not possible [49, 50, 48]. The best way to accommodate this pulse detection error is to use differential demodulation [49, 50], such as DBPSK, by using this technique the OFDM synchronization pulse, will be used not only for synchronization proposes but also as a phase reference for demodulating the OFDM data pulse. By using this technique the phase error produced in the data pulse will be canceled with the error produced in synchronization pulse since the phase shifts experienced for both pulses are the same.

In a simple algorithm: after the system detects the pulse, it chooses the maximum of the output of the detector on the next N samples (the length of the pulse plus the *Guard FFT* time if it exists); After that the system demodulates the data using the synchronization pulse by a differential comparison.

4.4.2 Modulation Choice

The DBPSK modulation was chosen because it presents a very good BER in the presence of noise as can be seen from Figure 4.40. The DBPSK needs only 1 dB more than the BPSK to present the same BER. Moreover, similar to the BPSK case, the noise needs to change the carrier's phase more than $\pi/2$ to produce an error into the data [49].

After the synchronization pulse detection, in order to obtain the information, the received data pulse, $Y_D(k)$, must be compared with the stored synchronization pulse, $Y_S(k)$. If the carrier phase difference is between $-\pi/2$ and $\pi/2$, the receiver considers the bit data zero otherwise it considers the bit data one.

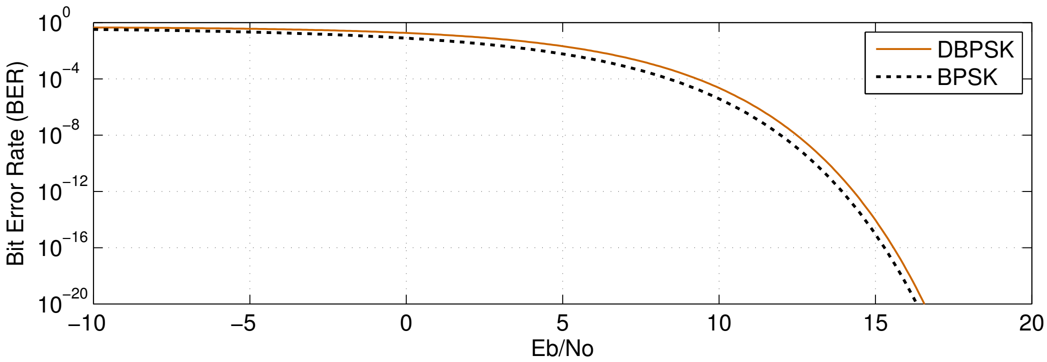


Figure 4.40: Noise performance for BPSK and DBPSK modulation.

4.4.3 Comparison with Other Pulses for Data Communication

To validate the proposed frame we will compare it with another possible method. This method uses a Chirp for synchronization and usually modulation DBPSK for data transmission. The block diagram of the proposed pulse is presented in Figure 4.41.

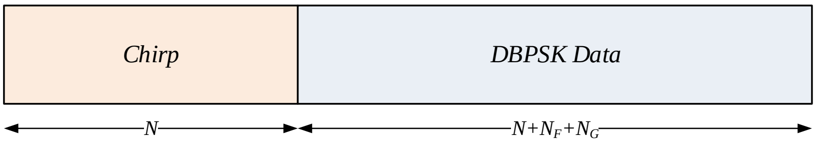


Figure 4.41: Block diagram of the frame prototype with a Chirp and DBPSK.

Under these conditions the Chirp pulse will have the same size , N, and the same bandwidth, N_c of the OFDM pulse. Moreover, for a fair comparison the proposed frame prototype must have the same size, and therefore, the *Data* can occupy $N + N_F + N_G$ samples. As a result of using DBPSK, for data transmission, the same amount of data will need two times more samples than the OFDM [49, 50, 48]. Therefore, N_F can be set to $N/4$ samples and N_G can be set to $3N/4$.

Considering a 10000 samples OFDM pulse with 1000 carriers and a sampling frequency of 250 kHz, it results into a bandwidth of 25 kHz (from 25 kHz to 50 kHz), a pulse duration of 40 ms and a frame duration of 120 ms. The result pulse duration values are presented in Figure 4.42.

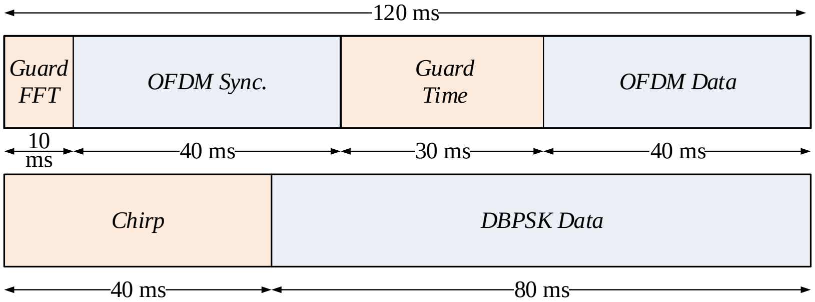


Figure 4.42: Pulse duration values for a 10000 samples OFDM pulse with 1000 carriers sampling at 250 kHz.

4.4.4 Pulses Comparison with Noise

From the literature [49, 50] the bit error rate for a DBPSK signal mixed with white Gaussian noise is given by:

$$\mathrm{BER} = 0.5 \times e^{-\gamma}, \tag{4.49}$$

where, γ is the relation between the signal energy and the power spectral density of the noise and it can be easily related to the SNR by;

$$\gamma = \mathrm{SNR} \times \frac{f_s}{2 \times B} = \mathrm{SNR} \times \frac{N}{2N_c}. \tag{4.50}$$

Figure 4.43 presents the BER for both techniques and it can be seen that both techniques follow the theoretical expression. At first sight, using one or another technique does not bring any advantaged to the user, however, it was considered that the receiver can synchronize very well with the received pulse, and sample the data in the correct position.

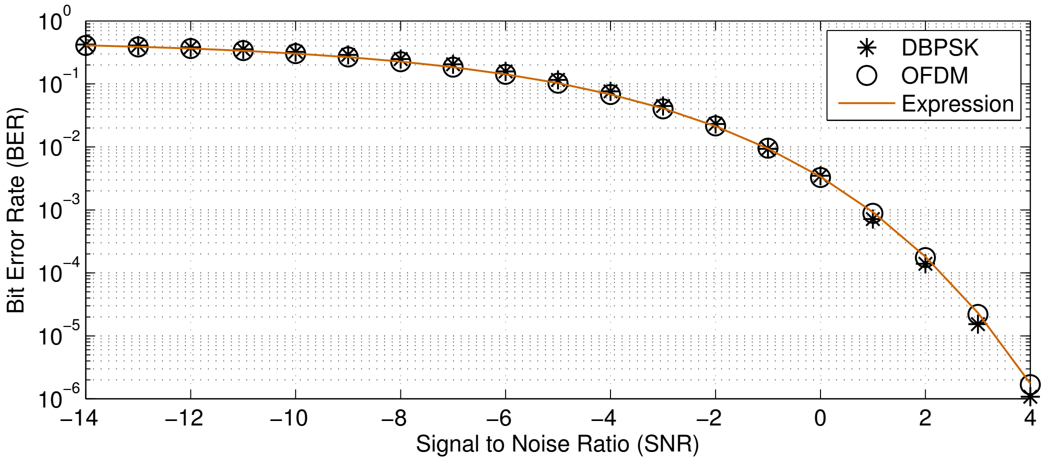


Figure 4.43: Comparison between the bit error rate of the OFDM and the Chirp + DBPSK pulse in the presence of white Gaussian noise.

As it was seen in the synchronization process (section 4.3) by using the threshold technique the pulses can be detected in the wrong position. Therefore, some jitter arises in the synchronization process leading to not sampling the data in the best position. Figure 4.44 presents the BER produced by the synchronization jitter for OFDM and DBPSK. In this test it was considered, in the DBPSK case, to sample the bit information in the middle of it and in the OFDM case, in the middle of the guard FFT. As a consequence of this the DBPSK only allows 20 samples of error in the synchronization process whereas the OFDM goes up to 1250 samples (half of the guard FFT).

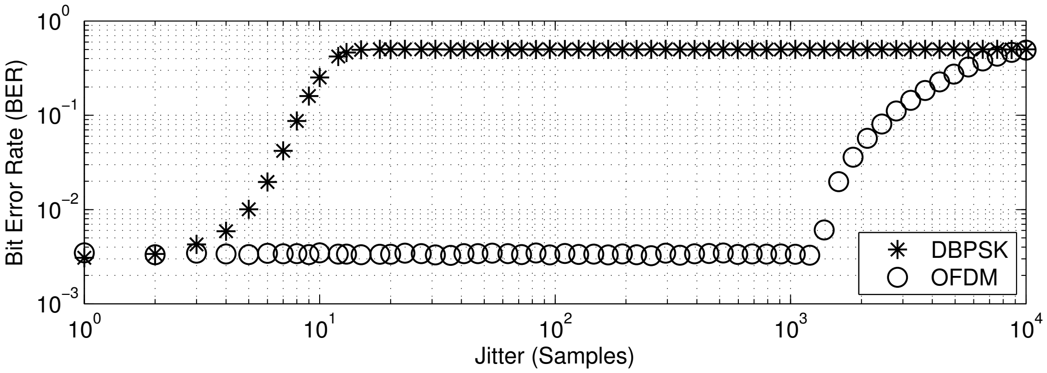


Figure 4.44: Comparison between the bit error rate of the OFDM and the Chirp + DBPSK pulse for a SNR of 0 dB in the presence of synchronization jitter.

4.4.5 Pulses Comparison with Multipath

For the follow three comparative tests it will be considered that the receiver is perfectly synchronized, avoiding the error produced by the synchronization jitter.

Test with One Reflection

For this test we will consider the system represented in Figure 4.45 where the received signal will be composed by two contributions: a direct wave and a reflected one which results from a single reflection on a surface.

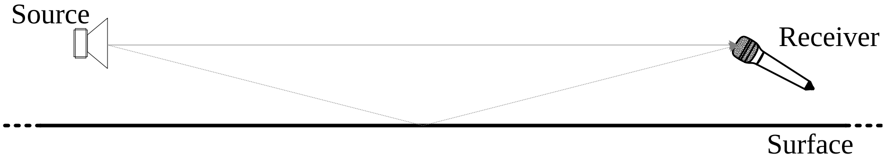


Figure 4.45: System representation used to compare the pulse performance in the presence of a single reflection.

Under these conditions the reflected wave is going to be received after some time in relation to the direct wave since it travels a longer distance. This process can be represented by the IR of Figure 4.46, where the direct wave is the reference and the reflected wave arrives after N_d samples with the same amplitude but with a phase randomly chosen between 0 and 2π.

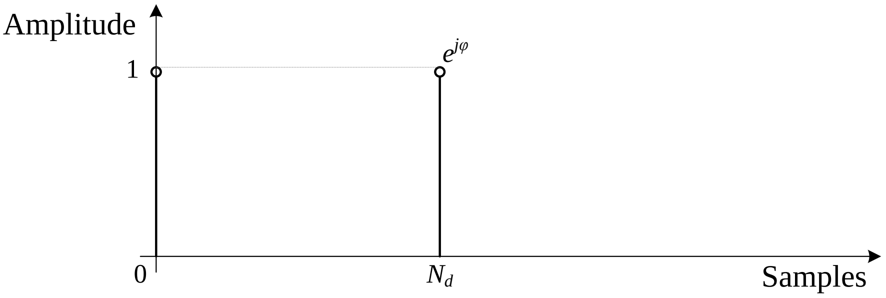


Figure 4.46: Impulse response used to compare the pulse performance in the presence of a single reflection with the same amplitude. Where φ is uniformly distributed between 0 and 2π.

The resulting BER for a SNR of 10 dB is represented in Figure 4.47 where it can be seen that the Chirp and DBPSK presents a huge BER (around 0.2) which results from a strong ISI (note that: the bit duration is only 20 samples). For the OFDM the BER is almost constant (less than 0.03) for a reflection delay up to 2500 samples which is almost ten times less than the DBPSK. And even after the 2500 samples delay, the BER increases slowly up to around 7500 samples delay (resulting into a BER of 0.05). These facts show the benefit of the OFDM and in particular the benefit of the guard bands used in the proposed pulse (2500 samples for the guard FFT and 7500 samples for the guard time).

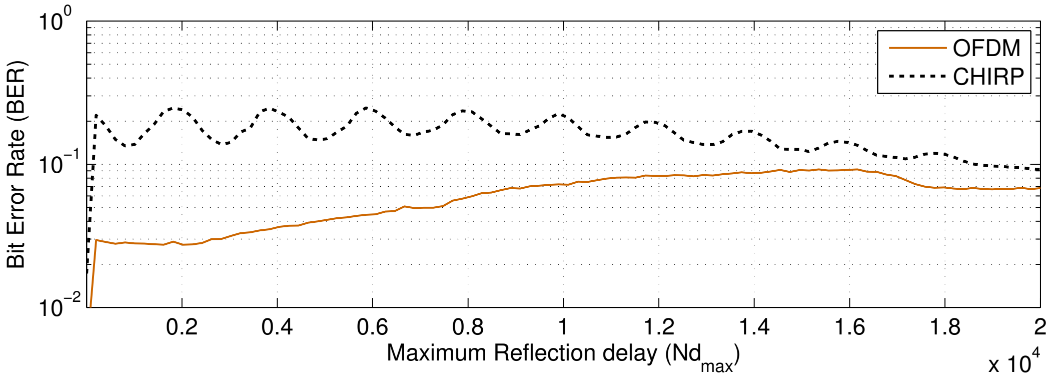


Figure 4.47: Bit error rate for OFDM and Chirp + DBPSK in the presence of a single reflection with the same amplitude for a 10 dB SNR.

Nevertheless, the IR used to compare the pulses is far from realistic since the reflected wave needs to travel a longer distance than the direct wave and, therefore, it will be more attenuated than the direct wave, as seen in Section 3.3.2. Under this assumption it is more realistic to consider that the reflected wave amplitude is a function of the arrival time difference as presented in Figure 4.48.

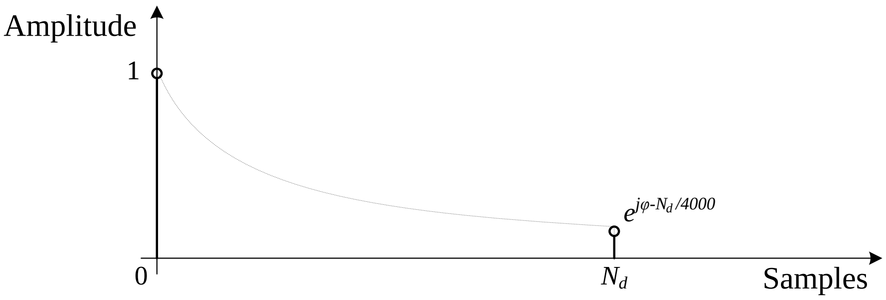


Figure 4.48: Impulse response used to compare the pulse performance in the presence of a single reflection where its amplitude decreases with the arrival time difference.

Under this assumption and considering that the source and the receiver are considerably apart (more than 10 m) the relation between the attenuation suffered by the direct and the reflect wave will be dominated by the propagation losses, see Section 3.3.2. Considering a propagation loss of 1.6 dB/m (typical value for ultrasonic range see Figure 3.11) and a SNR of 10 dB, the resulting BER is represented in Figure 4.49. From this figure it can be seen that as the delay value increases the reflected wave amplitude decreases and therefore the ISI decreases. This will reduce the BER and will approximate the Chirp and DBPSK performance to the OFDM case.

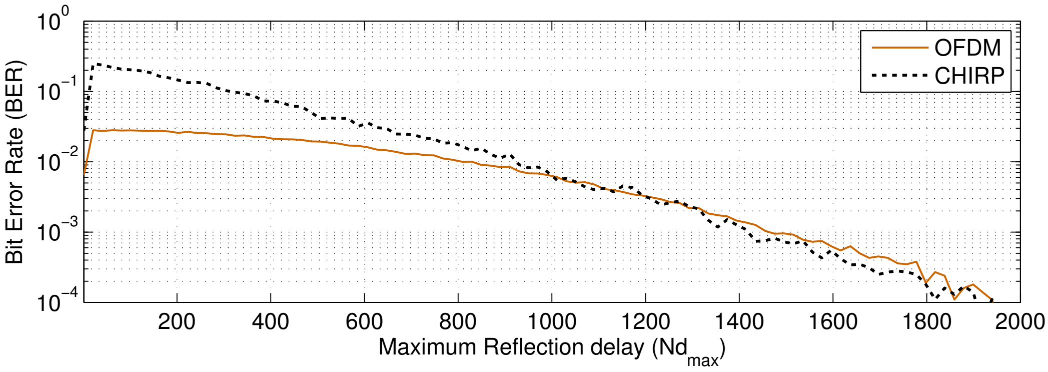


Figure 4.49: Bit error rate for OFDM and Chirp + DBPSK in the presence of a single reflection where its amplitude decreases with the arrival time difference.

Test with Multiple Reflection

Until now only a single reflection was considered, and once again this is a bit different from what happens in a real room environment, where the ultrasonic wave can be reflected in several objects and arrive to the receiver at different instants. As a result of this the received wave is going to be a mixture of the received reflections at different instants of time. In addition, the reflections amplitude is going to decrease as the time elapses. This can be represented by the IR of Figure 4.50 where the impulse response amplitude decreases as time elapses. However, in order to turn it more realistic it was considered that each sample of the IR will be a complex random variable with the variance given by the square root of the expected amplitude resulting in an impulse response similar to that presented in Figure 4.51.

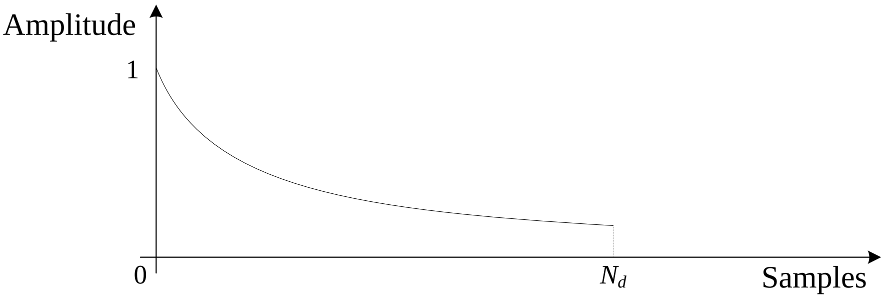


Figure 4.50: Impulse response used to compare the pulse performance in the presence of multiple reflections where the amplitude decreases with the arrival time difference.

For this case considering once again that the source and the receiver are considerably apart, a propagation loss of 1.6 dB/m and a SNR of 10 dB, the resulting BER is represented in Figure 4.52. From this figure it can be seen that the delay value is independent of the impulse duration. The DBPSK presents a poor result, a BER of 0.5, and the OFDM presents a BER around 0.01. Therefore, under these conditions the ISI is so strong that it is impossible to use the DBPSK.

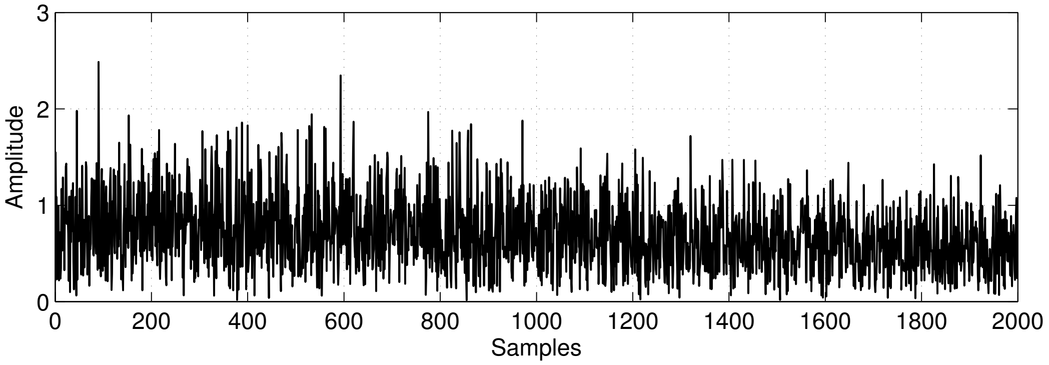


Figure 4.51: Example of an impulse response used to compare the pulse performance of the two methods in the presence of multiple reflections. Its amplitude is a complex random variable whose variance decreases with the arrival time difference.

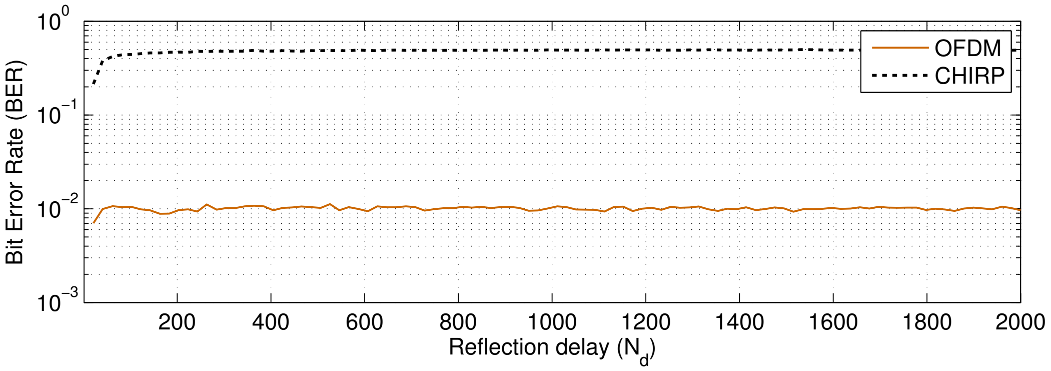


Figure 4.52: Bit error rate for OFDM and Chirp + DBPSK in the presence of multiple reflections where there amplitude decreases with the arrival time difference.

From the literature [49, 50] the bit error rate in the presence of multipath and with almost none ISI is given by:

$$\mathrm{BER} = \frac{0.5}{1+\gamma} \tag{4.51}$$

To test this expression and verify if the BER is independent of the IR duration, three different tests were carried out with different IR durations. The results are depicted in Figures 4.53, 4.54 and 4.55 for an IR duration of 100, 500 and 1000 samples respectively. As can be seen from the three figures the BER follows very well the expect value given by (4.51) and the result is almost independent of the pulse duration. Moreover, for the three cases the IR duration is bigger than the bit duration for the DBPSK modulated signal, but less than the bit duration for OFDM case. As a consequence of this, the DBPSK case will present a strong ISI and the OFDM will not. Nevertheless, it is important to mention that a 100 samples IR means that the last reflection arrives 100 samples (or 400 μs) after the direct wave and in US this represents only 14 cm. Taking only this into consideration it will be almost impractical to use DBPSK for data communication in an indoor US location system.

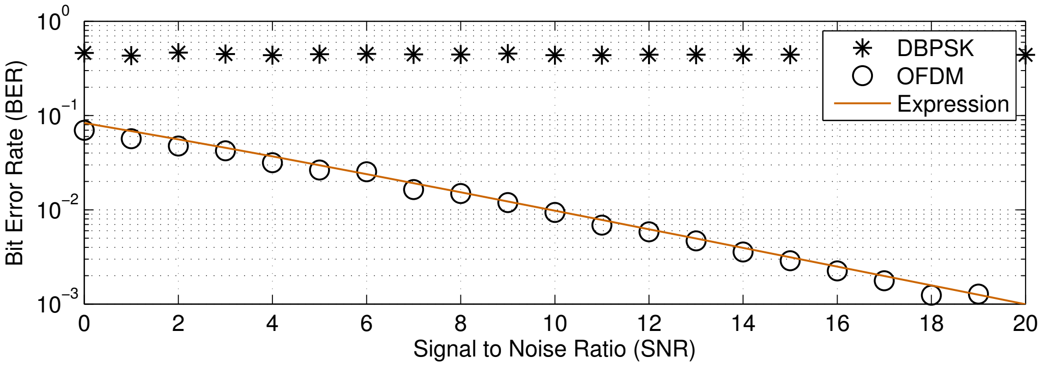


Figure 4.53: Bit error rate for OFDM and Chirp + DBPSK in the presence of multiple reflections for an impulse response duration of 100 samples.

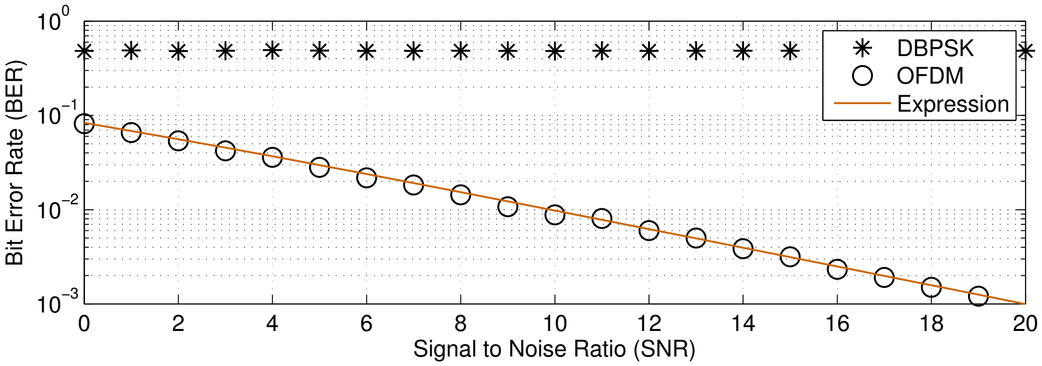


Figure 4.54: Bit error rate for OFDM and Chirp + DBPSK in the presence of multiple reflections for an impulse response duration of 500 samples.

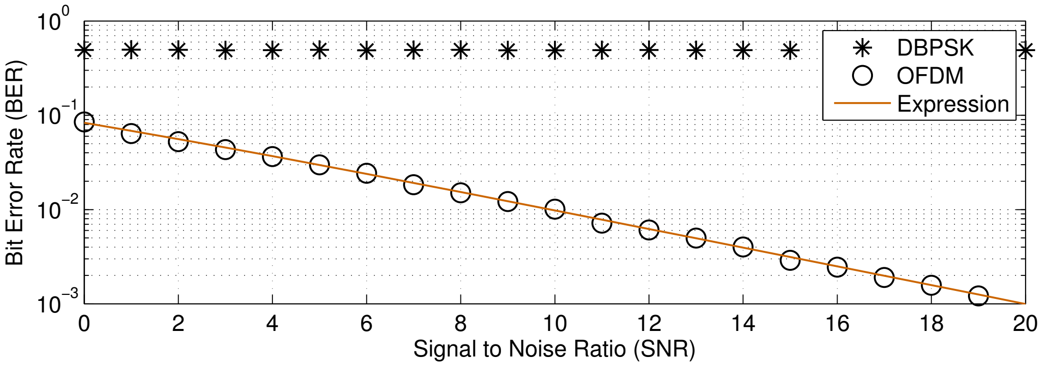


Figure 4.55: Bit error rate for OFDM and Chirp + DBPSK in the presence of multiple reflections for an impulse response duration of 1000 samples.

The Multipath Impact

To test only the multipath impact (without noise) in both methods we used the implemented ultrasonic simulator. The implemented room presents a very simple rectangular box shape with 15×6×3 m as it is shown in Figure 4.56. Moreover, the walls have a reflection coefficient of 0.5 which means that the reflected ultrasonic wave has fourth the energy of the incident wave. The room conditions were set to 20 °C of temperature, 50% of relative humidity and 1 atm pressure.

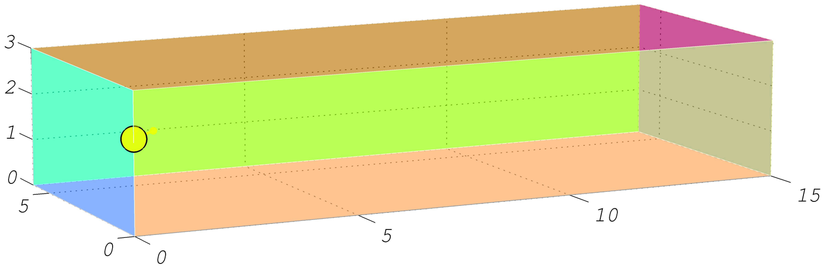
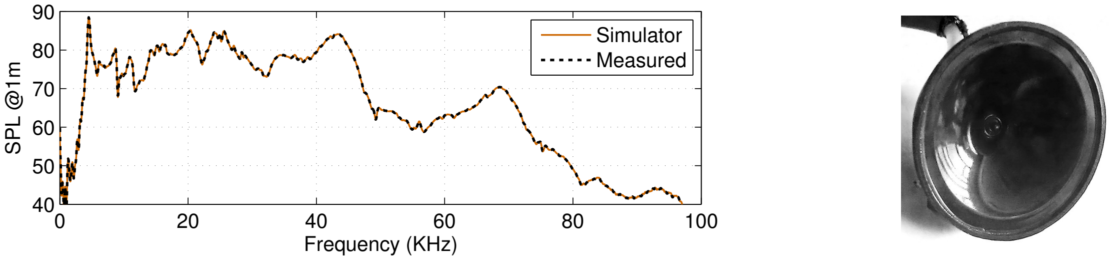


Figure 4.56: Room shape implemented into the simulator to compare the pulse performances in the presence of multipath.

The source was placed on a corner at 2 m from the floor as depicted in Figure 4.56. The model of the speaker presented in Figure 4.57 was used, it presents a beam width between 30 and 50º depending on the wave's frequency.



(a) Frequency response in front of the speaker (measured and the simulation model).

(b) Front view of the real speaker.

Figure 4.57: Ultrasonic speaker model used in the simulator.

Moreover, the receiver was considered to be omni-directional and it was placed at 1.5 m from the floor and 4 m from the $y = 0$ wall in 299 positions along the x-axis starting at 5 cm from the $x = 0$ wall and ending at 5 cm from the $x = 15$ wall in steps of 5 cm, as depicted in Figure 4.58.

For each position no noise was added to the incoming signal and the synchronization was manually implemented providing the true time position of the pulse avoiding any error that could arise from the synchronization process.

In the comparison two different situations were considered: a narrow band signal (2.5 kHz from 25 to 27.5 kHz) with 100 bits and a wide band signal (25 kHz from 25 to 50 kHz) with 1000 bits. For the narrow band case (Figure 4.59) the DBPSK only presents a good performance when it is inside the beam range of the source (namely between 1 to 6 meters), since the direct wave will probably have much more energy than the reflected waves resulting into a low ISI. On the other hand the BER for the OFDM case is only different from zero in 25 places and never exceeds 0.02. For the wide band case (Figure 4.60) the DBPSK performs even worst than in the previous example manly due to the speaker frequency response, becoming evident the need for equalization. In spite of the BER being different from zero in many more places (105), for the OFDM case, it never exceeds 0.005 which can be considered a better result than the previous one. As mentioned before, this arises from the fact that the OFDM using DBPSK does not need equalization even for variations greater than 20 dB in the frequency response (see Figure 4.57).

Concluding, the proposed OFDM pulse proved to be a good choice for pulse synchronization being side by side from the well used Chirp signal and it proved to be an excellent choice to be used for data communication for indoor US environment.

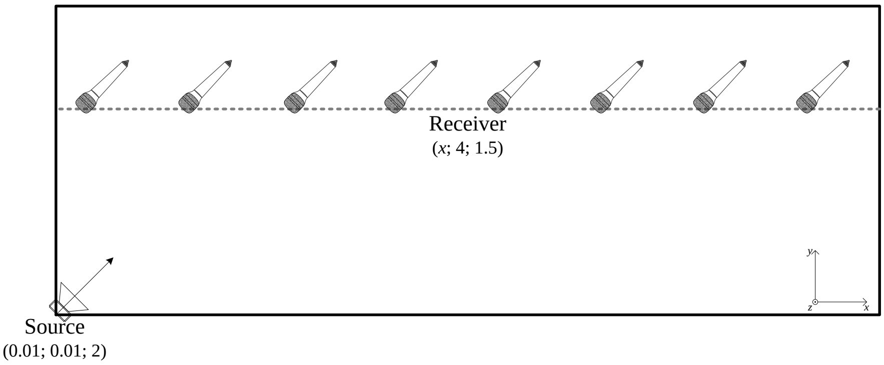
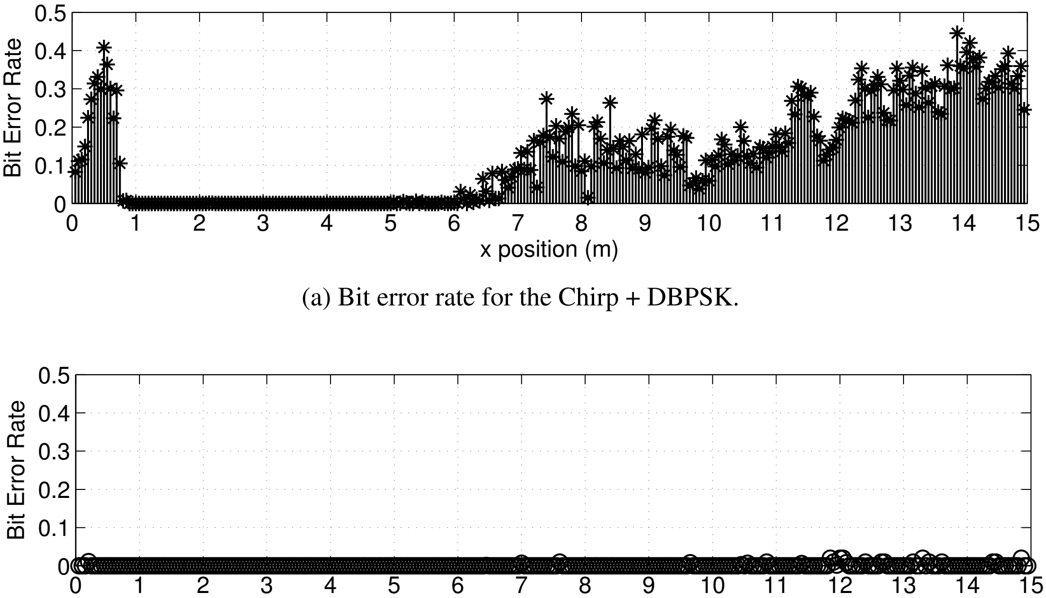


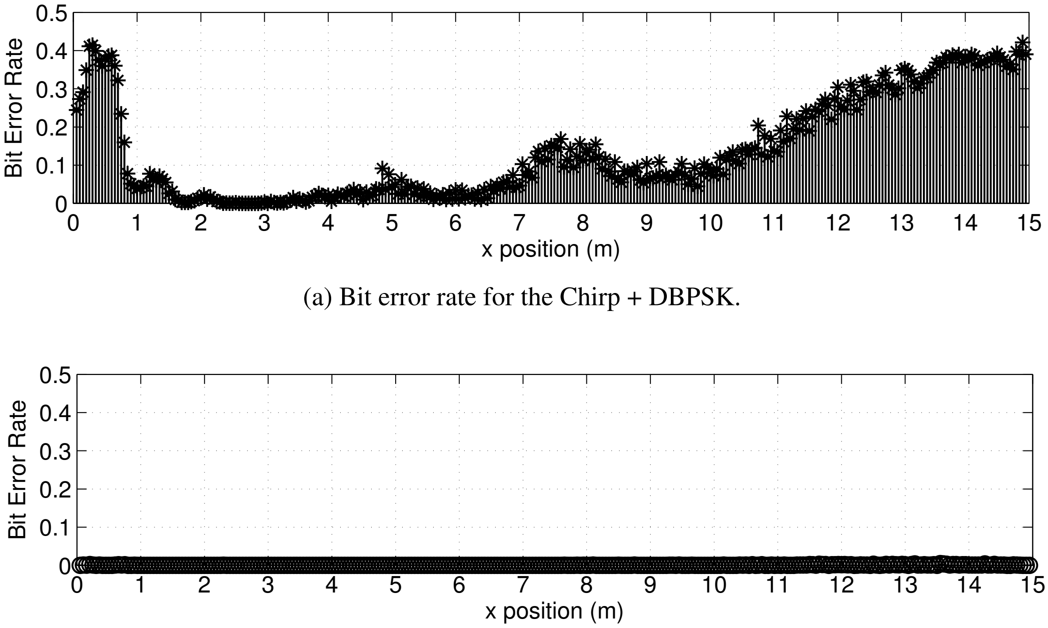
Figure 4.58: Ultrasonic receiver positions to compare the pulse performance in the presence of multipath.



(a) Bit error rate for the Chirp + DBPSK.

(b) Bit error rate for the OFDM.

Figure 4.59: Bit error rate for each x position for 100 bits and a bandwidth of 2.5 kHz.



(a) Bit error rate for the Chirp + DBPSK.

(b) Bit error rate for the OFDM.

Figure 4.60: Bit error rate for each *x* position for 1000 bits and a bandwidth of 25 kHz.

CHAPTER 5

ULTRASONIC LOCATION WITHOUT RF CHANNEL

This chapter presents the proposed ultrasonic location system architecture. Section 5.1 presents an overview of the proposed architecture and its advantages and disadvantages. This architecture consists on a support network of autonomous beacons that is fixed in the ceiling or walls of a room, and mobiles that need to obtain their position in that room. As a result of each beacon being completely autonomous a clock synchronization protocol is proposed in Section 5.2, this protocol will allow, not only, beacons to synchronize their clocks but also to obtain the distance measurements between them by exchanging only three messages from time to time, as will be seen in Section 5.3 and in the protocol results of the examples presented in Section 5.4. The mobiles are responsible to locate themselves applying a very simple algorithm based on TDOA presented in Section 5.6. This algorithm uses the ultrasonic messages received from the beacons with the message time-stamp and the beacon's coordinate. This approach allows the location system to be independent of the number mobiles and will provide a high degree of privacy since the mobiles do not need to transmit any message. On the other hand, beacons obtain their coordinates by the minimization of a local cost function using an iterative algorithm allowing to take advantage not only of the distance measurements to its neighbors but also the distance measurements between them as will be seen in Section 5.7. This approach will reduce the human intervention during the network implementation and malfunctioning beacons can be easily replaced without compromising the network operation. The mobile location algorithm and the beacon location algorithm will be tested in some examples presented in Section 5.8.

Additionally, it is important to mention that during the presentation and discussion on this chapter it will be considered that each node makes use of the ultrasonic pulse proposed in the previous chapter. Which means that each node could perform airborne communication and distance measurement.

5.1 Proposed Ultrasonic Location System Architecture

The proposed ultrasonic location system architecture is composed by a network of beacons that provide all the room coverage allowing mobile nodes to locate themselves using only the information obtained from this network. An example of the proposed architecture for a 2D case is depicted in Figure 5.1. The proposed architecture is composed of three different types of nodes as presented in Figure 5.2.

The network of beacons is composed by static beacons and it is spread over all the room to ensure a complete room coverage allowing a proper location to the mobile nodes. As a result of this, the network will usually be composed by a considerable number of beacons. Therefore, determining their location is an almost impossible task to perform manually, due to the large human intervention. In addition to this, whenever a beacon fails, the location of the replaced node must be determined by

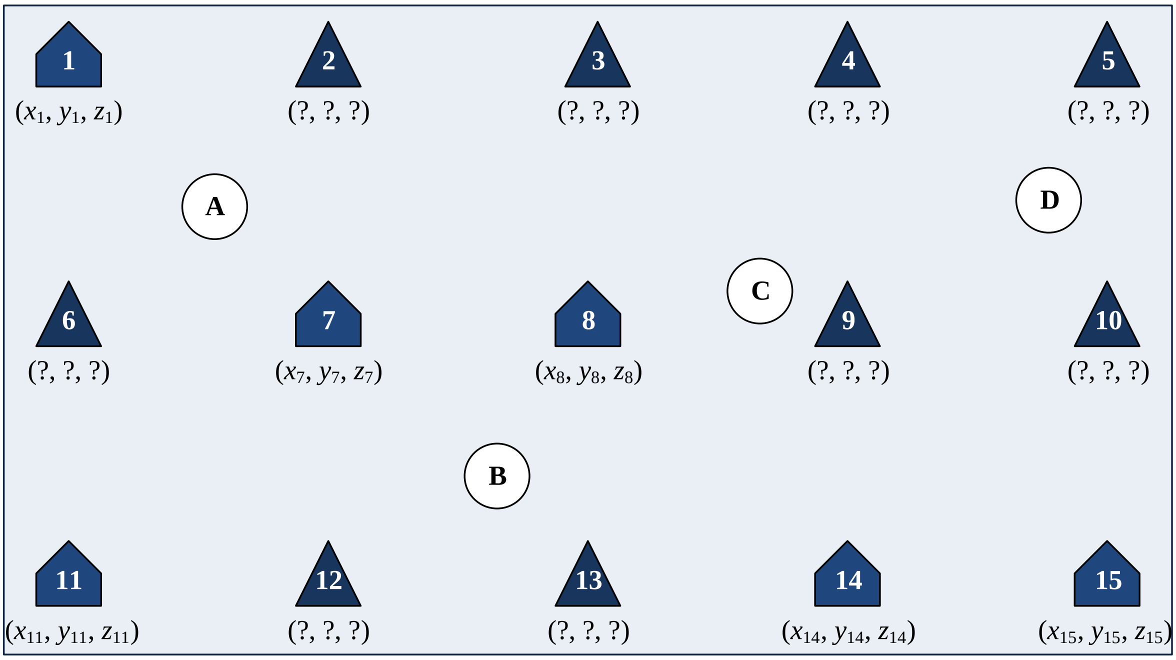
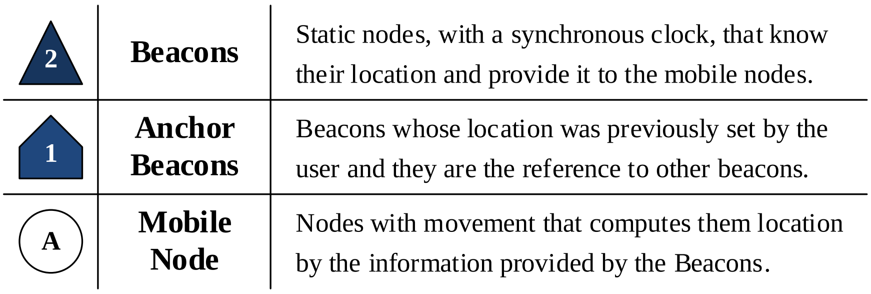


Figure 5.1: Example of the proposed ultrasonic location system architecture.



2	**Beacons**	Static nodes, with a synchronous clock, that know their location and provide it to the mobile nodes.
1	**Anchor Beacons**	Beacons whose location was previously set by the user and they are the reference to other beacons.
A	**Mobile Node**	Nodes with movement that computes them location by the information provided by the Beacons.

Figure 5.2: Different types of nodes present in the proposed architecture.

hand, increasing the maintenance complexity. In order to reduced the human intervention during the setup time and the maintenance complexity the network is composed by two different types of beacons: beacons with well known location and beacons that can compute their location by exchanging some US messages between their neighbors. Therefore, during the network setup it is only needed to determine by hand the location of a small set of beacons which could result in a considerable reduction of the setup time. Moreover, any node could be replaced by a new one without the need to determine its location which simplifies considerably the maintenance complexity.

The proposed location system can be divided in two distinct location processes that could run separately or simultaneously depending on the system's specifications or limitations:

1. Beacon Location Process;
2. Mobile Location Process.

5.1.1 Beacon Location Process

The beacon location process is schematically represented by the example of Figure 5.3. In this process each beacon exchanges some US messages with the near neighbors in order to synchronize their clocks and locate themselves. In the example of Figure 5.3 beacon 9 could compute its location by exchanging some US messages between neighbors: 3, 4, 5, 8, 10, 13, 14 and 15. This process has a very low refresh rate due the fact that beacons are considered to be motionless and therefore their location are constant over a long period of time. This process will be detailed later in Section 5.5 where the messages exchanged are reduce to the minimum in order to reduced the possible mobile location interference.

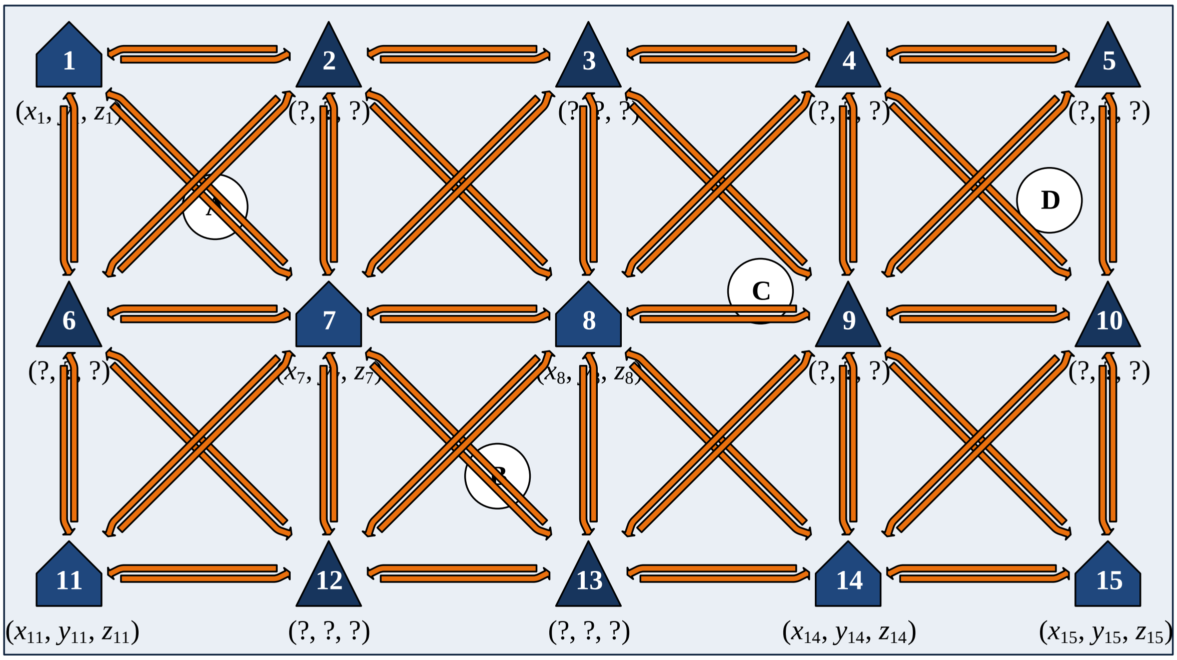


Figure 5.3: Beacon location process example.

5.1.2 Mobile Location Process

The mobile location process is schematically represented by the example of Figure 5.4. In this process each mobile is responsible for computing its own location using only US messages broadcast by the beacons network.

In the example of Figure 5.4 mobile node A computes its location by processing the information received from beacons: 1, 2, 6 and 7. By using this approach the proposed location system is completely independent of the number of mobile nodes. In addition to this, it provides privacy to the mobile node, which results from the fact that mobiles do not need to send any message. This process is completely independent of the beacon location process and assumes that beacons know their location precisely. Furthermore, it could be run at the same time as the beacon location process, nevertheless, the acoustic channel sharing complexity will increase. The refresh rate of this process must be much higher than that of the beacon location process. Since, mobile nodes present some kind

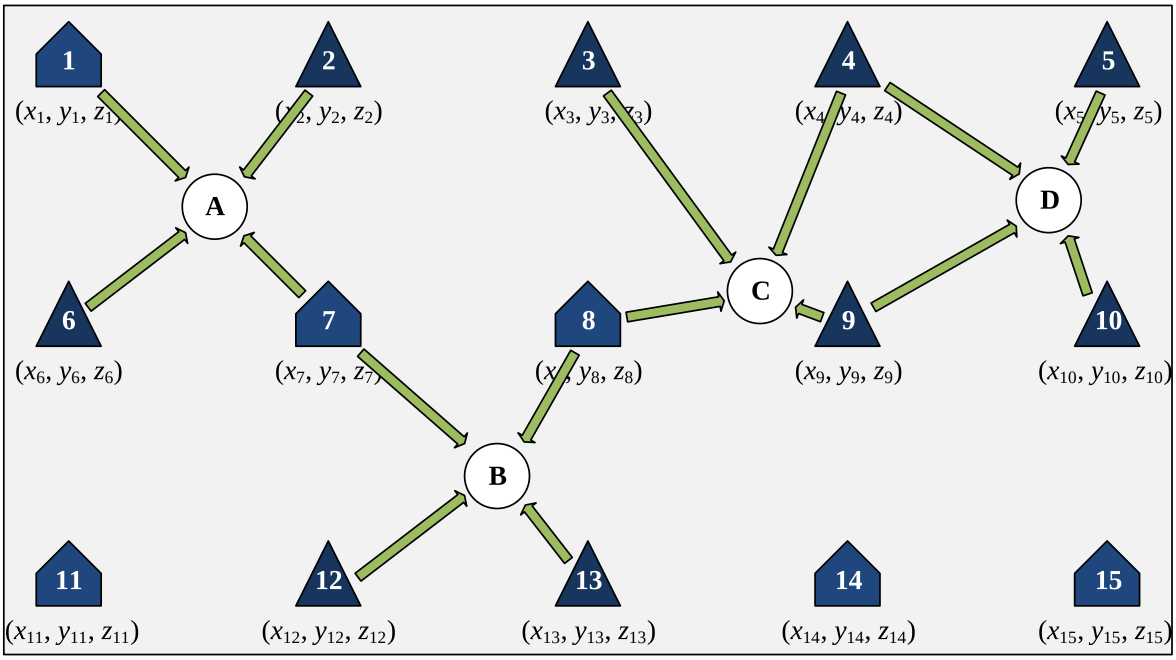


Figure 5.4: Mobile location process example.

of movement, and therefore, the location information must be obtained at useful time. This process will be detailed later in Section 5.5 where two different types of location algorithm will be used for the beacon location process and mobile location process.

5.1.3 Architecture Advantages and Disadvantages

As mentioned in the previous section, the proposed ultrasonic location system architecture presents several advantages, the main are summarized further:

Robust to RF interference: this is the first advantage that results from the system goal, which imposes that all possible communication between nodes must be carried out using ultrasounds;

Reduced setup time: since it only needs to set the coordinates to a small number of beacons (anchor beacons) the time that is spent during the setup is reduced considerably;

Unskilled maintenance: since any failure node can be replaced without needing to measure its coordinates this procedure may be performed by any unskilled worker. Additionally, if any beacon needs to be moved to other place this may also be performed by anyone;

High degree of scalability: the mobile nodes are responsible for computing its location by listening the messages from the network of beacons therefore there is not any limit in the number of mobiles. In addition to this, the beacon network could be easily expanded by only adding new beacons to the network;

User security and privacy: the mobile nodes do not need to send any message to estimate their location, they only listen. Therefore, it is impossible to locate, track or even sensing the presence of a mobile node.

Nevertheless, the proposed ultrasonic location system architecture presents some disadvantages, the main are summarized further:

Ultrasonic data communication: the data communication using ultrasounds by itself is a considerable disadvantage, especially if it is carried out in air. As it was pointed out in Chapter 3 and 4 the data communication will be strongly affected by the Doppler effect and multipath;

Low data bandwidth: as a result of the ultrasonic high frequencies attenuation the total bandwidth may be much less than 100 kHz. Additionally, the acoustic channel must be shared further reducing the available bandwidth;

Short range communication: as presented in Chapter 4 especially in Figure 4.15 due to air absorption the communication range is in the order of some tens of meter;

Difficult to track or locate users: as a result of the security and privacy orientation it will be difficult to track or locate the users which reduces the architecture versatility.

5.2 Clock Synchronization Between Beacons

In the proposed location system architecture each beacon is completed autonomous and therefore it has its own clock $\zeta_i(t)$. Ideally, each beacon's clock will be equal to the reference clock, $\zeta_i(t) = t$ [67, 68]. As a result of several imperfections in the oscillator, due to the production process and different surrounding environment, such as: temperature and humidity, the clock will usually be different of any two beacons [67]. Therefore, the clock of beacon i can be related to real time as follows [67, 68]:

$$\zeta_i(t) = \varpi_i t + \phi_i, \tag{5.1}$$

where ϕ_i and ϖ_i are the offset and skew of the beacons i's clock. The clock skew represents the frequency difference to the ideal clock ($\varpi_i = 1$), a faster clock will present a skew greater than one ($\varpi_i > 1$) and a slower clock less than one ($\varpi_i < 1$). Moreover, the offset represents the initial clock value at real time zero. A graphical representation of the clock offset and clock skew is depicted in Figure 5.5.

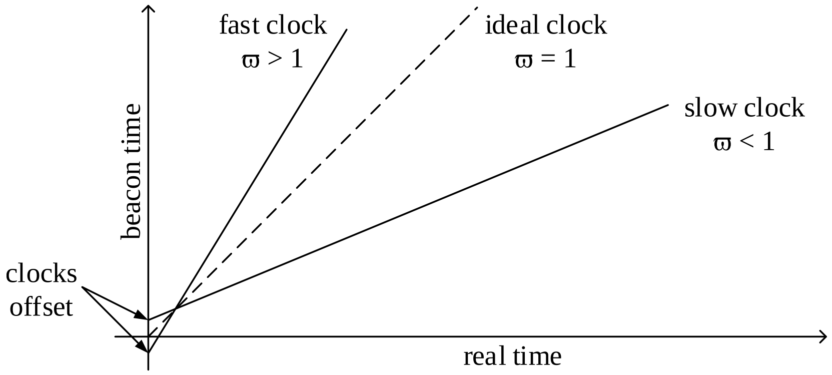


Figure 5.5: Beacons clock model and the ideal clock (adapted from [67]).

Regarding Equation 5.1, it is possible to compare the clocks of two beacons. The clock relation between beacon 1 and beacon 2 can be represented by:

$$\zeta_2(t) = \varpi_{1,2} t + \phi_{1,2}, \tag{5.2}$$

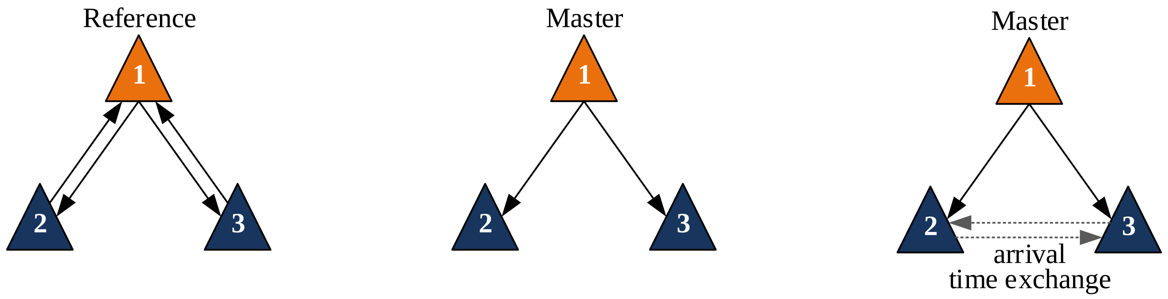
where $\phi_{1,2}$ is the relative offset and $\varpi_{1,2}$ is the relative skew between the clocks of beacon 1 and beacon 2. Ideally if both clocks are perfectly synchronized, $\phi_{1,2} = 0$ and $\varpi_{1,2} = 1$. Which means that, the two clocks have the same frequency and the same initial value. Based on both equations (5.1 and 5.2), the clock skew and offset of the location system beacons can be used to synchronize their local time [67]. Some traditional clock synchronization protocols, from wired networks, were designed to adjust the clock skew and offsets between the beacons achieving instantaneous synchronization. Unfortunately, in the proposed location system architecture, several non deterministic events can considerable affect the message transmission, turning the synchronization process much more difficult. Nevertheless, several clock synchronization protocols have been proposed for wireless sensor networks (WSNs)[1] [67].

5.2.1 Clock Synchronization in WSN

Usually, the clock synchronization process in WSN uses some timing message transmissions to estimate the relative clock skews and offsets. The way that the time messages are exchanged between beacons can be split in three different categories [67]:

1. two-way message exchange (or sender-receiver synchronization);
2. one-way message dissemination;
3. receiver-receiver synchronization.

Figure 5.6 presents a graphical interpretation of these three time messages exchanged mechanisms.



(a) Two-way message exchange.

(b) One-way message dissemination.

(c) Receiver-receiver synchronization.

Figure 5.6: The three ways that the time messages can be exchanged between beacons for clock synchronization.

[1] Note that: in the proposed location system architecture the beacons can be seen as a wireless sensor network.

Figure 5.6a: The two-way message exchange is one of the simplest processes of clock synchronization. In this process two messages are exchanged between the beacons. Usually, there is a reference beacon whose clock will be considered the network clock and therefore the other beacon's clock must be synchronized to this clock. Accordingly to this and regarding Figure 5.6a if beacon 1 is the reference beacon therefore beacon 2 and 3 must synchronize their clocks to beacon 1. To perform the clock synchronization beacon 2 sends a synchronization time message to beacon 1 with its time stamp ς_1, beacon 1 recorders its received time ς_2. After some processing time, beacon 1 replies to beacon 2 at time ς_3, which is received by beacon 2 at time ς_4. This process is presented in Figure 5.7. After several messages exchange cycles, beacon 2 can synchronize its own clock with beacon 1 and estimate the propagation time delay with some accuracy [67].

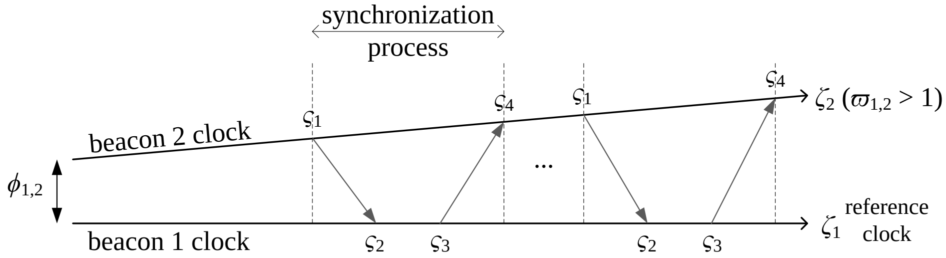


Figure 5.7: Time messages exchanged in the synchronization process between two beacons for the two-way message exchange.

Figure 5.6b: The one-way message dissemination presents a master-slave approach, where the master beacon regularly sends to other beacons (slaves) its clock information. Regarding Figure 5.6b the master beacon 1 periodically broadcast to beacon 2 and 3 its clock information and beacon 2 and 3 must synchronize their clock with beacon 1. The clock synchronization between two beacons is represented in Figure 5.8, where beacon 1 sends a synchronization time message to beacon 2 with its time stamp ς_1, beacon 2 records its received time ς_2. After repeating this process beacon 2 synchronizes its clock with beacon 1 [67]. Nevertheless, this technique cannot estimate the propagation time delay, and therefore, the algorithms based on this technique assume that the propagation time is negligible [69], which could only be considered true for RF communications. However, for US communications two beacons 1 m apart will introduce a propagation time delay around 3 ms and therefore it cannot be discarded. In addition to this, as seen at the beginning of this chapter, each beacon, to compute its own location, needs the distance between it and the other beacons. As it is clear, this distance can be easily computed by the propagation delay time.

Figure 5.6c: Finally, the receiver-receiver synchronization has the same approach that the one-way message dissemination where the master spreads its clock information. Additionally, the beacons can also synchronize with each other directly without the master intervention. This is achieved by sharing the arrival time messages between each other. At the end, the master can be removed from the WSN and the networks will continue the synchronization process without

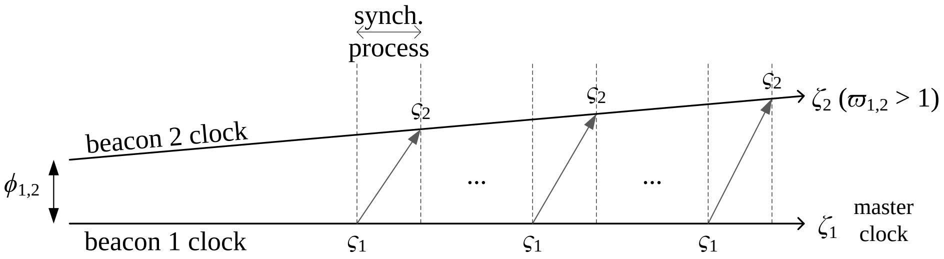


Figure 5.8: Time messages exchanged in the synchronization process between two beacons for the one-way message dissemination.

needing the master beacon [70]. Once again, this solution will also not be attractive to the proposed location system manly because it presents the problem of the time propagation delay absence [67].

5.2.2 Two-Way Message Exchange

In the proposed location system architecture where the two tasks, distance measurement and clock synchronization, have the same importance and impact on the location system, the two-way message exchange will the the best choice for clock synchronization. With this method, it will be possible to perform clock synchronization and distance measurement.

5.2.3 Two Beacon Synchronization

For two beacons synchronization analysis it will be considered that beacon 1 is the reference beacon and beacon 2 wants to synchronize its clock with beacon 1 (Figure 5.7 case). From [67] the beacon's clock synchronization process can be mathematically modeled as:

$$\varsigma_2 = \varpi(\varsigma_1 + \tau) + \phi_{1,2}, \tag{5.3}$$

$$\varsigma_3 = \varpi(\varsigma_4 - \tau) + \phi_{1,2}, \tag{5.4}$$

where, ϖ are the relative clock skew, ϕ the clock offset between beacon 1 and 2 and τ the propagation time delay. Considering that the clock skew of the beacons are almost equal and therefore $\varpi = 1$ then (5.3) and (5.4) can be simplified to:

$$\varsigma_2 - \varsigma_1 = \tau + \phi, \tag{5.5}$$

$$\varsigma_4 - \varsigma_3 = \tau - \phi. \tag{5.6}$$

As a result of this, the distance between the beacons, d, and the clock offset, ϕ, can be easily obtained from the time measurements:

$$\phi = \frac{(\varsigma_2 - \varsigma_1) - (\varsigma_4 - \varsigma_3)}{2}, \tag{5.7}$$

$$d = \frac{(\varsigma_2 - \varsigma_1) + (\varsigma_4 - \varsigma_3)}{2}c, \tag{5.8}$$

where, c is the ultrasonic wave speed in air. At the end, beacon 2 can correct its clock, by making it equal to beacon 1 and it also knows the distance to beacon 1. Note that it was assumed that the time measurements are noiseless, however in real environment this is not true. In order to make the location system robust to this errors, several synchronization cycles can be performed before beacon 2 performs its clock correction. As a consequence of this, beacon 2 will obtain a set of time stamps $[\varsigma_{1,k},\ \varsigma_{2,k},\ \varsigma_{3,k},\ \varsigma_{4,k}]_{k=1}^{K}$. If the time measurements were independent Gaussian distributed random variables, from [67, 71] the maximum likelihood estimator for the clock offset is given by:

$$\hat{\phi} = \frac{1}{2K}\sum_{k=1}^{K}(\varsigma_{2,k} - \varsigma_{1,k}) - (\varsigma_{4,k} - \varsigma_{3,k}), \tag{5.9}$$

which means that the clock offset estimation is independent of the propagation time. Moreover using this approach the maximum likelihood estimator for the distance is given by:

$$\hat{d} = \frac{c}{2K}\sum_{k=1}^{K}(\varsigma_{2,k} - \varsigma_{1,k}) + (\varsigma_{4,k} - \varsigma_{3,k}), \tag{5.10}$$

which means that the distance estimation will be independent of the clock offset.

5.2.4 Wide Beacons Spread Synchronization

There are several protocols based on the chosen method [67, 68]. Nevertheless, the main difference between them is the algorithm generalization process, passing from two beacons to several beacons. Moreover, they also present some differences to overcome the connectivity problem between beacons, such as, the absence of connectivity to the reference beacon.

One of the mostly used synchronization protocols based on the chosen method is the TPSN proposed by Saurabh Ganeriwal, et. al, in [72, 9]. The authors proposed this protocol as a flexible extension of the widely used Network Time Protocol to be applied in WSN. This protocol is a well studied protocol [67, 68] and, as consequence of this, several small changes and improvements were introduced in the past few years to increase its accuracy, introducing new features (e.g. security) or improving the fault tolerance [9, 73, 74, 75, 76, 77]. In spite of small differences between the several proposed protocols they share the same core and operation principle and therefore they are more or less the same protocol. As a consequence of this it was decided to only present the main idea behind the TPSN protocol as it was discussed in [9].

The TPSN has two phases of operation:

1. Level Discovery Phase;
2. Synchronization Phase.

Level Discovery Phase

This is the first phase of the algorithm and must be carried out when the network is deployed. In this phase a hierarchical or tree network topology is build, where each beacon will be assigned to a level. One beacon is elected or set to be the reference beacon (or root beacon) and therefore other beacons must synchronize their clock with the reference beacon[2]. Level 0 is assigned to the reference beacon which is responsible to launch this phase by broadcasting a *level discovery* packet. This packet contains its identity and level. All the beacons in the neighborhood that receive this packet will assign themselves the next level in hierarchy i.e., level 1. After that, these beacons also broadcast a *level discovery* packet with their level and identities. All the beacons in the neighborhood that receive this packet and do not have a level assigned will assign themselves the next level in hierarchy i.e., level 2. This process will continue until all the the beacons in the network have an assigned level in the hierarchy. Figure 5.9 describes how the process of the level discovery phase is unfolded. Note that, after the level discovery phase a hierarchical network is created and all the beacons in the hierarchy can only exchange messages with beacons in the previous or next level. This means that a beacon in level 3 can only exchange messages with beacons in levels 2 and 4.

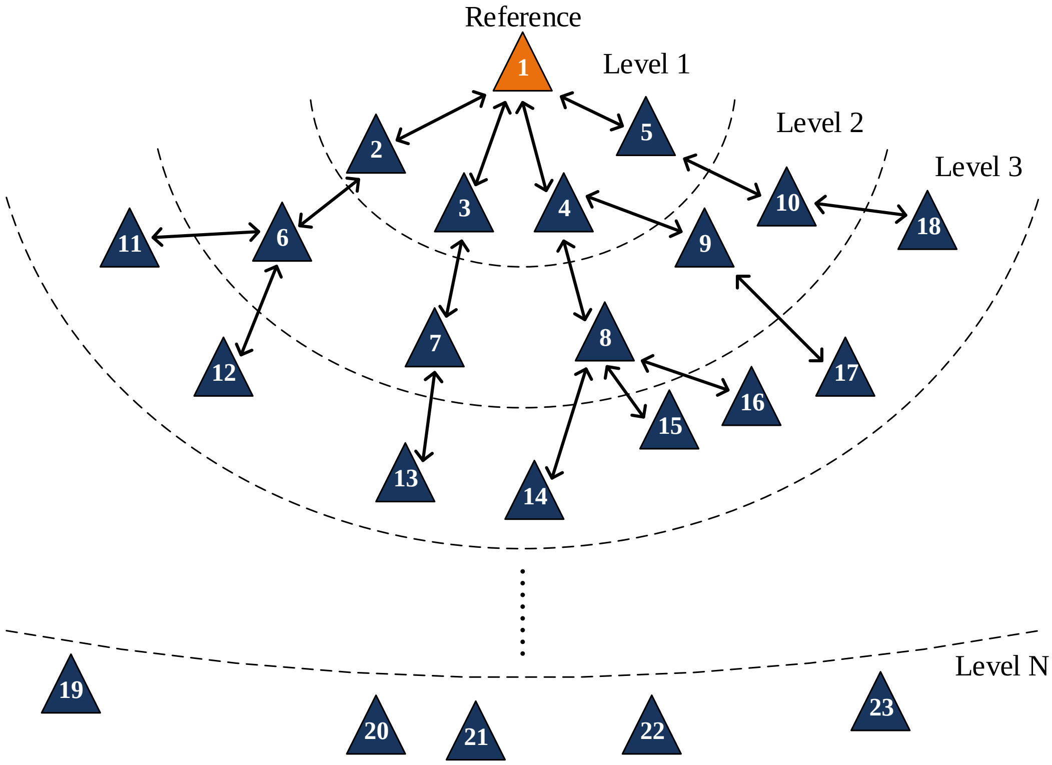


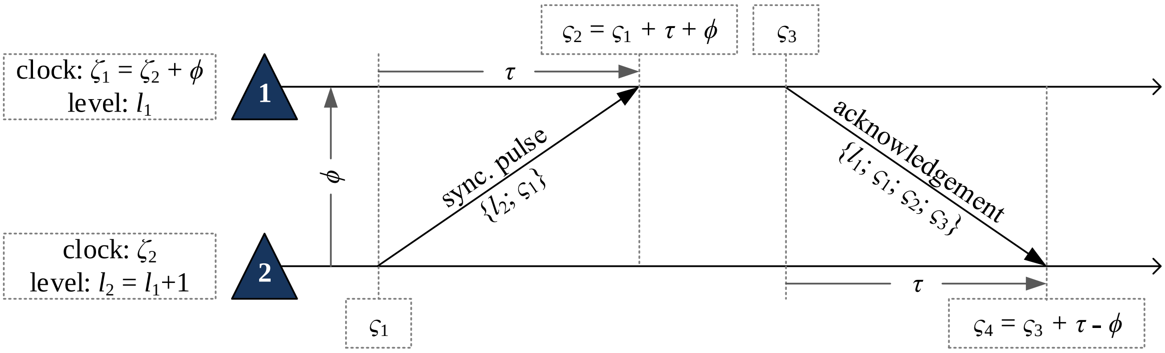
Figure 5.9: Connectivity between beacons in the level discovery phase.

Synchronization Phase

In the synchronization phase beacons synchronize their clock with beacons in the level above them, i.e. beacons with level l synchronize their clock with beacons with level $l - 1$. This phase is initiated by the reference beacon that broadcasts a *synchronization pulse* packet. After receiving this packet beacons from level 1 wait a random time before starting the two-way message exchange. This approach will reduce the probability of message collision and reduce the network congestion. After

[2] Although it is not entirely necessary still the reference beacon could have an accurate clock that can be provided by an external time reference, such as GPS [70].

receiving the *acknowledgment* message from the reference beacon, beacons of level 1 can adjust their clocks turn them identical to the reference clock. Beacons from level 2 know that beacons of level 1 are in synchronization process because they listen them asking for clocks. And therefore, they will wait some random time, greater than a certain amount of time to ensure that beacons of level 1 ended their synchronization process, before starting the two-way message exchange with beacons of level 1. Thereafter, beacons of level 2 can turn their clocks identical to the level 1 beacon's clock. This process will continue to the remaining levels until the last level. The message exchange between two consecutive levels is depicted in Figure 5.10.



$\varsigma_2 = \varsigma_1 + \tau + \phi$
ς_3
clock: $\zeta_1 = \zeta_2 + \phi$
level: l_1
1
τ
ϕ
sync. pulse
$\{l_2; \varsigma_1\}$
acknowledgement
$\{l_1; \varsigma_1; \varsigma_2; \varsigma_3\}$
clock: ζ_2
level: $l_2 = l_1$+1
2
τ
ς_1
$\varsigma_4 = \varsigma_3 + \tau - \phi$

Figure 5.10: Messages exchanged between two beacons in the synchronization phase.

Beacon 2 of level l_2 synchronizes its clock with a beacon of level $l_2 - 1$ (Beacon 1) by two-way message exchange. Beacon 2 sends a *synchronization pulse* packet with its level and its local sent time ς_1. Beacon 1 receives the packet at time ς_2, which is given by:

$$\varsigma_2 = \varsigma_1 + \tau + \phi, \tag{5.11}$$

where ϕ is the clock offset between the beacons and τ the propagation time delay. Then, after some processing time ($\varsigma_3 - \varsigma_2$), beacon 1 answers back to beacon 2 sending an *acknowledgment* packet at time ς_3, with its level and all the time information (ς_1, ς_2 and ς_3). Beacon 2 receives this packet at time ς_4, which is given by:

$$\varsigma_4 = \varsigma_3 + \tau - \phi. \tag{5.12}$$

At the end of this process beacon 2 can easily compute ϕ and τ and consequently d using (5.7) and (5.8) respectively.

Fault Tolerance

There are several scenarios that the protocol needs to deal with, but the most important are the following:

1. A beacon did not obtain a level assigned during the level discovery phase;
2. Add a new beacon to the network in synchronization phase;
3. A beacon of level l dies and some beacons from level $l + 1$ cannot synchronize their clocks;
4. The reference beacon dies.

The first two scenarios are very similar, and therefore the system can treat them the same way. During synchronization phase a beacon without level information can wait for some time to receive a *level discovery* packet if after this timeout it did not receive any *level discovery* packet, it broadcast a *level discovery* message. The neighbors reply to this message by sending their own level in the hierarchy. After that, the beacon can assign itself a level in hierarchy, which must be one position above the level that it has received. The authors call this process the *local level discovery* phase [9].

The third scenario could arise from the fact that a beacon is not receiving an *acknowledgment* for a sent *synchronization pulse*. Before, assuming that the beacon above it in the hierarchy is dead, it must try to re-transmit a *synchronization pulse* after some random amount of time. This approach is important because not receiving an *acknowledgment* could be due to a message collision. Only after some trials if the beacon continues without receiving the *acknowledgment* it can assume that the beacon above it in the hierarchy is dead. And therefore, it assumes that it does not have a valid level and broadcasts a *level discovery* message and by the replies it assigns itself a level in hierarchy similarly to the previous scenario.

The last scenario, and the most crucial, is how to deal with a dead reference beacon. If the reference beacon dies the beacons from level 1 cannot synchronize their clock and therefore neither the beacons in the remaining levels. Unfortunately, it not possible to follow the same approach of the previous scenario. The solution that the authors point out is that they could run a leader election algorithm and the elected leader will be considered the new reference beacon. As a consequence of this, it must start a new level discovery phase to create the new hierarchy.

5.3 Distance Measurement

In the previous section it was presented the TPSN that can provide not only the clock synchronization between beacons but also the distance measurement between them. Nevertheless, the TPSN can only provide distance measurement between it and the beacon that it is synchronizing with. Which may not be enough to run the location system. Moreover, as we will see later, location algorithm can improve the beacon location estimation not only by knowing the distance to the neighbors but also by knowing the distance between them. As a consequence of this, a very simple modification to the TPSN algorithm is proposed, it will allow a beacon to know not only the distance to its neighbors but also the distance between some of them. The level discovery phase will continue to be the same but the synchronization phase will be modified. From the previous section, the data content in the *acknowledgment* packet is enough for the beacon to compute the distance to the beacon that it exchanging messages with it. Nevertheless, this data will not be enough to the location algorithm. The beacon also needs to know the coordinates of the beacon that is exchanging messages. Therefore, the *acknowledgment* packet must include an additional field with the beacon coordinates. In addition to this, the two-way message exchange must be followed by a message broadcast with the message time stamp, the beacon's IDs, the coordinates of both beacons and the distance between them. This message will inform the neighbor beacons of these two beacons positions and their coordinates. The message needs to contain the ID information because this will allow the neighbors to update the correct coordinates in their local databases. Moreover, the time stamp will permit neighbors to compute the distance to this beacon by time difference between the sent message and the received message. Note that, this distance measurement can only be performed after both clocks are synchronized. As we will see later this mechanism will increase the accuracy of the beacon position estimation. The message exchange between two consecutive levels in the modified TPSN is depicted in Figure 5.11. Beacon 2 of level l_2 synchronizes its clock with a beacon of level $l_2 - 1$ (beacon 1) by two-way message exchange. Beacon 2 sends a *synchronization pulse* at ς_1 and beacon 1 receives it at ς_2 similar to the original TPSN algorithm. Then, beacon 1 responds back to beacon 2 sending an *acknowledgment* packet at ς_3, with its level and all the time information (ς_1, ς_2 and ς_3) in addition to the original method beacon 1 also sends its estimated coordinates. Beacon 2 receives this packet and can easily compute its new clock and the distance to beacon 1. At the end, beacon 2 broadcasts the time stamp ς_5 (using the new synchronized clock), its IDs and coordinates C_2, the ID and coordinates C_1 of beacon 1 and the distance between it and beacon 1, $d_{1,2}$.

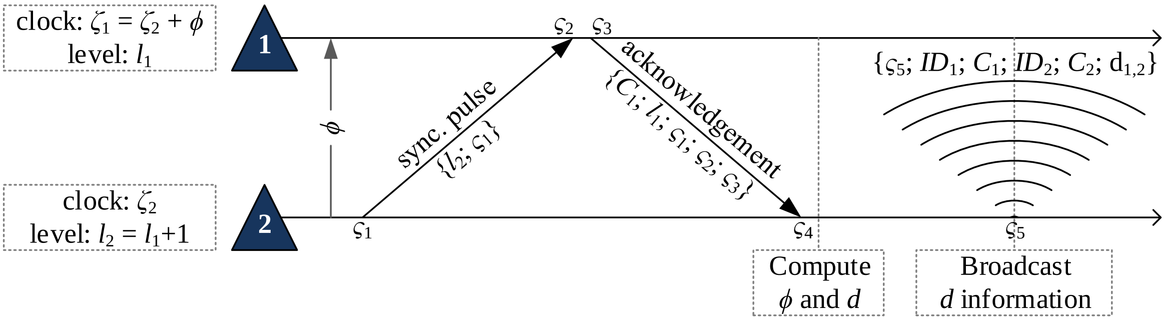


Figure 5.11: Messages exchanged between two beacons in the synchronization phase for the modified TPSN.

To better understand this modification and present its huge advantages in spite of introducing a 50% more overhead in the synchronization process (from two to three messages) the example in Figure 5.12 will be used. In this example 3 different levels apart from the reference beacon will be considered. One can see that by employing the traditional TPSN method all the beacons will obtain

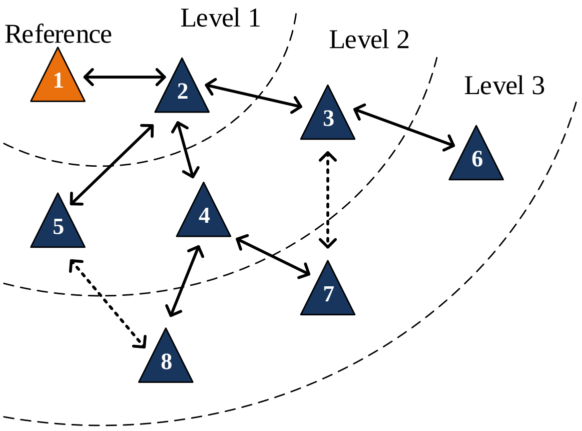
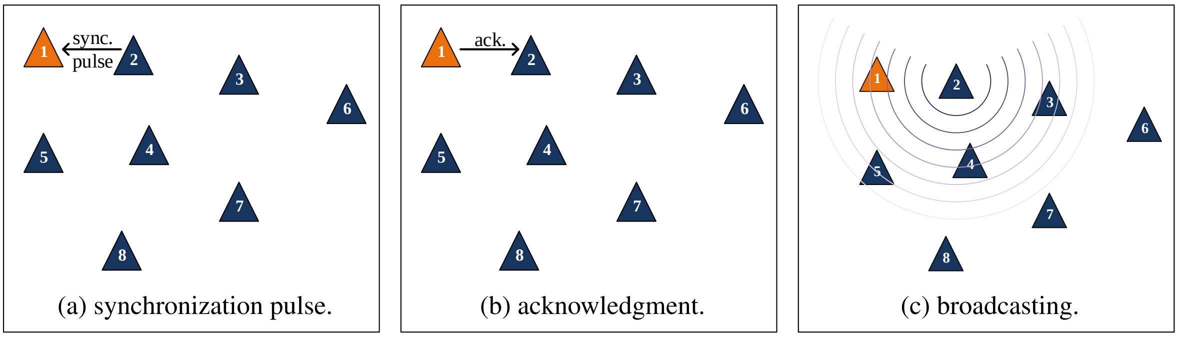


Figure 5.12: Example of a messages exchanged between beacons in the synchronization phase. The solid line represents frequent connectivity and the dashed lines sporadic connectivity.

only one distance measure without knowing the coordinates of that particular beacon, except beacon 7 and 8 that can obtain two distance measures and beacon 1 that does not get any distance measure at all.

Figure 5.13 presents the modified TPSN for beacon 2. After receiving the *acknowledgment* message (Figure 5.13b) from beacon 1, beacon 2 can compute the distance to beacon 1 and broadcast this distance to its neighbors. It can be seen in Figure 5.13c that this broadcasted message can be received by beacons 1, 3, 4 and 5.



(a) synchronization pulse.

(b) acknowledgment.

(c) broadcasting.

Figure 5.13: Example of the three messages exchanged for beacon 2 synchronization.

Table 5.1 presents the results for the remaining beacons during the synchronization process, and Table 5.2 presents the distances obtained by each beacon. One can see that, even for short range connectivity, beacons can get easily several distance measurements. Moreover, the obtained distances are not only between them and their neighbors but also the distances between the neighbors. For example, beacon 3 will know its distance to beacon, 2, 4, 6 and 7 and in addition it will also know the distance between beacon 1 and 2, beacon 2 and 4, beacon 4 and 7 and beacon 4 and 8.

Event	**Results after the synchronization**
Beacon 2 Sync.	Beacon 2 broadcasts the distance between it and beacon 1; this message will be received by the neighbors: beacons 1, 3, 4 and 5
Beacon 3 Sync.	Beacon 3 broadcasts the distance between it and beacon 2; this message will be received by the neighbors: beacons 2, 4, 6 and 7
Beacon 4 Sync.	Beacon 4 broadcasts the distance between it and beacon 2; this message will be received by the neighbors: beacons 2, 3, 5, 7 and 8
Beacon 5 Sync.	Beacon 5 broadcasts the distance between it and beacon 2; this message will be received by the neighbors: beacons 1, 2, 4 and 8
Beacon 6 Sync.	Beacon 6 broadcasts the distance between it and beacon 3; this message will be received by the neighbors: beacons 3 and 7
Beacon 7 Sync.	Beacon 7 broadcasts the distance between it and beacon 4; this message will be received by the neighbors: beacons 3, 4, 6 and 8
Beacon 8 Sync.	Beacon 8 broadcasts the distance between it and beacon 4; this message will be received by the neighbors: beacons 4, 5 and 7

Table 5.1: The results of the message broadcasting after each beacon synchronization.

Beacon	Distance Information	Total
1	{1 – 2}; {1 – 5}; {2 – 5}	3
2	{1 – 2}; {2 – 3}; {2 – 4}; {2 – 5}	4
3	{1 – 2}; {2 – 3}; {2 – 4}; {3 – 4}; {3 – 6}; {3 – 7}; {4 – 7}; {4 – 8}	8
4	{1 – 2}; {2 – 3}; {2 – 4}; {2 – 5}; {3 – 4}; {4 – 5}; {4 – 7}; {4 – 8}	8
5	{1 – 2}; {2 – 4}; {2 – 5}; {4 – 5}; {4 – 8}; {5 – 8}	6
6	{2 – 3}; {3 – 6}; {4 – 7}; {6 – 7}	4
7	{2 – 3}; {2 – 4}; {3 – 6}; {3 – 7}; {4 – 7}; {4 – 8}; {6 – 7}; {7 – 8}	8
8	{2 – 4}; {2 – 5}; {4 – 7}; {4 – 8}; {5 – 8}; {7 – 8}	6

Table 5.2: The collected distances between beacons in each beacon after all beacons synchronization. $\{x - y\}$ – distance between beacon x and y.

5.4 Clock Synchronization Results

To exemplify how the synchronization process works and to present its performance, a network of wide spread beacons will be used (see Figure 5.14). This network is composed by 100 beacons forming hexagonal shape cells where each beacon has six neighbors at 4 m of distance. In additions to this, the network was arranged in order to cover a room with 18 m×66.5 m. Moreover, it is considered that the maximum range of each beacon is about 5 m as depicted by the circle in Figure 5.14 for beacon 13 range.

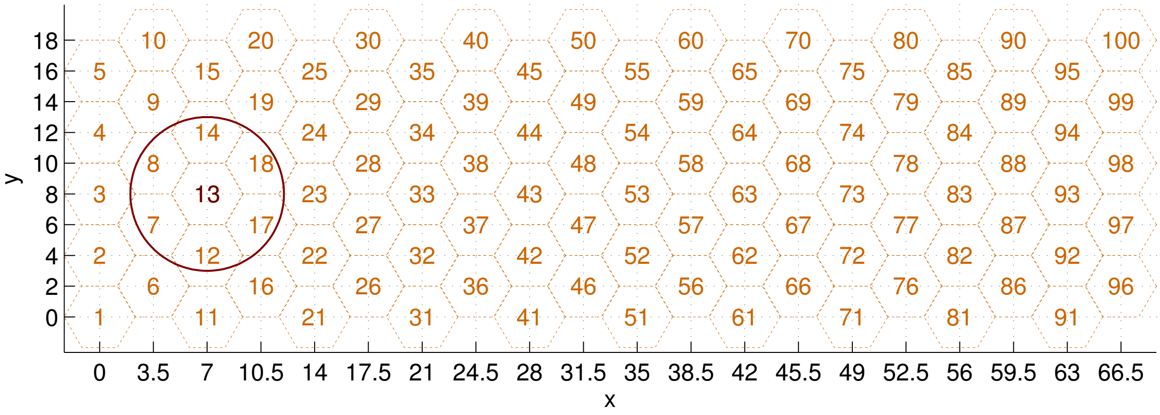


Figure 5.14: Network beacons in a hexagonal shape cells for clock synchronization in a room with 18 m × 66.5 m. Each beacon presents a maximum range of 5 m and it is represented by a circle for beacon 13.

5.4.1 Level Discovery Result

As a result of the short range communication, each beacon can only exchange US messages with its near neighbors. Resulting in a connectivity of six beacons in the center of the network and two in the networks corners (beacon 1 and 100). Additionally, to represent the worst hierarchy[3] scenario the reference will assigned to beacon 1 (in the network corner).

As a result of beacon 1 being the reference it has level 0 in the hierarchy and applying the proposed algorithm beacon 1 broadcasts a *level discovery* packet. This packet is only received by beacon 2 and 6 resulting in level 1 assignment in the hierarchy. After that, beacon 2 and 6 broadcast a *level discovery* packet which is received by beacons 3, 7, 11 and 12 (level 2 in the hierarchy). This process continues until beacons 96 to 100 receive level 19 in the hierarchy. The resultant level for each beacon is presented in Figure 5.15.

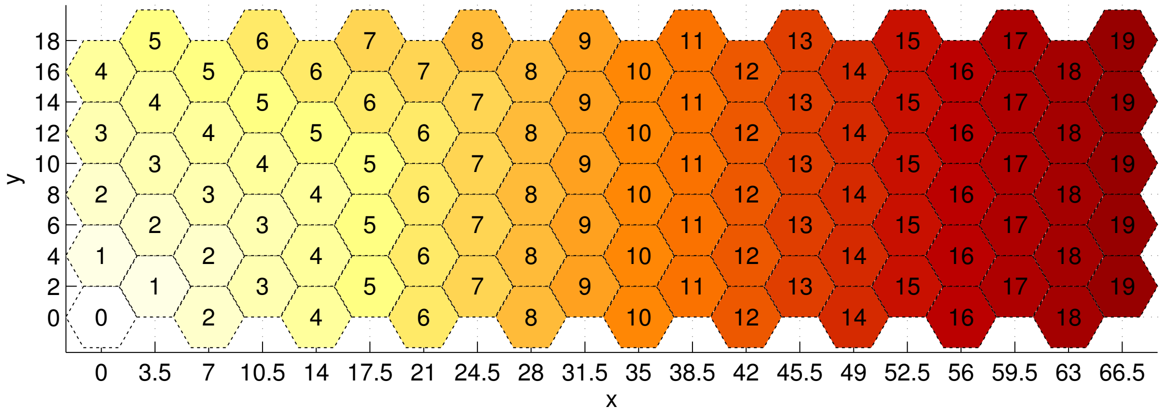


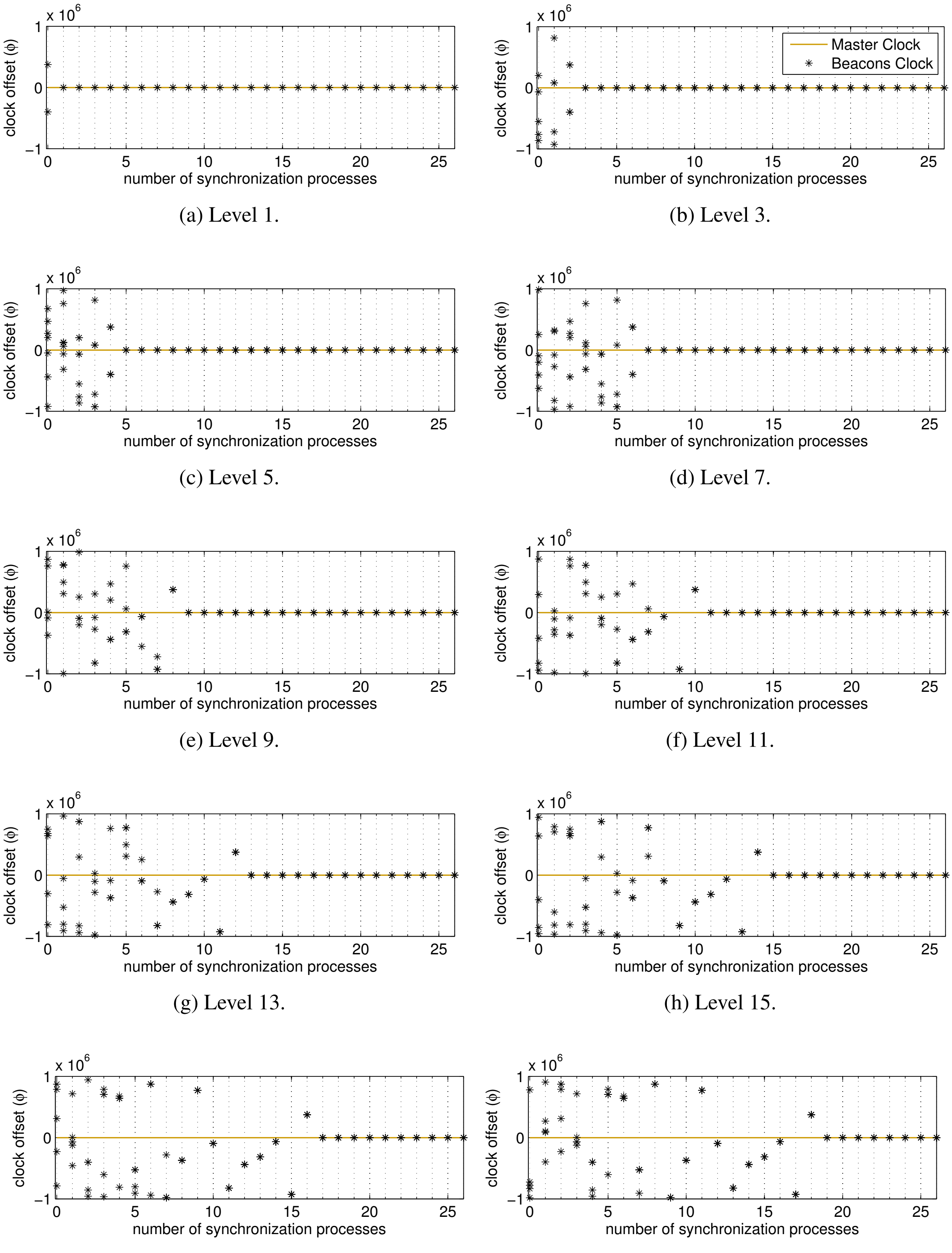
Figure 5.15: Hierarchy level of each beacon in the networks for beacon 1 as reference beacon. Each beacon displays its hierarchy level in the network.

5.4.2 Ideal Synchronization

After the level discovery phase, the beacons could start synchronizing their clocks with the reference beacon. Figure 5.16 presents the clock synchronization results for beacons in different hierarchy levels considering an ideal synchronization process. Where it is considered that after the synchronization process each beacon could synchronize their clocks without any error. Moreover, it was considered that the initial clock offset of each beacon relatively to the reference beacon is an uniform random variable between zero and one million seconds. One can see, from the results, that beacons of level l only obtain their clocks synchronized after l synchronization process. This means that beacons of level 1 synchronize their clock in the first synchronization process, whereas, beacons of level 2 can only synchronize their clocks correctly in the second synchronization process due to the fact that during the first synchronization process, beacons of level 1 had wrong clock values.

As a result of this clock synchronization delay, the time between synchronization processes must be as small as possible especially during the network setup. For example, if the clock synchronization is carried out every 10 min, beacon 100 needs to wait one hour and ten minutes to get its clock perfectly synchronized. Nevertheless, the number of levels can be easily reduced to half if the reference beacon is in the center of the network instead of being in the network corner, as can be seen in Figure 5.17 where the reference beacon was set as beacon 48.

[3] The worst hierarchy is considered to be the one with the greatest number of levels.



(a) Level 1.

(b) Level 3.

(c) Level 5.

(d) Level 7.

(e) Level 9.

(f) Level 11.

(g) Level 13.

(h) Level 15.

(i) Level 17.

(j) Level 19.

Figure 5.16: Clock results after the synchronization process for each odd level in the hierarchy for ideal synchronization process.

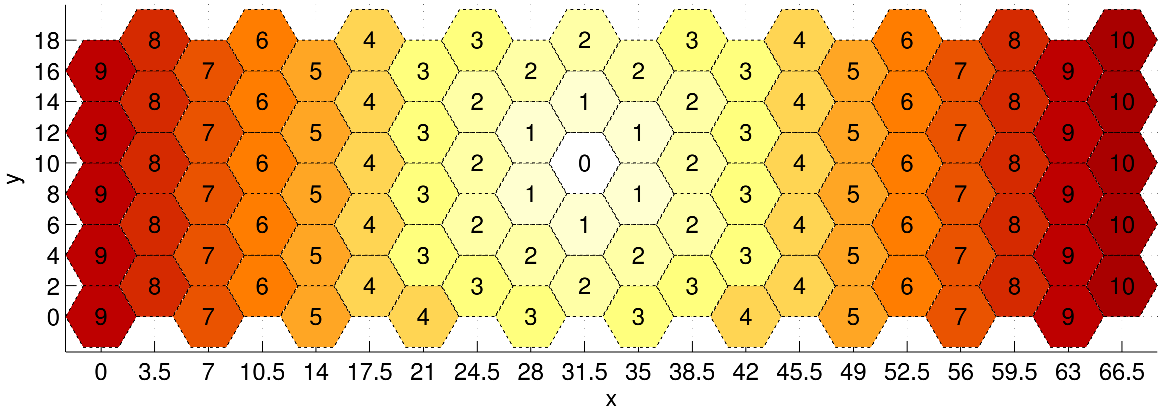
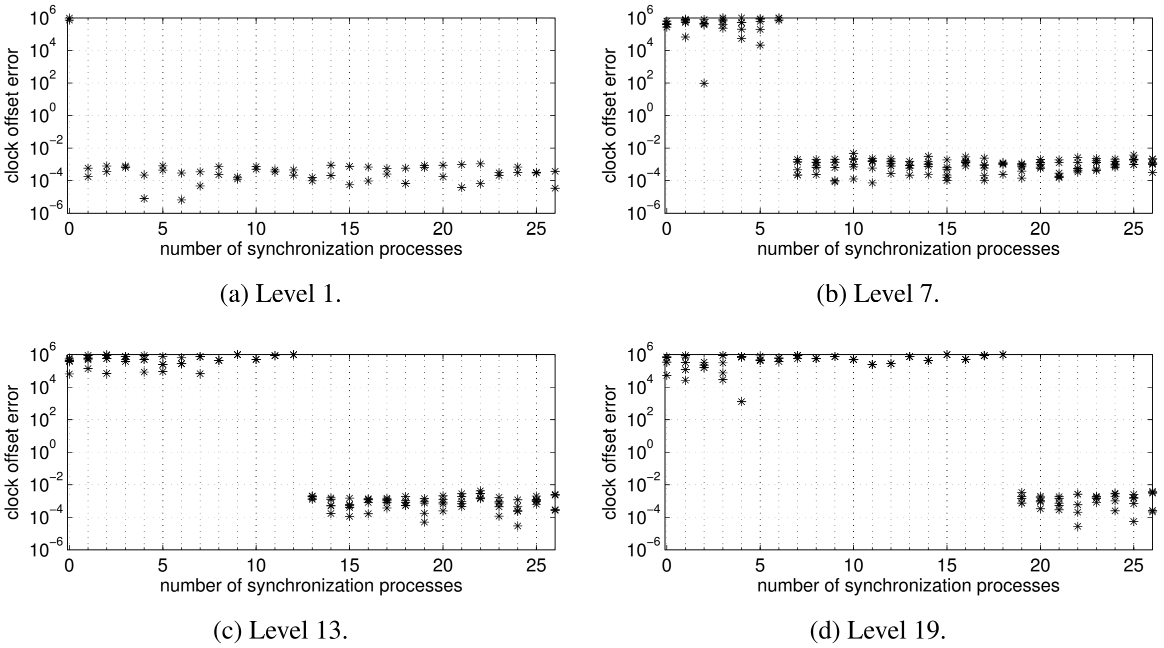


Figure 5.17: Hierarchy level of each beacon in the networks for beacon 48 as reference beacon. Each beacon displays its hierarchy level in the network.

5.4.3 Impact of the TOF Estimation Error

It was verified, in Chapter 4, that by using the threshold technique, the TOF estimation could present some error due to the fact that there is some probability of detecting the pulse around the true arrival position. The example of the previous section is also used here in order to evaluate the impact of this error on the clock synchronization results. Figure 5.18 presents the clock synchronization results for beacons in different hierarchy levels considering the time of flight as an independent Gaussian random variable with mean equal to the true TOF and 1 ms of standard deviation. One can see, from the results, that after the minimum number of synchronization processes the clock offset error is reduced to a minimum of the same order of the estimation error. However, the clock offset



(a) Level 1.

(b) Level 7.

(c) Level 13.

(d) Level 19.

Figure 5.18: Clock offset error relative to the master clock as a function of the synchronization processes for four different levels in the hierarchy, considering a time of arrival error standard deviation of 1 ms.

error increases as the level in the hierarchy also increases due to the propagation of error. Beacons from level 1 will synchronize their clocks with a clock without error but the time measurements will present a standard deviation error of 1 ms. From Equation 5.7 the standard deviation error of the estimation offset will be half of the time measurements error due to the fact that only measurements ς_2 and ς_4 are estimated. Furthermore, beacons from level 2 will synchronize their clocks, not only, with the same time measurements error, but also, they are synchronizing with a clock that presents a standard deviation error of half of the time measurements standard deviation error. Therefore, the resulting standard deviation error will be $\sqrt{2}/2$ of the time measurements standard deviation error. As a consequence of this error propagation the clock offset standard deviation error will be given by:

$$\sigma_l = \sigma_t \frac{\sqrt{l}}{2} \tag{5.13}$$

where σ_t is the time measurement standard deviation error. Figure 5.19 presents the clock offset standard deviation error as a function of the level in the hierarchy. It also presents the error of a simulation of one million synchronization processes. As a result of this, the clock offset standard deviation error will be proportional to the square root of the level in the hierarchy. Therefore, it is desirable to keep the number of levels as low as possible. In addition to this, the clock offset error may also be reduced by averaging several time measurements as proposed in Section 5.2.3.

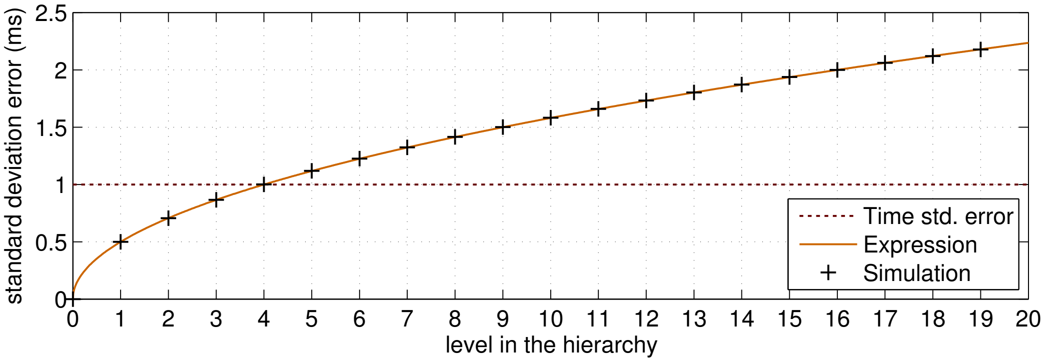


Figure 5.19: Standard deviation of the clock error in each level in the hierarchy for a million of synchronization processes, considering a time of arrival error with a standard deviation of 1 ms.

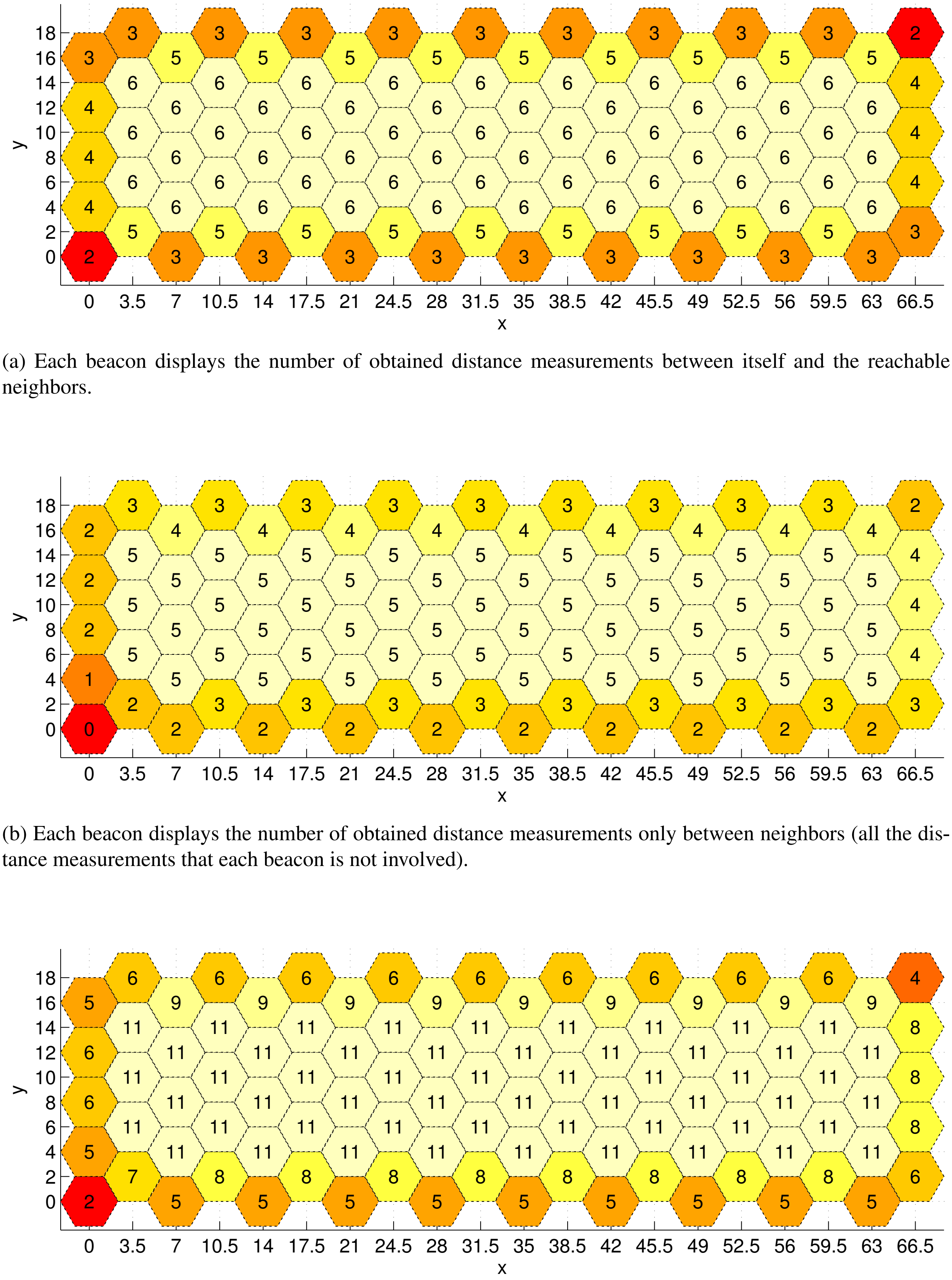
5.4.4 Distance Measurements and Network Position

To verify the beacon position influence in the number of distance measurements obtained, to be used for location proposed, it was considered the same example of the previous sections where beacon 1 is considered to be the reference beacon. Figure 5.20 presents the number of distance measurements obtained in each beacon divided in three classes:

(a) Number of obtained distance measurements **between itself and the reachable neighbors**;

(b) Number of obtained distance measurements **only between neighbors**, all the distance measurements that it is not involved;

(c) **The total number of distances measurements**, it is the sum of the two previous values.



(a) Each beacon displays the number of obtained distance measurements between itself and the reachable neighbors.

(b) Each beacon displays the number of obtained distance measurements only between neighbors (all the distance measurements that each beacon is not involved).

(c) Each beacon displays the total number of obtained distance measurements.

Figure 5.20: Distance measurements obtained in each beacon divided in three different classes. Considering beacon 1 (Figure 5.14) as the reference node.

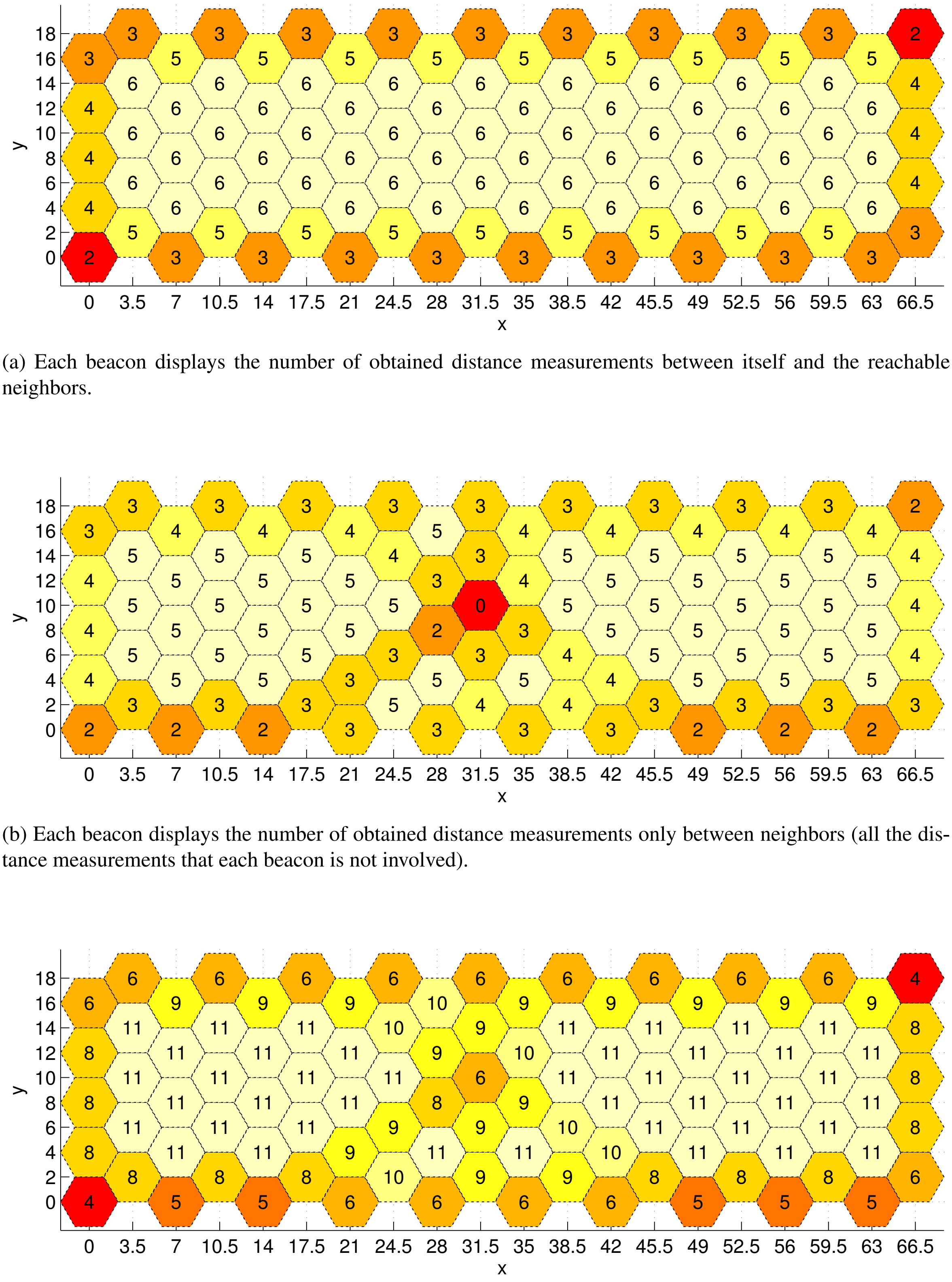
We can see that in the network center each beacon can easily obtain a total of eleven distance measurements, six of them to its neighbors and five between the neighbors. The first six distance measurements would be more than enough to compute the beacon position but the remaining five extra distance measurements may be used to improve the accuracy of this estimation. However, these numbers reduce considerably in the network edges, especially in the network corners, where the distance measurements could not be enough to obtain a proper location estimation. This results from the fact that due to the short range, beacons in the network corner can only communicate with few beacons. This problem can be easily overcome by applying one or both of the following strategies:

1. The network corners are only composed by anchor beacons, and therefore, their location are previously, known;

2. Near the corner, extra beacons are placed between beacons in the regular network, increasing the total number of possible beacons that a beacon in the corner could communicate with. For example, if in the previous beacons network three extra beacons are placed between beacon 1 and beacon 2 and 6, beacon 1 will also communicate with these extra beacons, increasing the total number of distance measurements from two to five.

In order to evaluate the influence of the reference beacon position on the number of distance measurements obtained the same example of the previous sections was considered, but now considering beacon 48 as the reference beacon. Figure 5.21 presents the number of distance measurements obtained in each beacon divided in the same three classes. As one can see, the number of distance measurements to neighbors is exactly the same of the previous example. In contrast, the number of distance measurements around the reference beacon decreases. This is manly due to the fact that the hierarchy is expanded from this point out. Which interferes into the process of obtaining distance information between neighbors because the probability of they having the same level in the hierarchy is very high and therefore they do not synchronize between them. It is also important to point out that, in some cases, the number of distance measurements obtained in the network edges has increased, especially where the reference beacon was located.

In conclusion, in order to reduced the error produced by the time of arrive estimation and maximizing the number of distance measurements obtained by each beacon the reference beacon must be located in the network center. Furthermore, it is recommended to place the anchor beacons in the network edges especially in the corners where the probability of receiving enough distance information to perform the beacon location is low.



(a) Each beacon displays the number of obtained distance measurements between itself and the reachable neighbors.

(b) Each beacon displays the number of obtained distance measurements only between neighbors (all the distance measurements that each beacon is not involved).

(c) Each beacon displays the total number of obtained distance measurements.

Figure 5.21: Distance measurements obtained in each beacon divided in three different classes. Considering beacon 48 (Figure 5.14) as the reference node.

5.4.5 Beacons Maximum Range Impact

The beacon maximum range has a strong impact in the structure of the hierarchy and in the number of distance measurements that each beacon could obtain. To exemplify this statement the previous beacon network will be used but now increasing the beacon range to the double, as depicted in Figure 5.22.

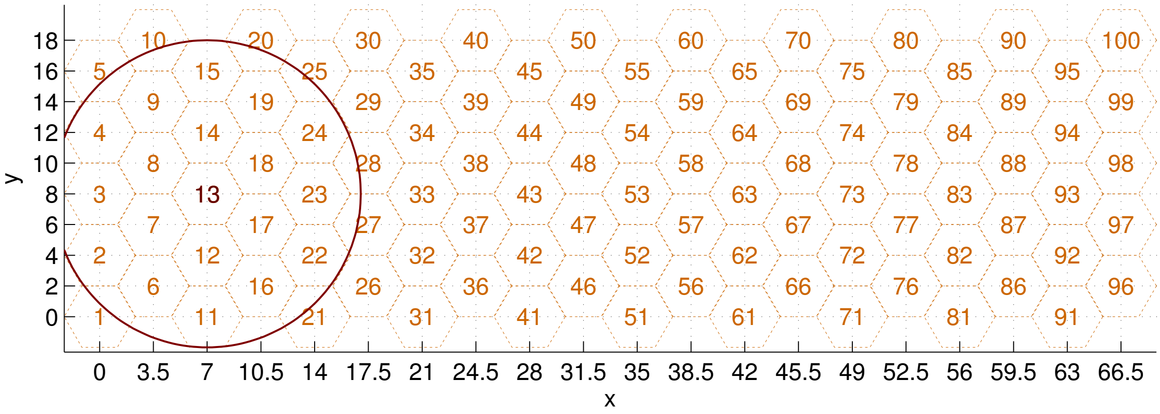


Figure 5.22: Network beacons in a hexagonal shape cells for clock synchronization in a room with 18 m × 66.5 m. Each beacon presents a maximum range of 10 m and it is represented by a circle for beacon 13.

One can see that, for example, beacon 13 extended the number of beacons that could exchange US messages with, from six to eighteen. As a result of this, the number of levels in the hierarchy decreases and consequently the number of beacons in each level increases. This effect can be observed in Figure 5.23 and Figure 5.24 where the reference role was set to beacon 1 and beacon 48 respectively. In both examples, the total number of levels in the hierarchy is reduced to half of the examples presented in Figure 5.15 and 5.17. This reduction in the number of level may decrease the settling time and the clock synchronization error (see Sections 5.4.2 and 5.4.3).

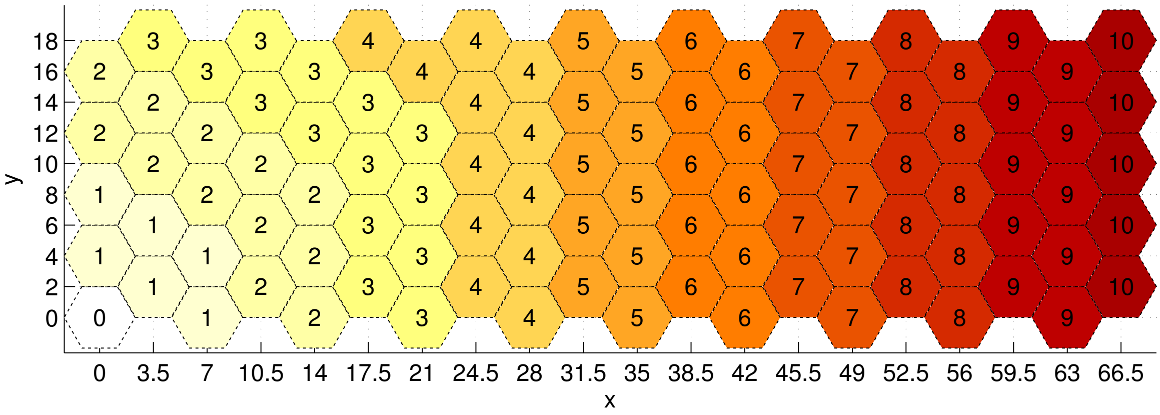


Figure 5.23: Hierarchy level of each beacon in the networks for beacon 1 (Figure 5.22) as reference beacon and considering a maximum range of 10 m.

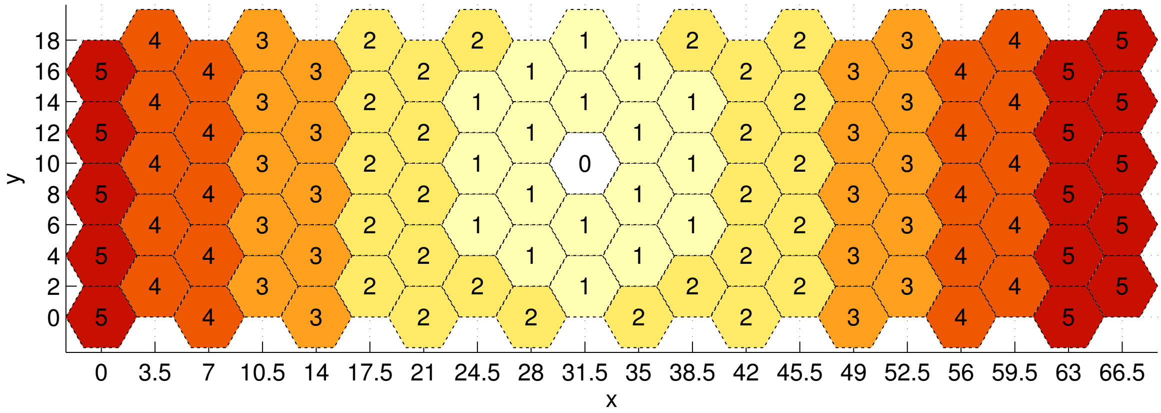


Figure 5.24: Hierarchy level of each beacon in the networks for beacon 48 (Figure 5.22) as reference beacon and considering a maximum range of 10 m.

As a result of the reduction of levels in the hierarchy, the total number of obtained distances will increase considerably. This effect can be observed in Figure 5.25 where the reference role was set to beacon 1 and Figure 5.26 where the reference role was set to beacon 48.

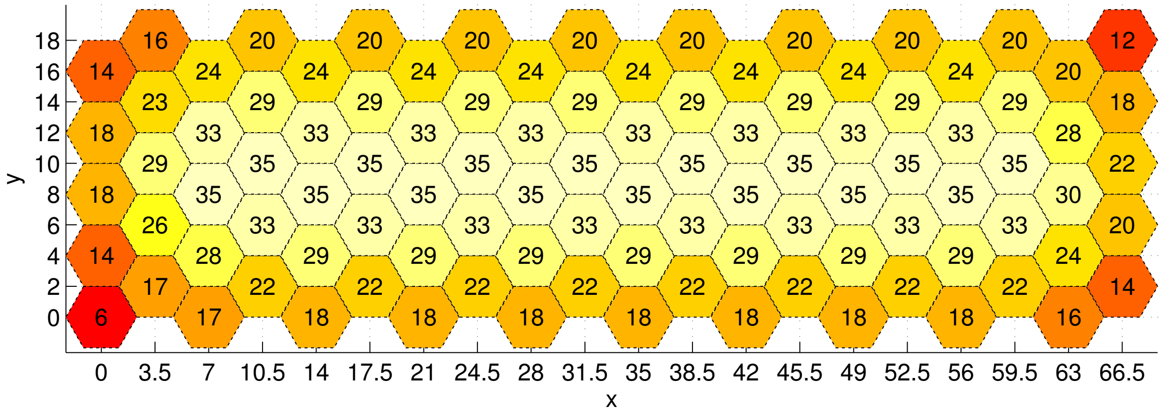


Figure 5.25: Total number of distance measurements obtained in each beacon. Considering beacon 1 (Figure 5.22) as the reference node and a maximum range of 10 m.

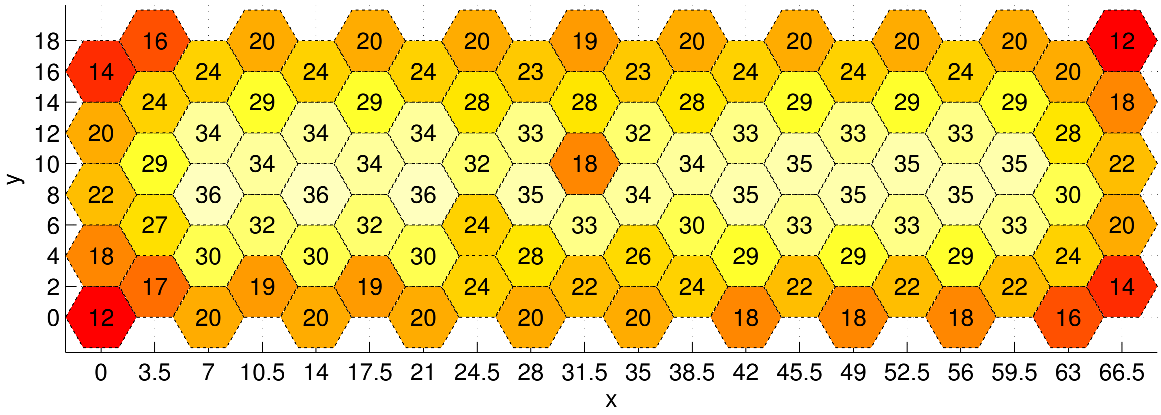


Figure 5.26: Total number of distance measurements obtained in each beacon. Considering beacon 48 (Figure 5.22) as the reference node and a maximum range of 10 m.

It can be seen in Figure 5.25 and 5.26 that each node obtains about three times more distances information, which may improve the location algorithm accuracy, as will be seen later in Section 6.6.2. Despite all this advantages, increasing the beacons range brings a very important disadvantaged that cannot be discarded. By increasing the beacon range the probability of message collision will increase considerably and consequently the mechanism of sharing the acoustic channel will be more complex which could result in a reduced refresh rate and energy efficiency.

5.5 Location Algorithms

As it was presented in section 5.1 the proposed US location system is based on the distance measure between nodes (beacons or mobiles). Therefore, the location algorithms must use these distances information to estimate the position of a beacon or mobile with unknown position. The majority of the distance based location algorithms belong to one of two categories [78]:

Geometric-based methods are the most widely used methods manly due to their computational efficiency providing the positioning estimation by the solution of one or several simple mathematical equations [78]. Nevertheless, they need to have all distance measurements with small error and therefore they do not perform well in the presence of non-line-of-sight (NLOS).

Minimization of a cost function methods are less frequently used in location due to their computational complexity. However, they behave very well in the presence of outliers[4] in distance measures which can arise from NLOS [78]. The position estimation is obtained by finding the minimum of a specif cost function.

The category that best suits the mobile location are the geometric-based algorithms due to the fact that they present a low computational complexity allowing to be easily implemented in the limited hardware resources. Moreover, for the mobile node due to the movement associated to it, it is more important to provide a fast position with a little bit less accuracy than a more accurate position but that would take several minutes to compute. On the other hand, for the beacon network, due to their static nature, the beacon's position will be exactly the same for a long time. Therefore, position refresh could take several minutes or even hours to be carried out in return to a much more accurate position estimation. Moreover, the distance between the beacons is obtained from the clock synchronization process and therefore it is almost impossible to obtain all the possible distance measures between them. Each beacon will only have access to a sub set of these distance measures. As a consequence of all of this, the minimization of a cost function will be the best choice to estimate the beacon's position.

In conclusion, to obtain a very accurate position estimation for the beacons and take advantage from all the possible distance measures obtained for each node, a minimization of a cost function must be used, in return this position estimation will present a very slow refresh rate. On the other hand, the mobile nodes must use a geometric-based algorithm that provides a much higher refresh rate allowing proper positioning.

[4] A distance outlier is distance measure that is numerically distant from the rest of the measurements.

5.5.1 Problem Definition

Before describing the different location algorithms it is important to define the problem and the various variables involved. For that the example of Figure 5.27 will be used.

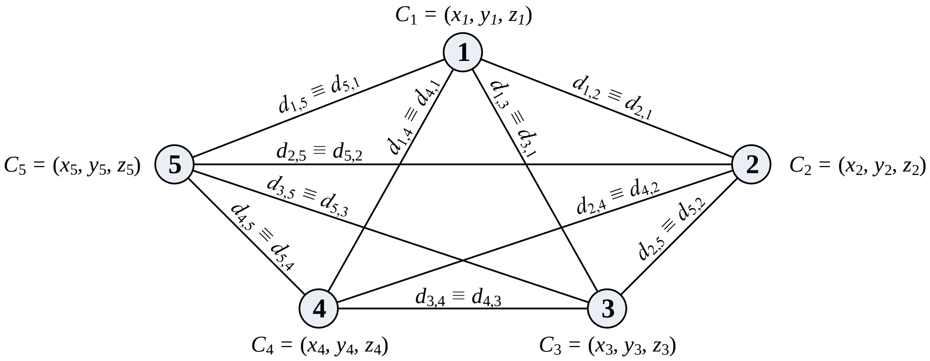


Figure 5.27: Example of a simple node displacement with several different variables.

The actual coordinates of node i are defined by C_i and its individual Cartesian coordinates are represented by $(x_i;\ y_i;\ z_i)$:

$$C_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}. \tag{5.14}$$

And the matrix of coordinates is defined by:

$$\mathbf{C} = \begin{bmatrix} C_1 & C_2 & C_3 & \cdots & C_N \end{bmatrix}, \tag{5.15}$$

where N is the total number of nodes.

The euclidean distance between the coordinates of node i and node j will be denominated by $d_{i,j}$ (which is the same of $d_{j,i}$). It can be computed from the coordinates as:

$$\begin{aligned} d_{i,j} &= d_{i,j}(\mathbf{C}) = d\left(C_i,\ C_j\right) = \left\|C_i - C_j\right\| \\ &= \sqrt{\left(C_i - C_j\right)^{\mathrm{T}}\left(C_i - C_j\right)} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} \end{aligned} \tag{5.16}$$

For example, the coordinates of node 1 are represented by C_1 and its individual Cartesian coordinates are $(x_1;\ y_1;\ z_1)$ moreover its distance to node 2 will be represented by $d_{1,2}$ which is the same of the distance of node 2 to node 1, $d_{2,1}$. Nevertheless, this is the euclidean distance between node's coordinates which is not the distance measurements obtained from the TOF. Therefore, the distance measurement between node i and j is going to be denominated by $\delta_{i,j}$. And for additive noise in the distance measurements, $\delta_{i,j}$ relates to $d_{i,j}$ by:

$$\delta_{i,j} = d_{i,j} + \varepsilon_{i,j}, \tag{5.17}$$

where, $\varepsilon_{i,j}$ is the distance measure error.

5.6 Mobile Location

In the proposed architecture the mobile node does not send any type of message and it must compute it location from the information sent from other beacons. For that goal each beacon must broadcast a *location* message. This message must contain the time stamp and beacon coordinates. Figure 5.28 presents an example where each beacon broadcasts its coordinates, C, and the message time stamp, ς. In order to reduce collisions between the broadcast messages several techniques can be employed, such as, TDMA and/or frequency division multiple access (FDMA). Therefore each beacon could randomly send a *location* message in a randomly chosen frequency band. This approach will increase the number of received *location* messages for each mobile node. In addition to this, after receiving each *location* message the mobile node must record the received time stamp.

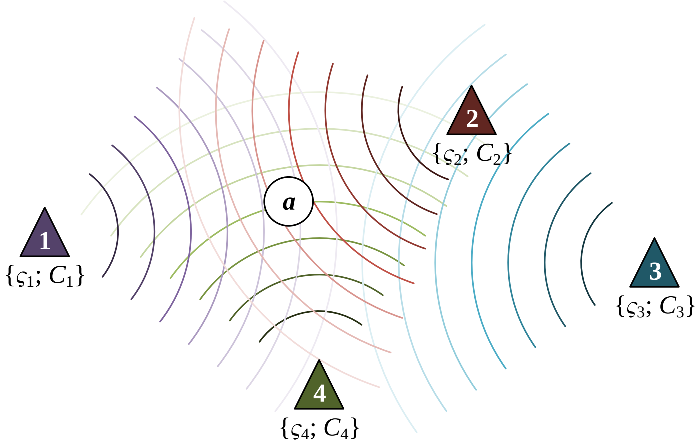


Figure 5.28: Example of the *location* message broadcast from 4 beacons.

The mobile node after collecting a set of *location* messages from different beacons can compute its own location. Figure 5.29 presents an example of four received messages. Each beacon i sends a *location* message at time ς_i, in the beacons network clock, this message is received in the mobile

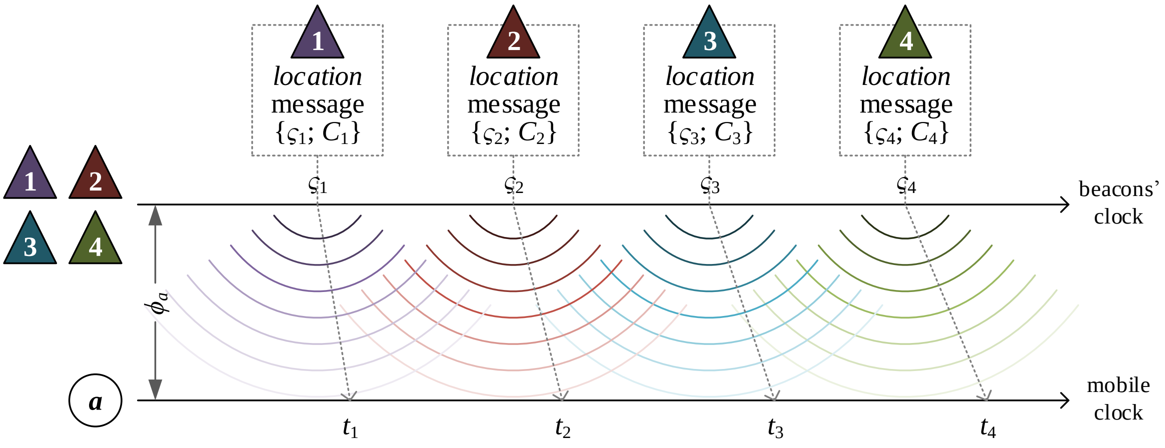


Figure 5.29: *Location* messages received by the mobile at different times from 4 beacons.

node at time t_i in local clock. This times are related by:

$$t_i = \varsigma_i + \tau_i - \phi_a \tag{5.18}$$

where, τ_i is the propagation delay and ϕ_a the clock offset between the mobile node a and the beacons network. After receiving all *location* messages the mobile node cannot determine its location by processing only the TOA instants $(t_1, ... , t_4)$ due to the unknown clock offset ϕ_a and therefore TOA based algorithms cannot be used [10]. Nevertheless, algorithms based in the TDOA can overcame this problem by estimating the mobile node position through the difference between the TOAs of each received message and the first received message:

$$t_{i,1} = t_i - t_1 = \varsigma_i - \varsigma_1 + \tau_i - \tau_1, \qquad i = 2, ..., L, \tag{5.19}$$

where, L is the total number of beacons. Note that the time difference is not affected by the clock offset ϕ_a and the ς_i is received in the *location* message. As a result of this, the distance difference between the beacons is given by:

$$\Delta_i = c \times (t_{i,1} - \varsigma_i + \varsigma_1) = c \times (\tau_i - \tau_1). \tag{5.20}$$

As a result of this the distance between the mobile node and the different beacons is given by:

$$d_i = d_1 + \Delta_i. \tag{5.21}$$

Without loss of generality, the origin of the Cartesian coordinate system is going to be set at beacon 1:

$$\widetilde{C}_i = \begin{bmatrix} \widetilde{x}_i \\ \widetilde{y}_i \\ \widetilde{z}_i \end{bmatrix} = C_i - C_1 \qquad \Longrightarrow \qquad \widetilde{C}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.22}$$

Under this simplification, and using (5.16) the relationship between the beacons coordinates, the mobile node a coordinates and the distances can be represented by the following equations:

$$d_1^2 = \tilde{x}_a^2 + \tilde{y}_a^2 + \tilde{z}_a^2 \tag{5.23}$$

$$(\Delta_i + d_1)^2 = (\tilde{x}_i - \tilde{x}_a)^2 + (\tilde{y}_i - \tilde{y}_a)^2 + (\tilde{z}_i - \tilde{z}_a)^2 \qquad i = 2, ..., L \tag{5.24}$$

the second equation (5.24) can be simplified to:

$$\Delta_i^2 + 2\Delta_i d_1 + d_1^2 = \tilde{x}_a^2 + \tilde{y}_a^2 + \tilde{z}_a^2 - 2(\tilde{x}_i\tilde{x}_a + \tilde{y}_i\tilde{y}_a + \tilde{z}_i\tilde{z}_a) + \tilde{x}_i^2 + \tilde{y}_i^2 + \tilde{z}_i^2. \tag{5.25}$$

As proposed in [79] this second order equation can be transformed into a linear equation by subtracting (5.23) from (5.25), resulting in:

$$\Delta_i^2 + 2\Delta_i d_1 = -2(\tilde{x}_i\tilde{x}_a + \tilde{y}_i\tilde{y}_a + \tilde{z}_i\tilde{z}_a) + \tilde{x}_i^2 + \tilde{y}_i^2 + \tilde{z}_i^2, \tag{5.26}$$

Rearranging the equation terms, the above equation can be written in matrix form as:

$$\begin{bmatrix} \widetilde{C}_2^{\mathrm{T}} \\ \widetilde{C}_3^{\mathrm{T}} \\ \vdots \\ \widetilde{C}_L^{\mathrm{T}} \end{bmatrix} \widetilde{C}_a = \frac{1}{2} \begin{bmatrix} ||\widetilde{C}_2||^2 - \Delta_2^2 \\ ||\widetilde{C}_3||^2 - \Delta_3^2 \\ \vdots \\ ||\widetilde{C}_L||^2 - \Delta_L^2 \end{bmatrix} - d_1 \begin{bmatrix} \Delta_2 \\ \Delta_3 \\ \vdots \\ \Delta_L \end{bmatrix} \tag{5.27}$$

Therefore, this equation can be rewritten as

$$\widetilde{\mathbf{C}}^{\mathrm{T}} \widetilde{C}_a = \mathbf{r} - d_1 \mathbf{\Delta} \tag{5.28}$$

where

$$\widetilde{\mathbf{C}} = \begin{bmatrix} \widetilde{C}_2 & \widetilde{C}_3 & \cdots & \widetilde{C}_L \end{bmatrix}; \qquad \mathbf{r} = \frac{1}{2} \begin{bmatrix} ||\widetilde{C}_2||^2 - \Delta_2^2 \\ ||\widetilde{C}_3||^2 - \Delta_3^2 \\ \vdots \\ ||\widetilde{C}_L||^2 - \Delta_L^2 \end{bmatrix}; \qquad \mathbf{\Delta} = \begin{bmatrix} \Delta_2 \\ \Delta_3 \\ \vdots \\ \Delta_L \end{bmatrix}$$

The least-squares solution for this equation in terms of the unknown d_1 is given by:

$$\widetilde{C}_a = \left(\widetilde{\mathbf{C}} \widetilde{\mathbf{C}}^{\mathrm{T}} \right)^{-1} \widetilde{\mathbf{C}} \left(\mathbf{r} - d_1 \mathbf{\Delta} \right) \tag{5.29}$$

Combining this equation with (5.23) it is possible to obtain $\widetilde{C}_a$ [10]. The real mobile position estimation can be obtained by changing the origin of the Cartesian coordinate system to its original position:

$$\hat{C}_a = \widetilde{C}_a + C_1. \tag{5.30}$$

5.7 Beacon Location

In the proposed architecture during the synchronization process each beacon can collect a set of distances. Using the same example used in Section 5.3 beacon 4 obtains 8 different distance measures as shown in Figure 5.30. To estimate the position of beacon 4, most of the location algorithms use only the distances from beacon 4 to other beacons, such as beacons 2, 3, 5, 7 and 8. These algorithms do not take advantages of the extra distances from beacon 2 to beacon 1, 3 and 5 [78, 80]. On the other hand, there are algorithms that take advantage of the distance between adjacent beacons, however these algorithms need to have all the distance measures between all the beacons involved [78, 80]. One possible example is the multidimensional scaling (MDS) algorithm [81]. The MDS is a method (widely used in the statistical field) used to visualize dissimilarities between objects in a predetermined N-dimensional space that best represents the given dissimilarities. For N less than 3, the resulting locations could be presented in a 3D graph [81]. While the nature of these dissimilarities is arbitrary, in a location system the dissimilarities can be obtained from the euclidean distances. And consequently, these dissimilarities must be represented in a 2D or 3D space depending on the type of location system. Since the MDS solution only depends on the distances between beacons, it is not unique and can be translated, rotated and flipped. In order to find a unique

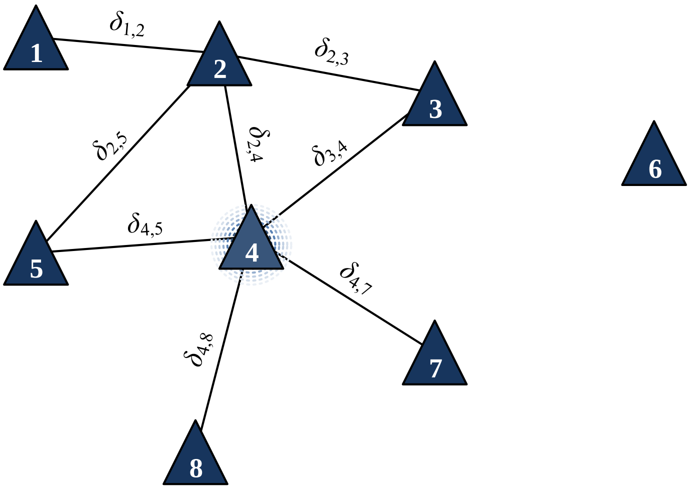


Figure 5.30: All the distance measures obtained by beacon 4 during the time synchronization process.

solution another step must be performed in order to match the obtained beacons coordinates with the coordinates of beacons with well known position by translating, rotating and flipping [81].

As mentioned before in classical MDS all inter-point dissimilarities must be known. However, this problem was overcame by several algorithms [78, 80]. One of the best algorithms from the accuracy point of view is the distributed weighted-multidimensional scaling (dw-MDS) [78]. Desai and Tureli shown in [80] that the dw-MDS presents the best accuracy even when a simplified version is used.

Distributed Weighted-Multidimensional Scaling

Distributed weighted-multidimensional scaling was originally proposed by Jose Costa, Neal Patwari and Alfred Hero III in [8]. The dw-MDS is a distributed and iterative algorithm based on the classic MDS that provides additional weight factors to each distance measurement according to its expected accuracy. Furthermore, it is based on a majorization method which guarantees that a cost function, S, is improved for each iteration.

Consider a beacon network composed by B beacons where A of them are anchor beacons (with a well known position). They are arranged in a such way that the last A beacons are the anchor beacons and the first $B-A$ are beacons with no knowledge of their position or have a imperfect initial position $\widehat{C_i}$ with an accuracy confidence a_i. Note that, for beacons with no knowledge of their position $a_i = 0$. If we have up to K distance measurements for each pair of beacons, the cost function, S, will be defined by:

$$S = 2\sum_{i=1}^{B-A}\sum_{j=i+1}^{B}\sum_{k=1}^{K} w_{i,j}^{(k)}\left(\delta_{i,j}^{(k)} - d_{i,j}\right)^2 + \sum_{i=1}^{B-A} a_i \left\|C_i - \widehat{C_i}\right\|, \tag{5.31}$$

where, the euclidean distance $d_{i,j}$ is given by (5.16) and the weight $w_{i,j}^{(k)}$ can be used to quantify the predicted accuracy of the distance measurement $\delta_{i,j}^{(k)}$. If a particular distance measurement between beacon i and j is not available then $w_{i,j}^{(k)} = 0$.

Adapting to Beacons Location

In the proposed location system, there is no a priory knowledge of their location and therefore a_i will be always zero. Moreover, due to the beacons objectives the distance measures between the beacons must be reduced to the minimum between two successive beacon location processes, in the limit the algorithm must perform location with only one distance measurement, $K = 1$. Under this constraints the cost function can be simplified to:

$$S = \sum_{i=1}^{B-A} \sum_{j=i+1}^{B} w_{i,j} \left(\delta_{i,j} - d_{i,j}\right)^2 . \tag{5.32}$$

Note that the scale factor of two was removed because it is not important in the minimization process.

In the original algorithm proposal this cost function was converted into a sum of local cost functions that each beacon with unknown location must run and share with neighbors (the details can be found in [8]). However, by this approach each beacon can only use the distance to its neighbors and it discards the distance between them. Moreover, beacons must exchange messages in each algorithm iteration which it is not recommended in our location system. Therefore, in order to take advantage of the distance measurements between neighbors each node must try to reduce their own cost function with the information that it could obtain during the synchronization process. Only, in the next synchronization process, beacons can share the result of their algorithms. As a result of this, the cost function in each beacon can be written as follow:

$$S = \sum_{i \in \mathcal{L}} S_i + S_k . \tag{5.33}$$

The cost function is written as a sum of individual cost functions associated to each beacon in $\mathcal{L}$ set plus a constant S_k independent of the $\mathcal{L}$ beacons coordinates. The set $\mathcal{L}$ is composed by beacons for which there are enough distance measurements to compute their location. Recalling the example of Figure 5.30 beacon 4 has enough information to compute also the coordinates of beacon 2 and therefore the beacon 4 local set will be $\mathcal{L} = \{2,\ 4\}$. Although, this approach is more computational intensive, it will increase the accuracy of the estimation of beacon's 4 position and it will reduce considerably the number of iterations needed as can be seen later in Section 5.8.2. Note that this set does not include anchor beacons manly due to the fact that their coordinates are considered to be exact. Under these assumptions, by minimizing each individual cost function, S_i, the global cost function S is also minimized. Moreover, each individual cost function can be written as:

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^{B} \hat{w}_{i,j} \left(\delta_{i,j} - d_{i,j}\right)^2 , \tag{5.34}$$

where

$$\hat{w}_{i,j} = \begin{cases} \dfrac{w_{i,j}}{2} & \text{if } i \in \mathcal{L} \wedge j \in \mathcal{L}; \\ w_{i,j} & \text{otherwise}; \end{cases} \tag{5.35}$$

where the weight $w_{i,j}$ will represent the availability (or accuracy) of the distance measurement $\delta_{i,j}$

received by the beacon. Note that the weights $w_{i,j}$ are specific for each beacon.

Using the same approach that was used in the original dw-MDS method the cost function $S_i = S_i(C_i)$ can be minimized iteratively using the quadratic majorizing functions presented in [82]. The idea behind the use of a majorizing function is to minimize a much simple auxiliary function with a larger dimension that touches the original function in the supporting point. A majorizing function $M_i(C, D)$ for $S_i(C)$ is a function:

$$M_i : \mathbb{R}^2 \times \mathbb{R}^2 \longrightarrow \mathbb{R},$$

for beacons coordinates in 2D or

$$M_i : \mathbb{R}^3 \times \mathbb{R}^3 \longrightarrow \mathbb{R},$$

for beacons coordinates in 3D. This majorizing function must satisfy the follow constraints:

$$M_i(C, D) \geq S_i(C) \qquad\qquad S_i(C) = M_i(C, C)$$

This function can be used to implement a iterative minimization process [82]. By starting at initial condition $C^{(0)}$, the function $M_i\left(C, C^{(0)}\right)$ is them minimized as a function of C. The resultant minimum $C^{(1)}$ can then be used to create a new majorizing function $M_i\left(C, C^{(1)}\right)$ that must be minimized to find $C^{(2)}$ and so on. This process ends when convergence is achieved [82]. As suggested in [8] a good majorizing function that can be minimized analytically is a quadratic function. For that reason, we must start by rewriting the individual cost function as:

$$S_i(C_i) = f_i^2(C_i) - 2g_i'(C_i) + h_i, \tag{5.36}$$

where

$$f_i^2(C_i) = \sum_{\substack{j=1\\j\neq i}}^{B} \hat{w}_{i,j} d_{i,j}^2 = \sum_{\substack{j=1\\j\neq i}}^{B} \hat{w}_{i,j} \left(C_i - C_j\right)^{\mathrm{T}} \left(C_i - C_j\right), \tag{5.37}$$

$$g_i'(C_i) = \sum_{\substack{j=1\\j\neq i}}^{B} \hat{w}_{i,j} \delta_{i,j} d_{i,j} = \sum_{\substack{j=1\\j\neq i}}^{B} \hat{w}_{i,j} \delta_{i,j} \sqrt{\left(C_i - C_j\right)^{\mathrm{T}} \left(C_i - C_j\right)}, \tag{5.38}$$

$$h_i = \sum_{\substack{j=1\\j\neq i}}^{B} \hat{w}_{i,j} \delta_{i,j}^2 \tag{5.39}$$

The first equation (5.37) is quadratic function of C_i,(5.38) depends on C_i through a sum of square roots and(5.39) does not depend on C_i. Defining the majorizing function as:

$$M_i(C_i, D_i) = f_i^2(C_i) - 2g_i(C_i,\ D_i) + h_i, \tag{5.40}$$

where

$$g_i(C_i, D_i) = \sum_{\substack{j=1\\j\neq i}}^{B} \hat{w}_{i,j} \delta_{i,j} \frac{\left(C_i - C_j\right)^{\mathrm{T}} \left(D_i - D_j\right)}{\sqrt{\left(D_i - D_j\right)^{\mathrm{T}} \left(D_i - D_j\right)}}. \tag{5.41}$$

From the Cauchy inequality [83]:

$$\left|\left\langle C_i - C_j, D_i - D_j \right\rangle\right| \leq \left\|C_i - C_j\right\| . \left\|D_i - D_j\right\|, \tag{5.42}$$

where $\langle \cdot, \cdot \rangle$ represents the inner product. Equation 5.42 can be rewritten as follow:

$$\frac{\left(C_i - C_j\right)^{\mathrm{T}} \left(D_i - D_j\right)}{\sqrt{\left(D_i - D_j\right)^{\mathrm{T}} \left(D_i - D_j\right)}} \leq \sqrt{\left(C_i - C_j\right)^{\mathrm{T}} \left(C_i - C_j\right)} \tag{5.43}$$

From this inequality it can be easily seen that:

$$g_i(C_i, D_i) \leq g'_i(C_i) \tag{5.44}$$

and therefore

$$S_i(C_i) \leq M_i(C_i, D_i). \tag{5.45}$$

To minimize S_i it is proposed to minimize M_i first:

$$\frac{\partial M_i(C_i, D_i)}{\partial C_i} = 0. \tag{5.46}$$

The expression for this gradient is given by:

$$\frac{\partial M_i\left(C_i, D_i\right)}{\partial C_i} = 2 \sum_{\substack{j=1 \\ j \neq i}}^{B} \hat{w}_{i,j} \left[\left(C_i - C_j\right) + \frac{\delta_{i,j}\left(D_j - D_i\right)}{\sqrt{\left(D_i - D_j\right)^{\mathrm{T}} \left(D_i - D_j\right)}} \right] \tag{5.47}$$

If $\mathbf{C}^{(k)}$ is a matrix with all the coordinates estimates at iteration k, the coordinates update for beacon i can be derived from (5.47):

$$C_i^{(k+1)} = \frac{\mathbf{C}^{(k)} \mathbf{u}_i^{(k)}}{n_i}, \tag{5.48}$$

where n_i is given by:

$$n_i = \sum_{\substack{j=1 \\ j \neq i}}^{B} \hat{w}_{i,j}. \tag{5.49}$$

and $\mathbf{u}_i^{(k)}$ is a column vector whose elements p are given by:

$$\mathbf{u}_i^{(k)} = \begin{cases} \hat{w}_{i,p} \left(1 - \dfrac{\delta_{i,p}}{\sqrt{\left(C_i^{(k)} - C_p^{(k)}\right)^{\mathrm{T}} \left(C_i^{(k)} - C_p^{(k)}\right)}} \right) & \text{if } p \neq i; \\ \displaystyle\sum_{\substack{j=1 \\ j \neq i}}^{B} \frac{\hat{w}_{i,j} \delta_{i,j}}{\sqrt{\left(C_i^{(k)} - C_j^{(k)}\right)^{\mathrm{T}} \left(C_i^{(k)} - C_j^{(k)}\right)}} & \text{if } p = i; \end{cases} \tag{5.50}$$

Note that $C_j^{(k)}$ is only different from $C_j^{(k-1)}$ if $j \in \mathcal{L}$ or if during the synchronization process a new C_j coordinate is received.

The location algorithm running in each beacon is summarize in Figure 5.31. Regarding the algorithm, a stopping criterion ϵ was introduce to stop the algorithm if after one iteration the cost function is not reduced or if it is only reduced by a little amount, reducing the computational impact of the algorithm. After that the algorithm stops and it is only restarted if new information arrives during a synchronization process.

```
loop
    $k = 0$
    compute $S^{(0)}$ from Equation 5.32
    for $i \in \mathcal{L}$ do
        compute $n_i$ from Equation 5.49
    end for
    repeat
        $k = k + 1$
        for $i \in \mathcal{L}$ do
            compute $\boldsymbol{u}_i^{(k-1)}$ from Equation 5.50
            compute $C_i^{(k)}$ from Equation 5.48
        end for
        compute $S^{(k)}$ from Equation 5.32
    until $S^{(k-1)} - S^{(k)} < \epsilon$
    wait for next synchronization process
end loop
```

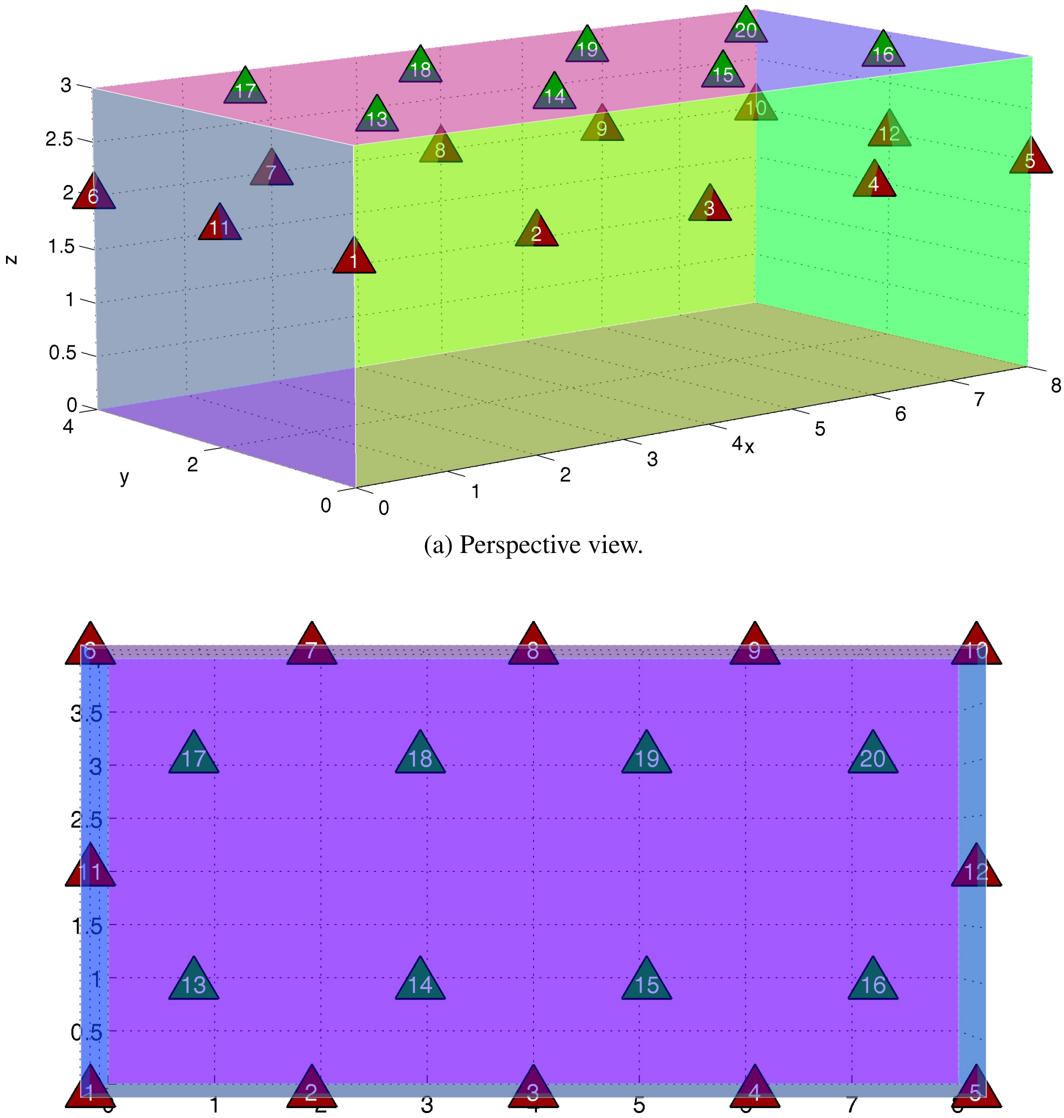
Figure 5.31: Beacons location algorithm running in each beacon.

5.8 Location Algorithms Results

This section presents the performance of both algorithms. The analysis will start with the mobile location problem. An example will be used to evaluate the impact of the distance errors and the beacons position errors on the mobile location algorithm results. In the second part of this section the beacons location algorithm will be analyzed. A complex beacon structure, where all the beacons lies in the same plain will be used. This example will be used to verify the impact of the number of anchor beacons, the distance error and the beacon failure.

5.8.1 Mobile Location

To present the mobile location performance and possible problems that could emerge, the beacon network of Figure 5.32 will be used, it consists of a room 8 × 4 × 3 m with 20 beacons, 12 in the walls at 2 m from the floor and 8 in the ceiling. During the experiments it will be considered that the distance and the coordinates are known and beacon 1 the reference beacon. Moreover, a receiver at 1.5 m from the floor ($z = 1.5$ m) and at different x and y positions will be considered.



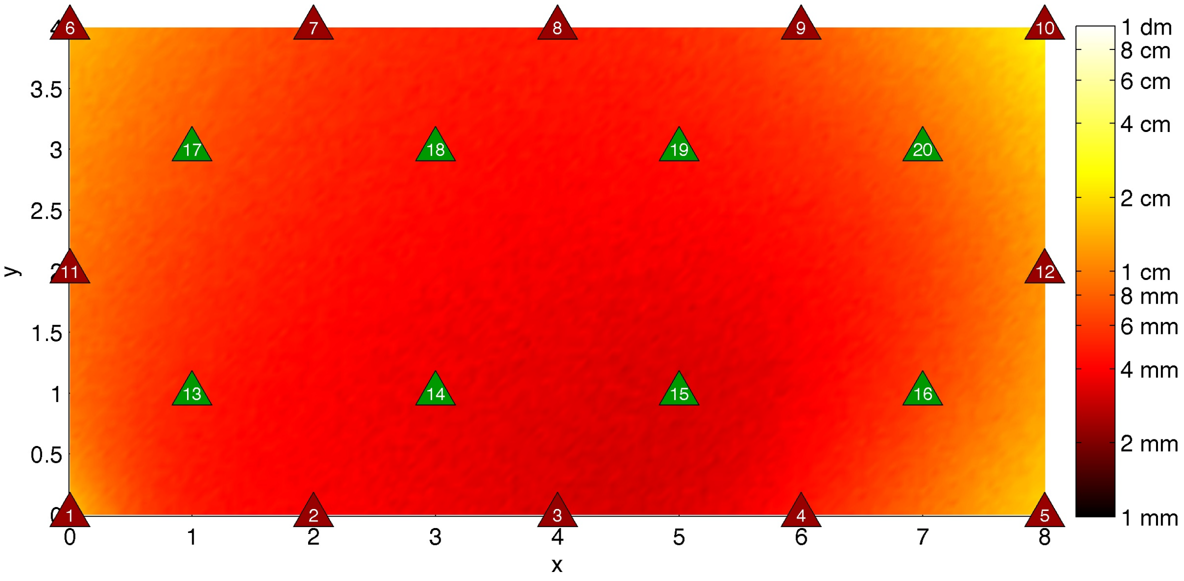
(a) Perspective view.

(b) Top view.

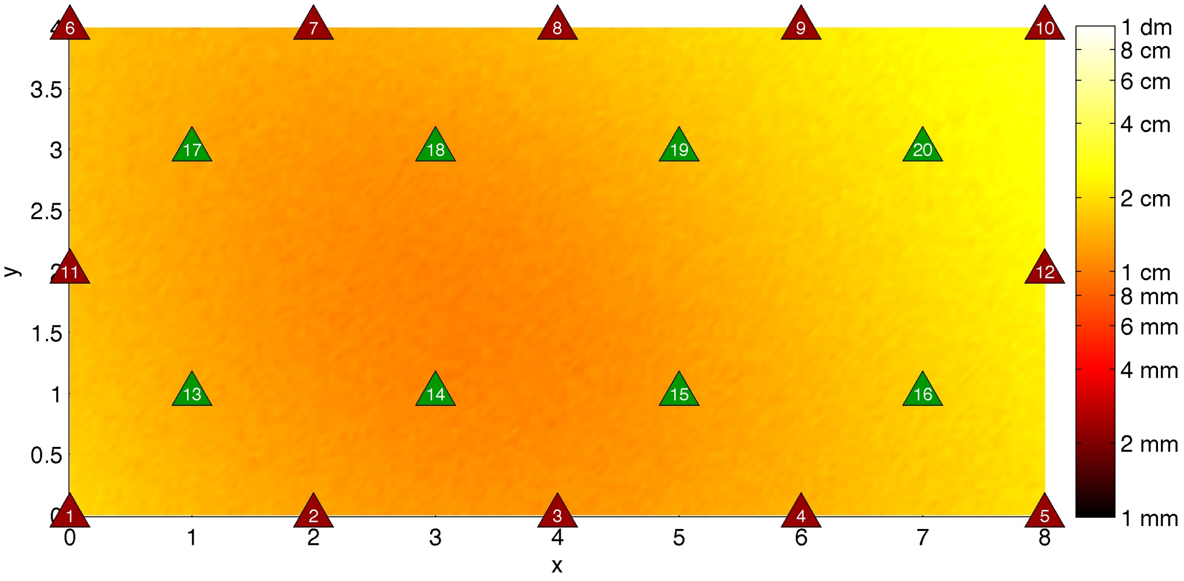
Figure 5.32: Simulation environment to test the mobile location algorithm.

Distance Error

In this experiment it will be considered that the distance measurements present additive white Gaussian noise with a standard deviation of 1 cm. The algorithm, in the receiver, uses a grid of 5 cm in the *z-plane* at 1.5 m from the floor. For each receiver position 500 trial were carried out. The resultant error is presented in Figure 5.33. The error in the *x-coordinate* is almost constant over all positions and around a few millimeters. Nevertheless, the error increases a little bit in the room corners where the error increases to a few centimeters, especially, in the corner near beacon 10. Where the error is bigger due to the fact that the TOA will be compared to the TOA of the pulse from beacon 1. The error in *y-coordinate* is not presented, because, it follows the same error pattern of the *x-coordinate*. On the other hand, the error in the *z-coordinate* presents a bigger error than in the other coordinates and it is around a few centimeters.



(a) Error in the x coordinate.

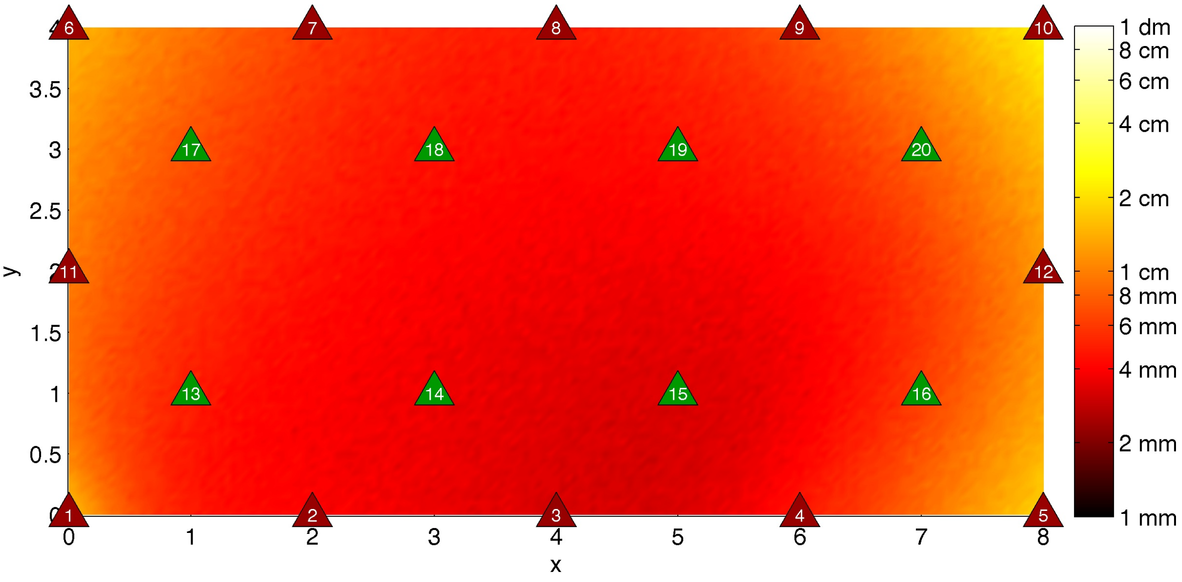


(b) Error in the z coordinate.

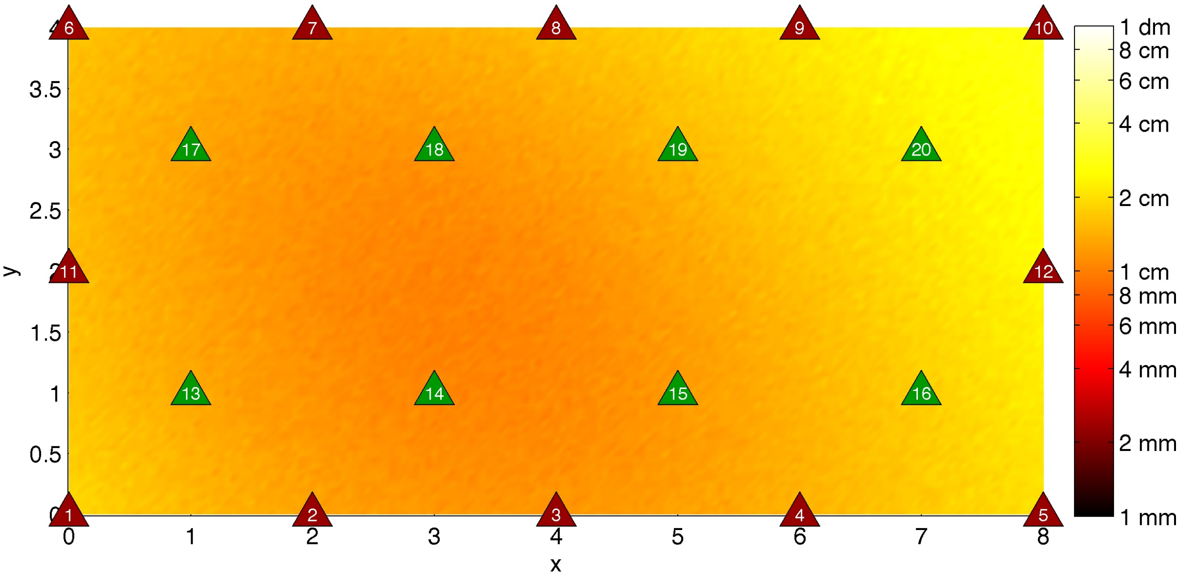
Figure 5.33: Average error for x and z coordinate, considering a distance measurement error with 1 cm of standard deviation.

Beacon Position Error

The experiment described here is similar to the previous one. However, in this example it is considered that the beacons coordinates present additive white Gaussian noise with a standard deviation of 1 cm and the distance is known accurately. The resultant error is presented in Figure 5.34. As one can see the error in the *x-coordinate* and *z-coordinate* follows precisely the same error pattern of the previous example.



(a) Error in the x coordinate.

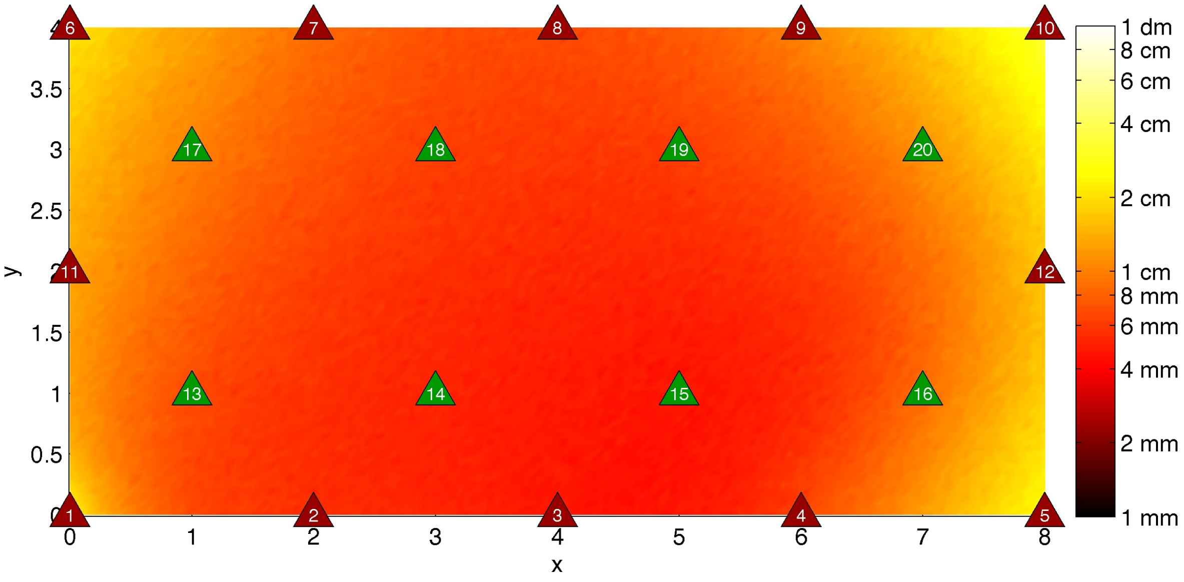


(b) Error in the z coordinate.

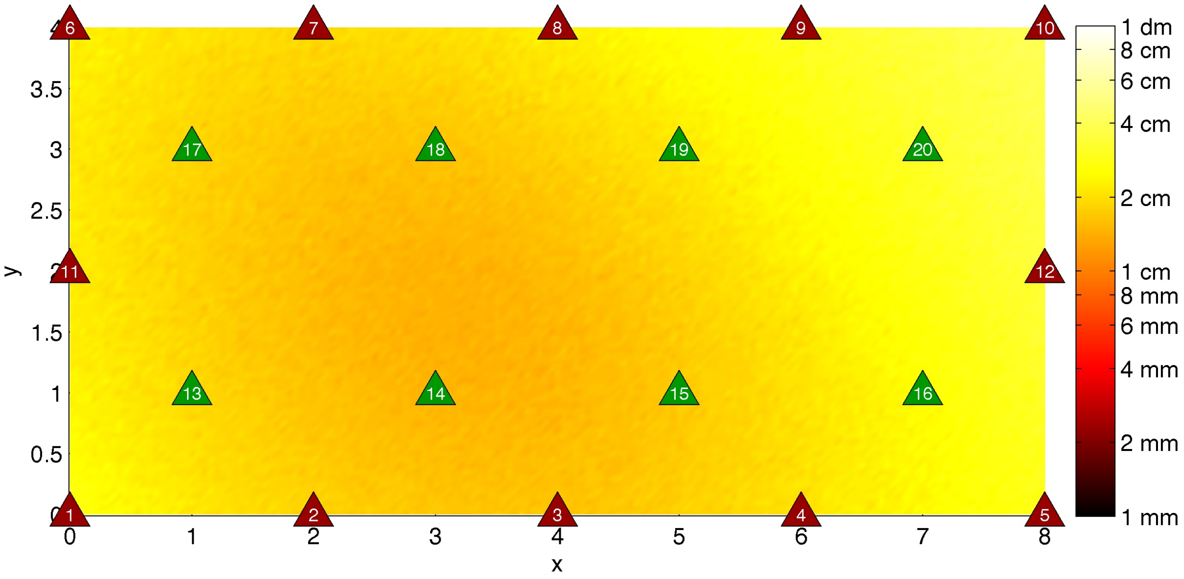
Figure 5.34: Average error for x and z coordinate, considering a beacon position error with 1 cm of standard deviation.

Beacon Position and Distance Error

This experiment is a very similar to both previous examples. However, in this example it is considered that the beacons coordinates and the distance measurements present additive white Gaussian noise with a standard deviation of 1 cm. The resultant error is presented in Figure 5.35. As expected the error in the *x-coordinate* and *z-coordinate* increases a little bit in comparison to both previous examples.



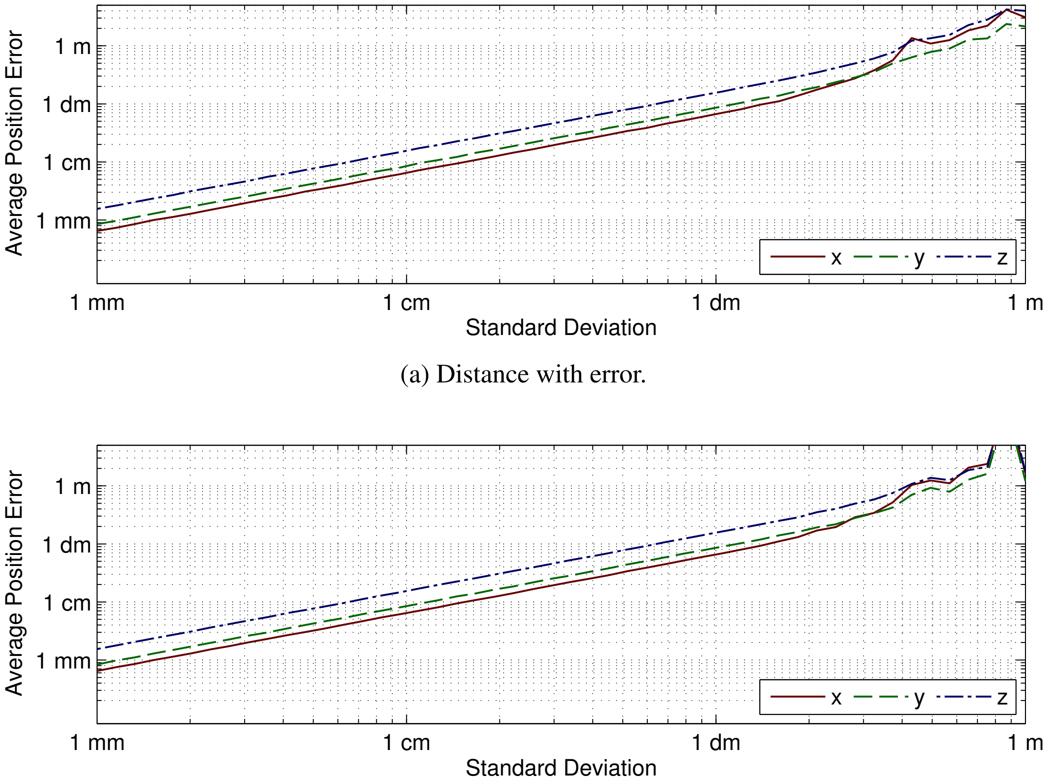
(a) Error in the x coordinate.



(b) Error in the z coordinate.

Figure 5.35: Average error for x and z coordinate, considering a beacon position and distance error with 1 cm of standard deviation.

Figure 5.36 presents the average position error considering a distance error (Figure 5.36a) or a beacon position error (Figure 5.36b) as a function of the noise standard deviation. As one can see, the resultant mobiles position errors are similar in both cases and they have the same magnitude order of the distance or beacons position error. Therefore, to keep the estimation error of mobiles position as low as possible, both the distance and the beacons position errors must be as low as possible.



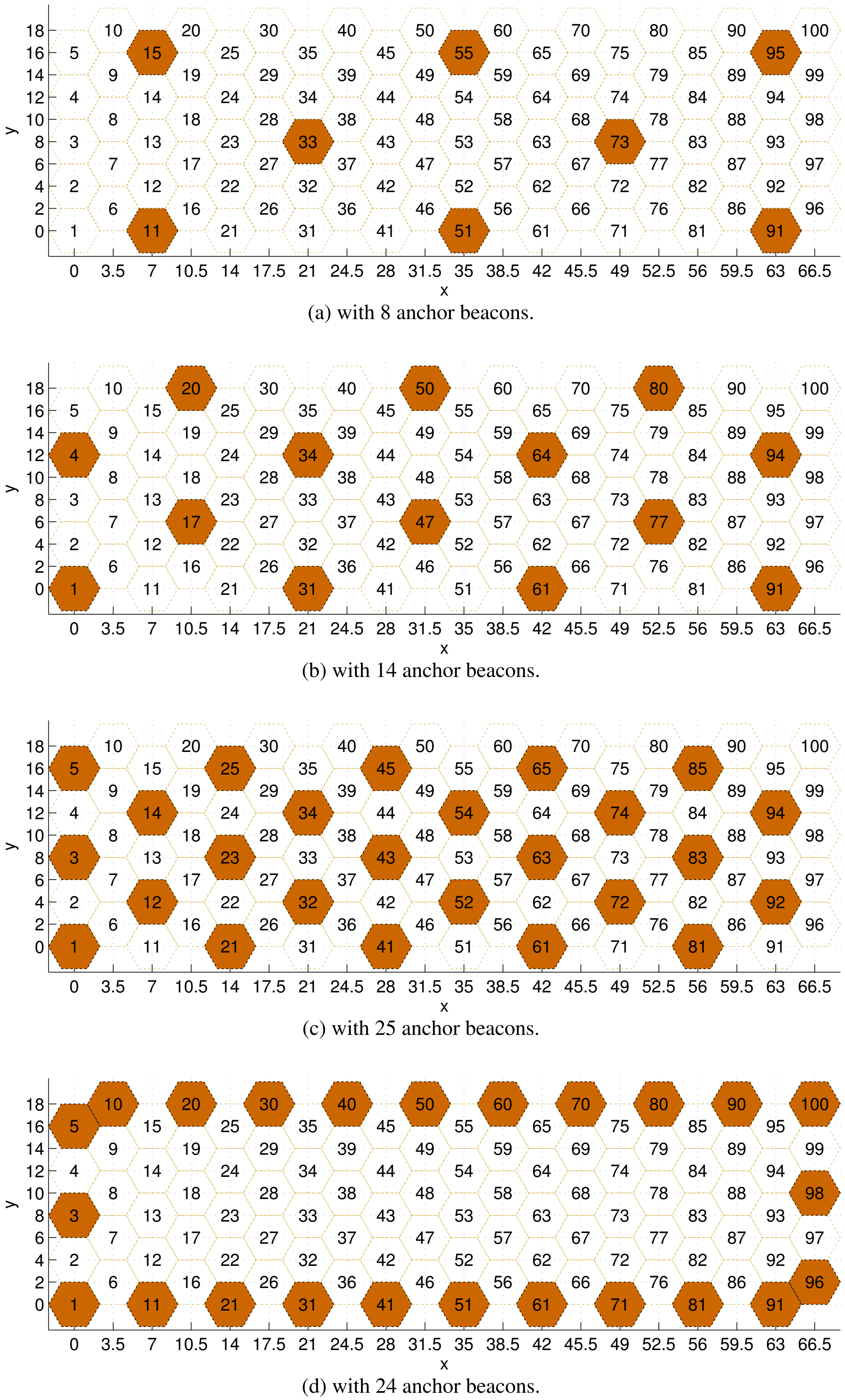
(a) Distance with error.

(b) Beacon position with error.

Figure 5.36: Average error for each coordinate, considering beacon position (a) or distance error (b) as a function of noise standard deviation.

5.8.2 Beacon Location

The wide spread beacons network, presented in Section 5.4 is used here to exemplify the beacon location performance and possible problems that might arise. Moreover, it is considered that the maximum range of each beacon is about 10 m and four arrangements with different numbers of anchor beacons are used, Figure 5.37. In the eight anchor beacons example, Figure 5.37a, the anchors were distributed to ensure that the majority of the beacons could exchange messages with at least one anchor beacon. In the fourteen anchor beacons example, Figure 5.37b, the anchors were distributed to ensure that the majority of the beacons could exchange messages with at least two anchor beacons. On the other hand, in the twenty five anchor beacons example, Figure 5.37c, the anchors were distributed to ensure that the majority of the beacons could exchange messages with at least four anchor beacons. At last, in the twenty four anchor beacons example, Figure 5.37d, the anchors were distributed around the edge of the network.



(a) with 8 anchor beacons.

(b) with 14 anchor beacons.

(c) with 25 anchor beacons.

(d) with 24 anchor beacons.

Figure 5.37: Four different anchor beacons arrangement with different number of anchor beacons.

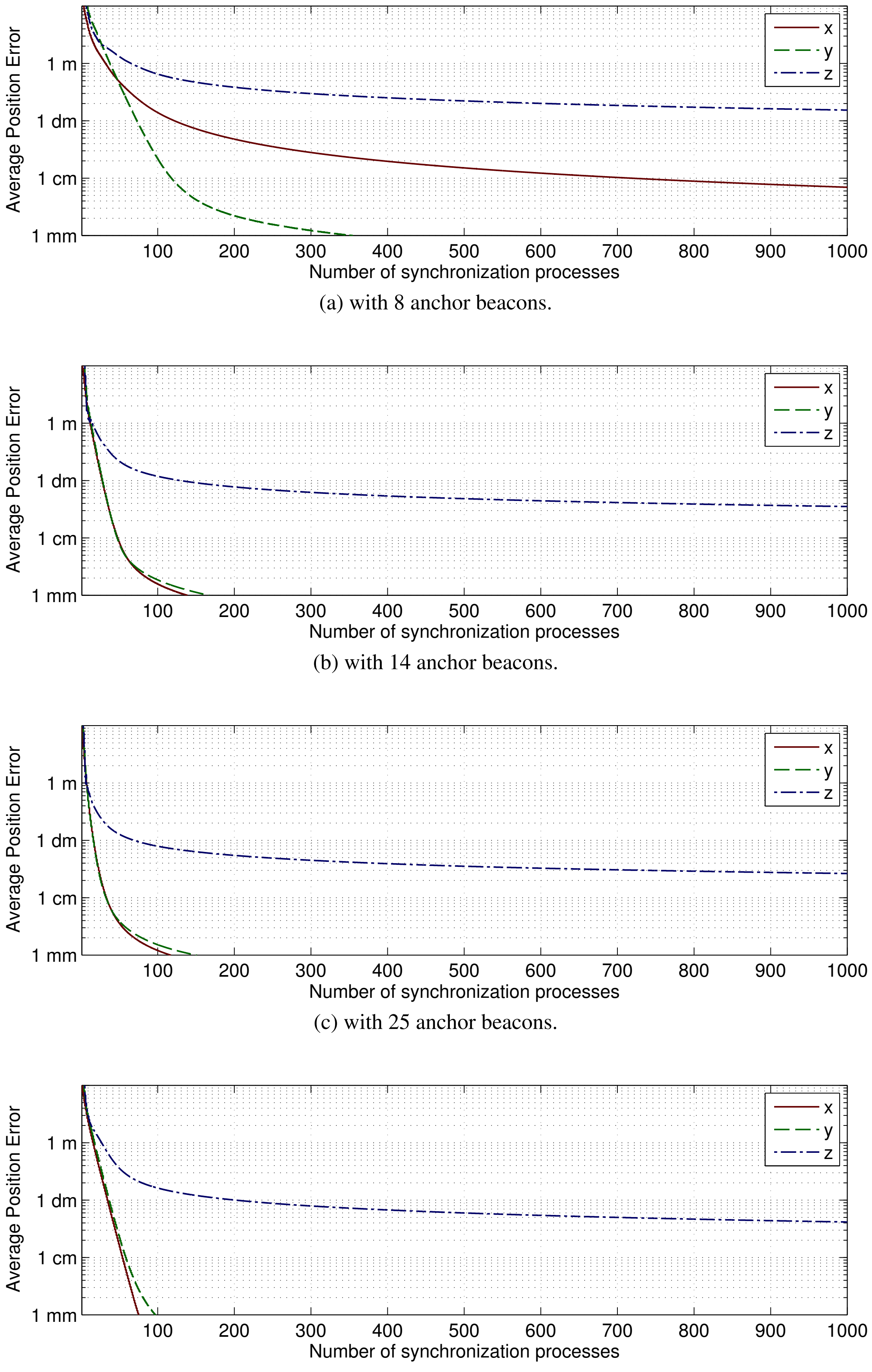
Beacon Distribution

To evaluate the impact of the beacon distribution the proposed algorithm was implemented in each beacon with the following characteristics:

- The initial beacon's coordinates was uniformly set between 0 and 100;
- In each beacon the local set $\mathcal{L}$ was chosen considering only beacons that have more than two distance measurements;
- The algorithm was iterated one hundred times between synchronization processes;
- One thousand synchronization processes were carried out.

The average position error of the algorithm's estimate for each coordinate in the four anchor distributions is depicted in Figure 5.38. For the eight anchors beacon example, Figure 5.38a, the position error reduces very slowly. On the other hand, for the remaining anchor distributions, Figure 5.38b, 5.38c and 5.38d, the position error is reduced to the order of a millimeter after about one hundred synchronization processes. Nevertheless, the error in the *z-coordinate* estimate remains very high even after a thousand synchronization processes. This is due to the fact that beacons are distributed in a plane (*z-plane*) which increases considerable the complexity to determine correctly the *z-coordinate*.

Figure 5.39d presents the *x-coordinate* error for the four anchor beacons distribution. As one can see the error in the *x-coordinate* is greater in the network corners, especially, where the beacons are far apart from the anchor beacons. Nevertheless, the error for the 14, 25 and 24 anchor distribution examples is around a few millimeters. On the other hand, the error in the *z-coordinate* has a more complex error distribution as can be seen in Figure 5.40. In the eight anchor beacons example, Figure 5.40a, the error is considerably high, in the order of a few centimeters, this error is especially high in the network extremities where the error increases to some meters. Nevertheless, the error in the *z-coordinate* may be reduced if the number of anchor nodes is increased, see Figure 5.40b and 5.40c. The twenty four beacons case, Figure 5.40d, increases a little bit the *z-coordinate* error, however the maximum error is reduced considerably.



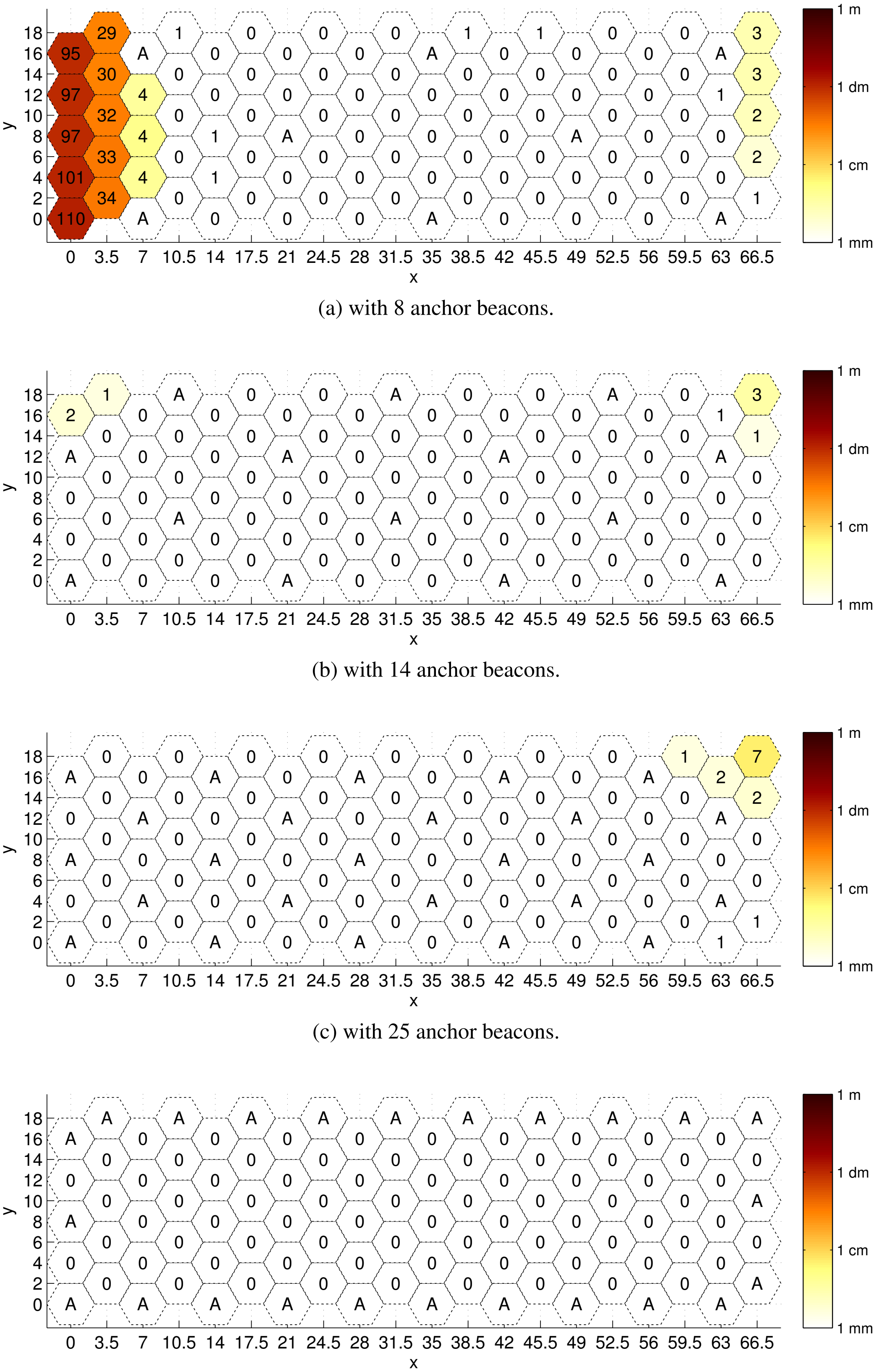
(a) with 8 anchor beacons.

(b) with 14 anchor beacons.

(c) with 25 anchor beacons.

(d) with 24 anchor beacons.

Figure 5.38: Average position error for each coordinate in the four anchor distributions after one thousand synchronization processes.



(a) with 8 anchor beacons.

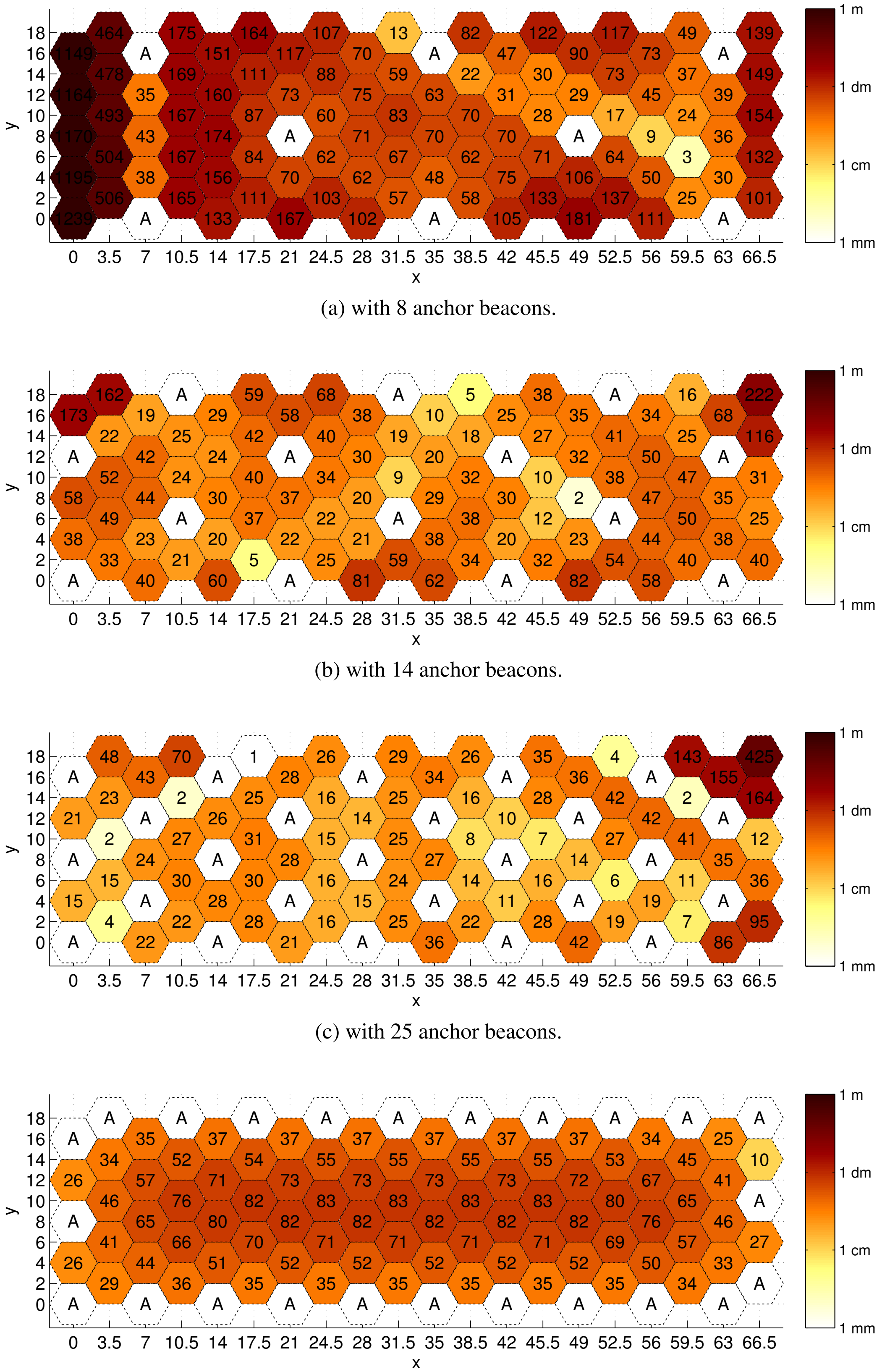
(b) with 14 anchor beacons.

(c) with 25 anchor beacons.

(d) with 24 anchor beacons.

Figure 5.39: The error of beacon's *x-coordinate*, in millimeter, for the four anchor beacons distribution after one thousand synchronization processes.

)



(a) with 8 anchor beacons.

(b) with 14 anchor beacons.

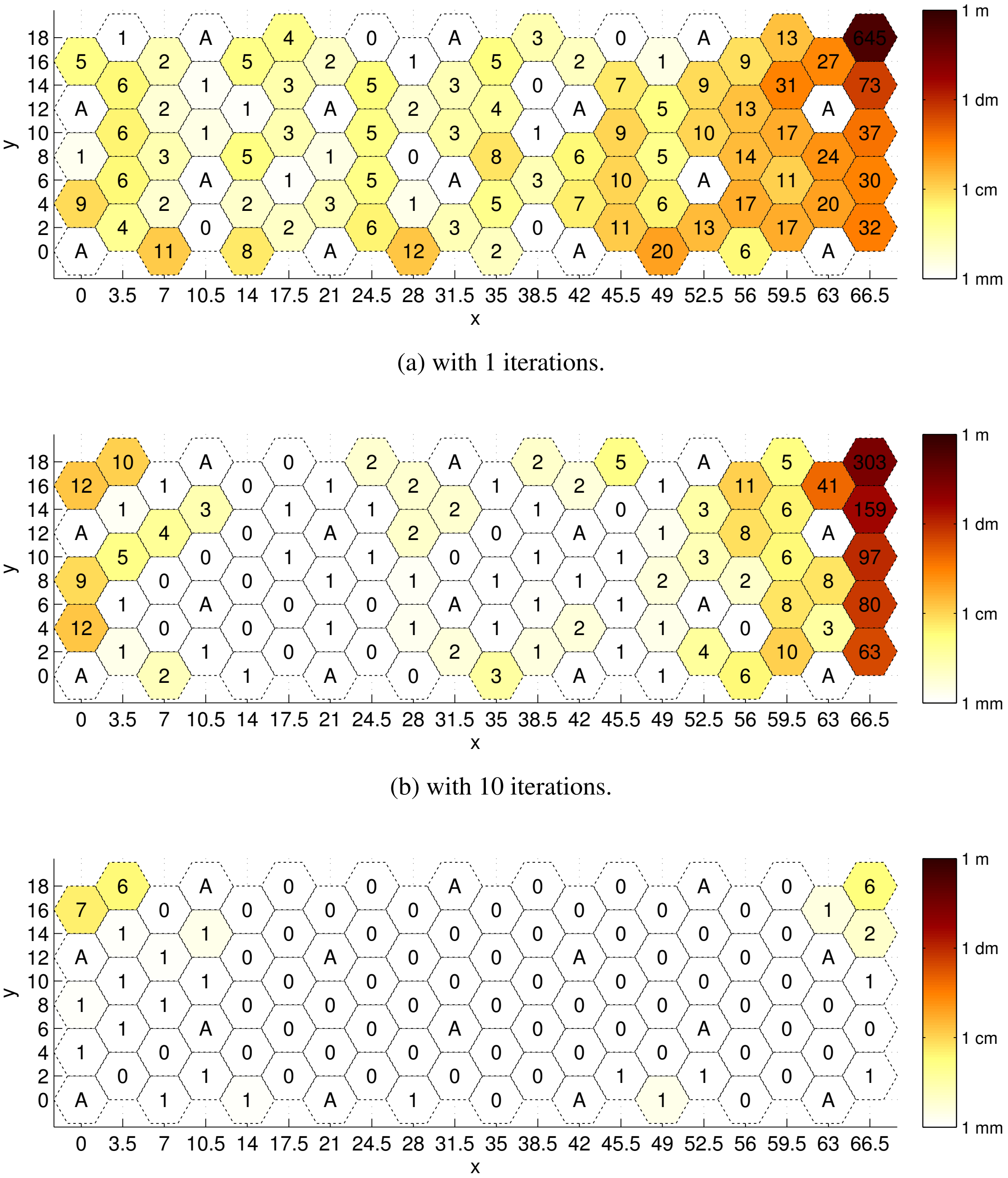
(c) with 25 anchor beacons.

(d) with 24 anchor beacons.

Figure 5.40: The error of beacon's *z-coordinate*, in millimeter, for the four anchor beacons distribution after one thousand synchronization processes.

Number of Iterations

To evaluate the impact of the number of iterations in the results of the proposed algorithm the same example of the previous section with fourteen anchor beacons will be used. Three values for the number of iterations between synchronization processes were used: one, ten and one hundred iterations. The error of the *x-coordinate* after 250 synchronization processes is depicted in Figure 5.41. For the same number of synchronization processes the increase on the number of algorithm iterations reduces considerable the error value from some centimeters, see Figure 5.41a, to less than seven millimeters, see Figure 5.41c.



(a) with 1 iterations.

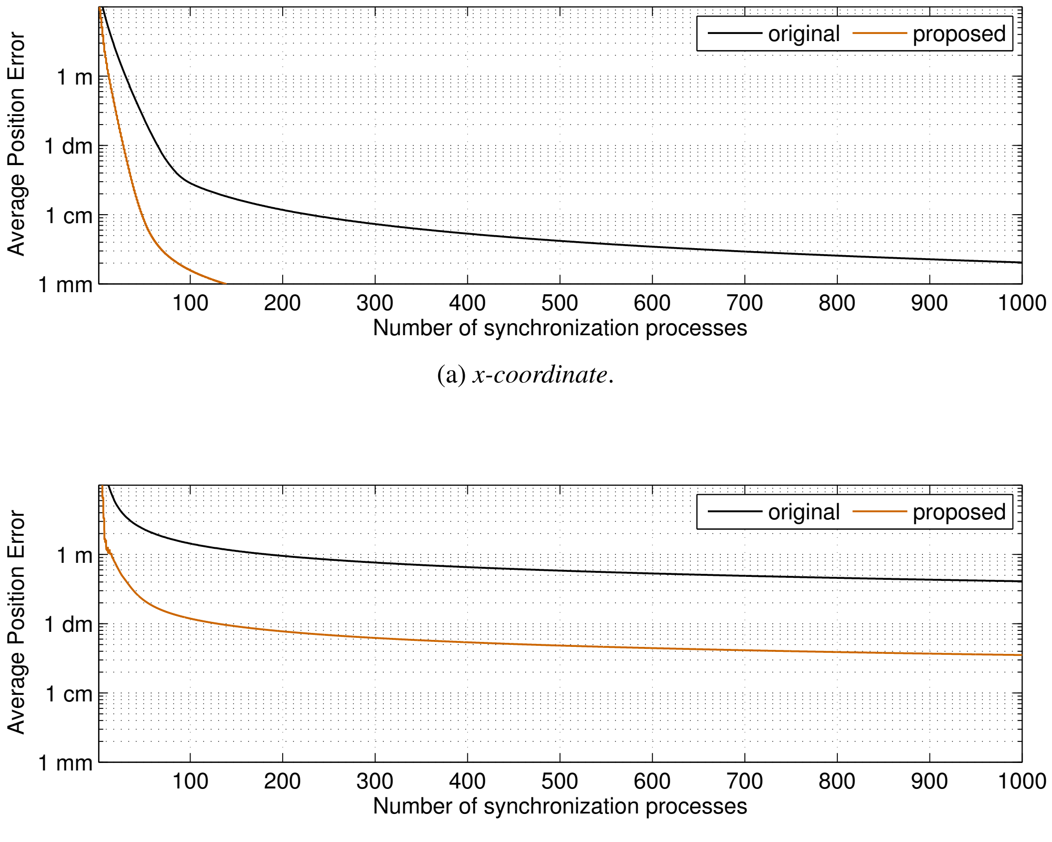
(b) with 10 iterations.

(c) with 100 iterations.

Figure 5.41: Error of the *x-coordinate* after 250 synchronization processes for three different numbers of iterations between synchronization processes.

Comparison to Original Algorithm

To compare the proposed modified dw-MDS algorithm with the original dw-MDS algorithm the same previous example with fourteen anchor beacons was used. The comparison was performed under the same conditions and the results are presented in Figure 5.42. As it can be seen, the results show that the proposed modified dw-MDS location system presents a better average position error that the original one for the same number of synchronization processes. Additionally, the results have shown that the number of synchronization processes, in the original algorithm, can be reduced by performing more algorithm iteration per synchronization process. In the original algorithm each node only performs a single algorithm iteration per synchronization process, whereas, the modified algorithm performs a hundred iterations. For example, for estimating the *x-coordinate*, Figure 5.42a, the original algorithm needs to perform 200 synchronization processes to present an error around 1 cm which represents a total of 200 algorithm iterations. On the other hand, the modified algorithm achieves the same error after 50 synchronization processes which represents a total of 5000 algorithm iterations. This demonstrates that the number of messages exchanged was reduced considerably, however, the algorithm needs more computation time.



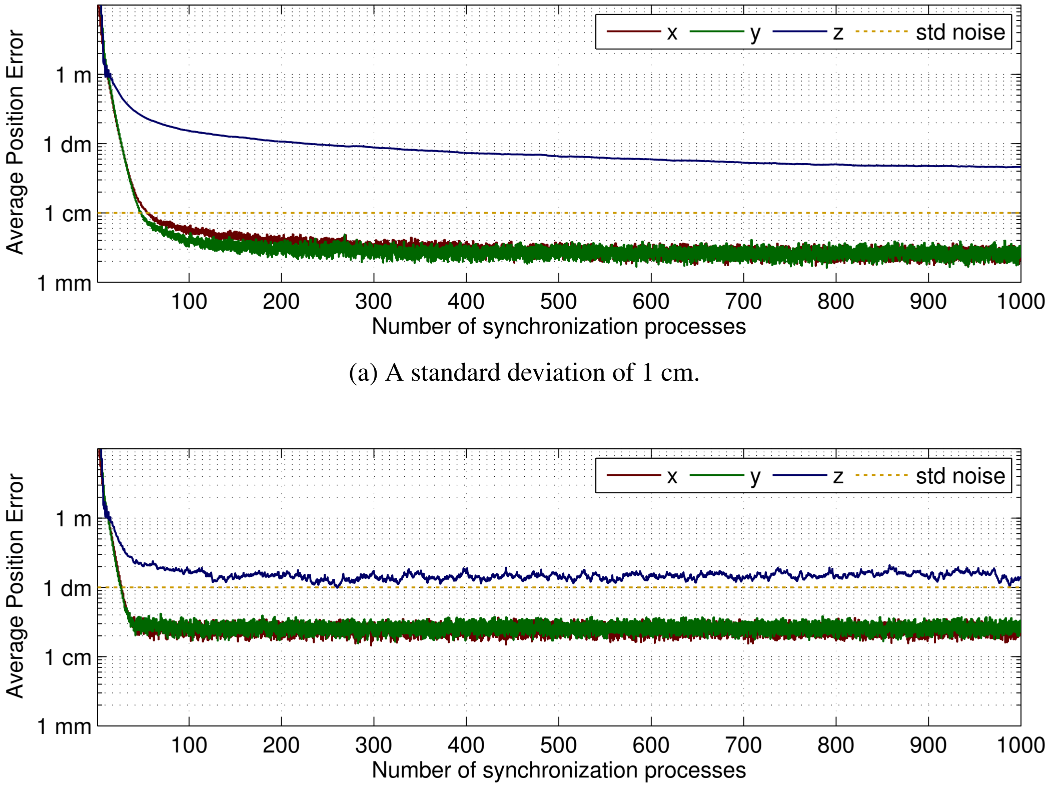
(a) *x-coordinate*.

(b) *z-coordinate*.

Figure 5.42: Comparison between the original dw-MDS and the proposed modified dw-MDS algorithm.

Distance Error

The previous example with fourteen anchor beacons was used to evaluate the impact of the distance error in the beacons position estimate. In this experiment, it was considered that distance measurements contain additive white Gaussian noise with two different standard deviations: 1 cm and 10 cm. The results are presented in Figure 5.43. As it can be seen, the results show that the algorithm presents an excellent performance in the presence of noise. The resultant position error is less than the noise standard deviation. Only the *z-coordinate* presents an error bigger or of the same order of the noise standard deviation. Nevertheless, these errors are in line with the error presented by the algorithm without noise, see Figure 5.38b.



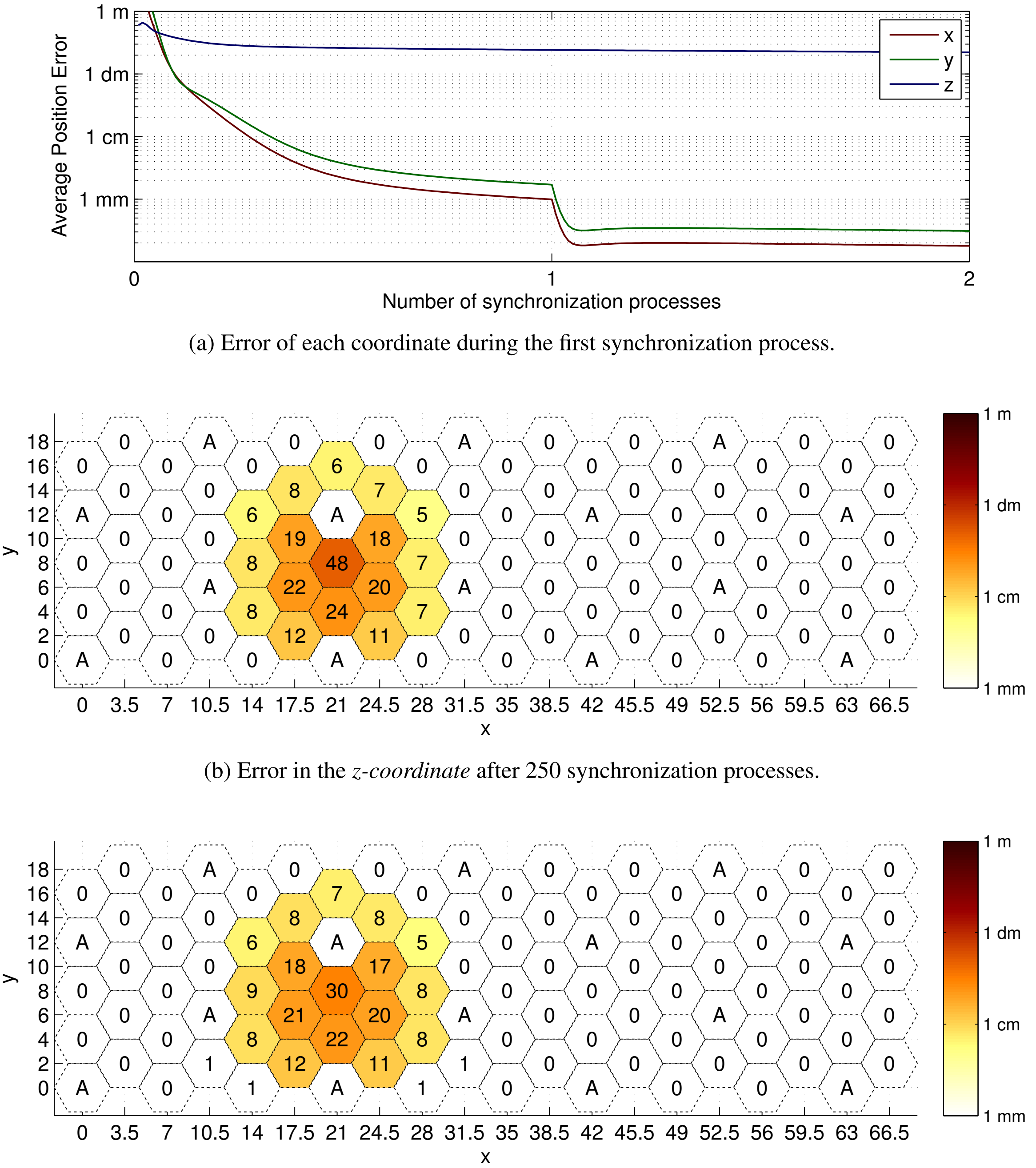
(a) A standard deviation of 1 cm.

(b) A standard deviation of 10 cm.

Figure 5.43: Average position error for each coordinate as a function of the synchronization processes considering that the distance measurements contain additive white Gaussian noise with two different standard deviations.

Beacon Failure

The previous example was used here to evaluate the impact of the beacon failure in the beacons position estimate. In this experiment it was considered that only beacon 33 fails and its replaced by a new beacon with random position. The results are presented in Figure 5.43 where the number of synchronization process are counted after the beacon starts its location algorithm. Moreover, it was considered that at 0 synchronization process the beacon receives the position and the distance information to its neighbors. As one can see, the results show that even before the first synchronization process beacon 33 position error is rapidly reduced to a few millimeters. However, due to planar geometry of the network the error in the *z-coordinate* remains considerably high. However, this error is reduced to less than 5 cm after 250 synchronization processes and to 3 cm after 1000 synchronization processes, as it can be seen in Figures 5.44b and 5.44c. Additionally, it can also be observed that the beacon failure will affect the neighbors' position. Nevertheless, its impact is small for beacons that are not in the neighborhood of the failure beacon.



(a) Error of each coordinate during the first synchronization process.

(b) Error in the *z-coordinate* after 250 synchronization processes.

(c) Error in the *z-coordinate* after 1000 synchronization processes.

Figure 5.44: Impact in coordinates errors after the replacement of the failure beacon 33.

CHAPTER 6

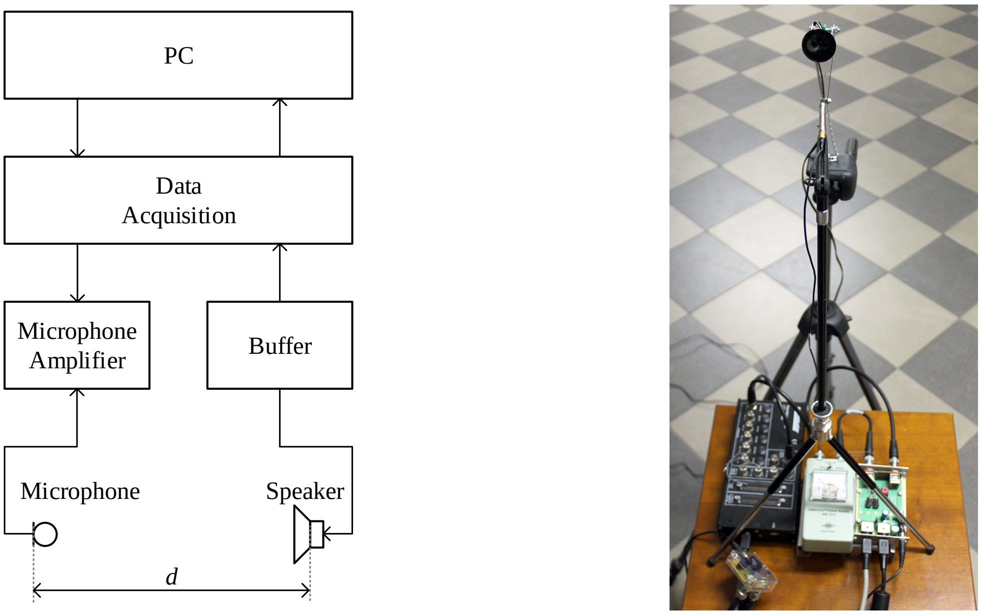
IMPLEMENTATION AND VALIDATION

In this chapter some final results that were not covered in the previous chapters will be presented. This chapter starts with the validation of the proposed models both for the simulator and for the detection pulse, presented in Chapter 3 and 4 respectively. At the end, the beacons location and synchronization process will be combined with the mobile location process, which were presented separately in Chapter 5.

At the beginning of this chapter, Section 6.1 will present all the hardware that will be used for the models validation and verification of the validity of some assumptions that were made. This hardware is going to be used in Section 6.2 to validate the simulator's model from Chapter 3 and in Section 6.4 to validate the noise model and the pulse detection equations, presented in Chapter 4. In Section 6.3 the proposed simulator will be compared to another free simulator also implemented in Matlab. Finally, Section 6.6 will present an example where the proposed location system will be tested, combining the beacon synchronization and location together with mobile location.

6.1 Practical Tests

The system measurement chain presented in Figure 6.1a will be used to perform the practical tests. Where, the distance between the microphone and the speaker could be fixed or variable depending on the type of measurement. Each block of the system measurement chain will be presented and detailed in the further subsections. Figure 6.1b presents the real devices used in the measurement system.



(a) Measurement system chain.

(b) Real devices.

Figure 6.1: Measurement system used for practical tests.

6.1.1 PC and Data Acquisition

A PC with the Matlab Software package that communicates with the data acquisition hardware, DT9836S-6-2-BNC, from Data Translation[1] (see Figure 6.2) was used for sending, receiving and processing the ultrasonic waves. The DT9836S is a data acquisition module that provides simultaneous analog input with a sampling frequency up to 800 KSamples/s per channel and simultaneous analog output with a sampling frequency up to 500 KSamples/s per channel.

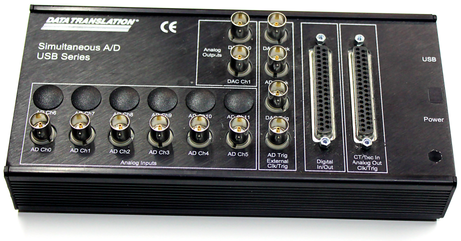
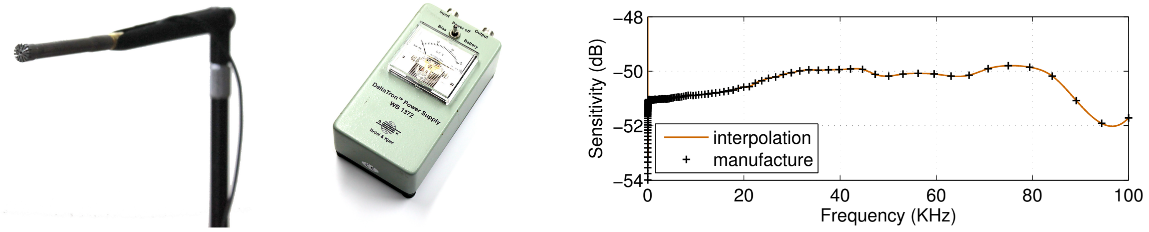


Figure 6.2: Data acquisition hardware.

6.1.2 Microphone

The Brüel & Kjær 4954-A microphone was chosen. The microphone, its power supply and its frequency response are shown in Figure 6.3. This microphone was chosen because it is a calibrated microphone that operates in the ultrasonic range (up to 100 kHz), with a well-known frequency response provided by the manufacturer, see Figure 6.3c.



(a) Microphone. (b) Power Supply. (c) Frequency Response.

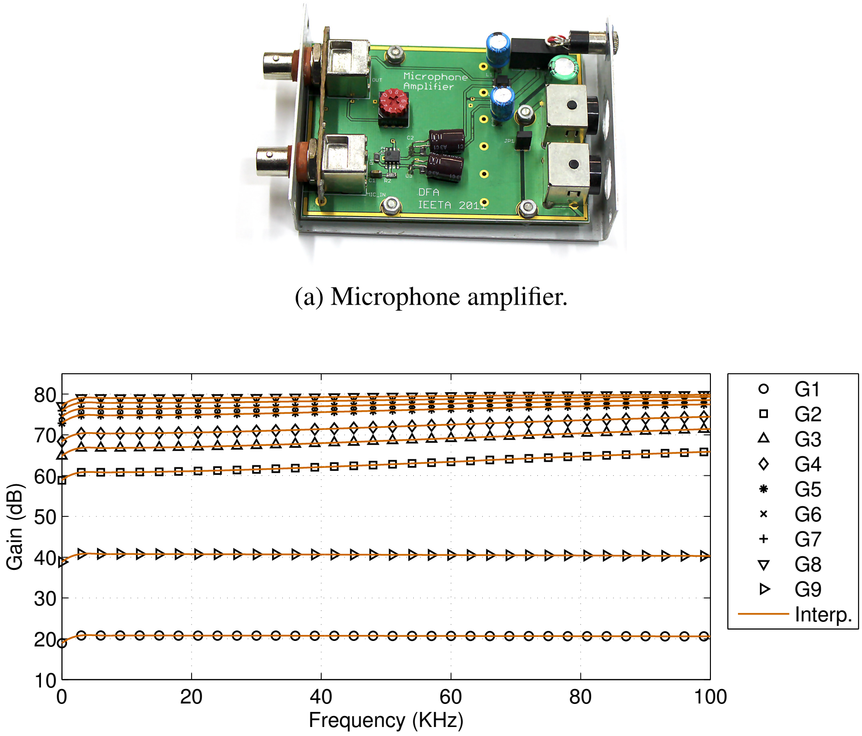
Figure 6.3: Brüel & Kjær 4954-A microphone, its power supply and its frequency response.

6.1.3 Microphone Amplifier

The microphone amplifier was designed by the author in order to condition the microphone output signal to the dynamic range of the data acquisition hardware. This microphone amplifier has nine preset gains that can be manually chosen between 20 dB and 80 dB. A photo of this microphone amplifier and its frequency response for the nine gains[2] is depicted in Figure 6.4. More details about this microphone amplifier can be found in Appendix A.

[1] Data Translation is a designer, manufacturer, and provider of data acquisition solutions for test and measurement.

[2] The amplifier's frequency response was measured with the Audio Precision Portable One Plus Audio Measurement System.



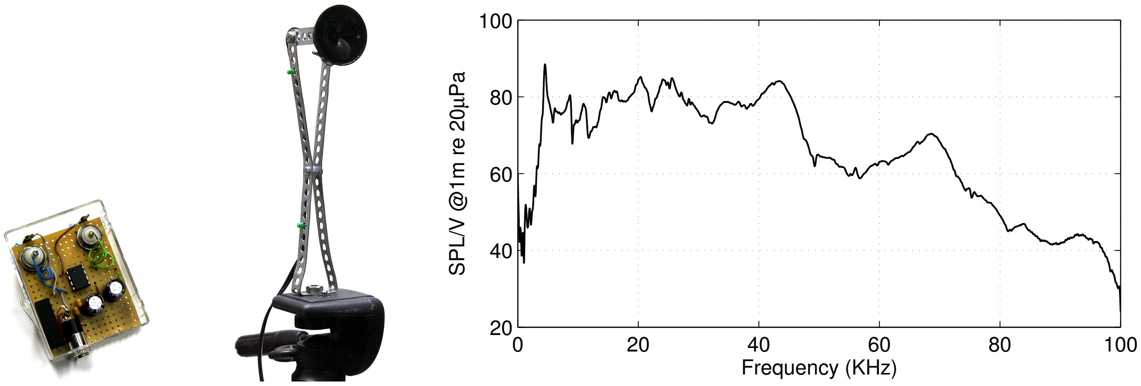
(a) Microphone amplifier.

(b) Frequency response of the amplifier.

Figure 6.4: Microphone amplifier and its frequency response for various preset gains (G1–G9).

6.1.4 Buffer and Speaker

The Buffer is just an audio driver to avoid overload the data acquisition output. The speaker that was used is a Kemo L10, which has an excellent frequency response between 15 kHz and 45 kHz, nevertheless it can produce waves up to 80 kHz, as can be seen in Figure 6.5c. The buffer and the speaker used are depicted in Figure 6.5.



(a) Buffer.

(b) Speaker.

(c) Frequency response of the speaker.

Figure 6.5: Speaker buffer and speaker Kemo L10 and its frequency response.

6.2 Simulator Validation

This section presents the simulator validation procedure, where several tests will be carried out in order to validate each simulator's model. It starts by testing and validating the sound speed equation. After that, the air attenuation expression especially the air attenuation regarding the air absorption is validated. At the end, the behavior of the simulator in the presence of a single reflection will be evaluated. The output of the simulator obtained for complex room environments will be compared with measurements obtained in real rooms.

6.2.1 Sound Speed

In order to validate Equation 3.6 two different experiments will be carried out:

first experiment: 25 measurements of the TOF were obtained for a fixed distance between the speaker and the microphone and for a constant room temperature;

second experiment: 5 measurements of the TOF were obtained for a fixed distance between the speaker and the microphone and for six different room temperatures.

Both experiments used the measurement scheme presented in Figure 6.6 and a 20 seconds OFDM pulse sampled at 300 kHz with two million and one carriers from 0 Hz to 100 kHz.

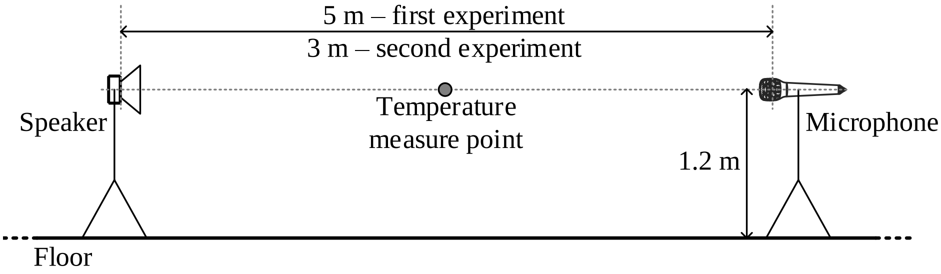


Figure 6.6: Scheme of the sound speed measurement experiments.

For detection proposes the technique presented in Section 4.3 and a pulse with a very low PAPR of 1.6664 were used. This can be confirmed by observing the amplitude histogram of the pulse samples depicted in Figure 6.7.

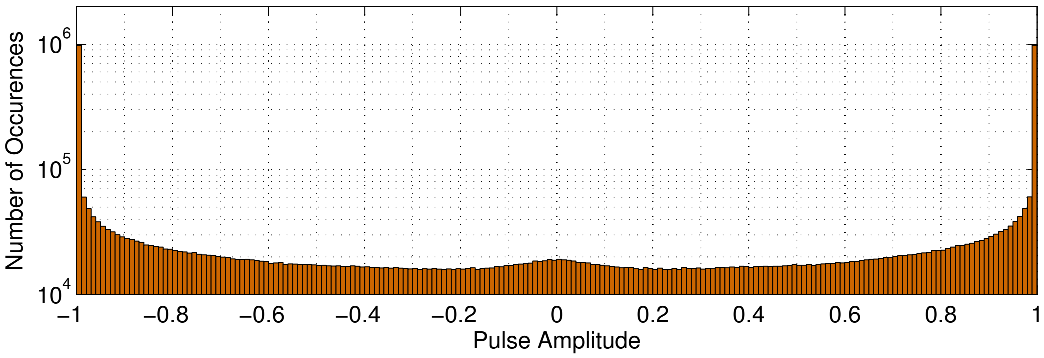


Figure 6.7: Amplitude histogram of the OFDM pulse samples used for sound speed measurement.

First Experiment

In this experiment, the speaker and the microphone were placed 5 m apart, as depicted in Figure 6.8. At this distance, the OFDM pulse was simultaneously transmitted and recorded for 21 seconds. This procedure was carried out twenty five times in intervals of one minute to reduce any error that could be produced by possible echos. Moreover, the temperature was monitored by the author several times during the whole experiment to verify that it was constant. The pulse was detected using a matched filter and the threshold was set to one hundred times bellow the maximum amplitude of the matched filter output, as presented in Figure 6.9. Additionally, in order to increase the accuracy in the time of flight measurement, the match filter output was interpolated using cubic spline interpolation. The measurements are presented in Figure 6.10 where the error is less than one sample.

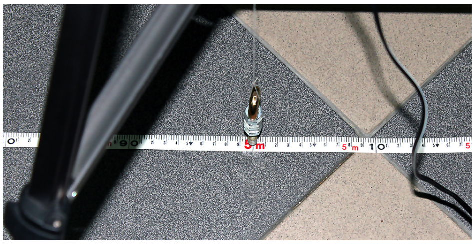


Figure 6.8: Speaker and microphone 5 m apart.

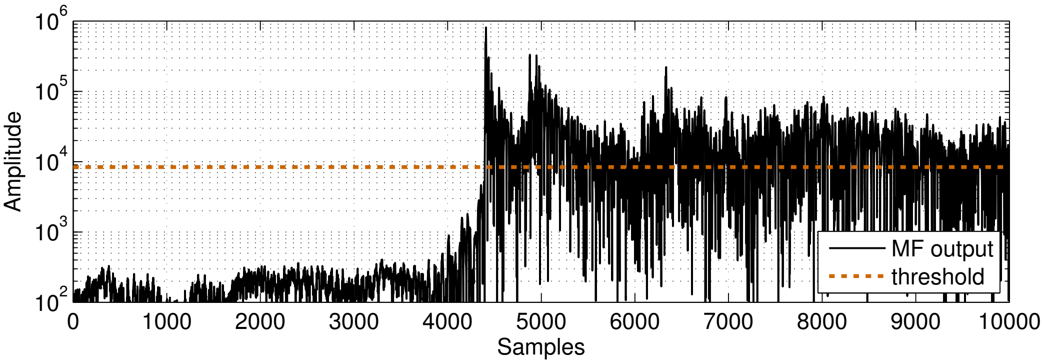


Figure 6.9: Signal obtained at the matched filter output and the threshold value used to measure the sound speed at 5 m.

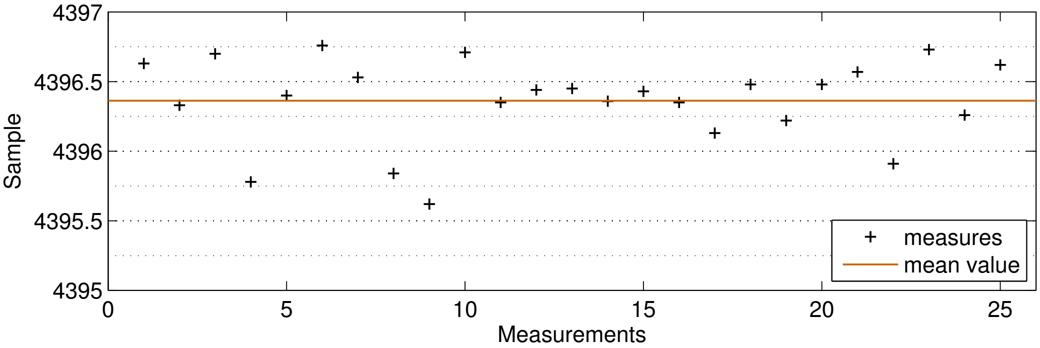


Figure 6.10: Time-of-flight measurements and mean value.

The resultant mean value is 4396.363 which represents a time of flight of 14.65 ms and, consequently, a sound speed of 341.19 m/s. From Equation 3.6, the correspondent temperature is 16.09 °C which is within the accuracy of the information presented by the thermometer, see Figure 6.11.

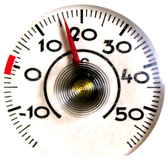


Figure 6.11: Temperature presented in the thermometer during the first experiment.

Second Experiment

In this experiment, the speaker and the microphone were placed 3 m apart in a small room. To control the room temperature a domestic heater was used. The sound speed was measured using the same procedures in the first experiment, however, for each room temperature the TOF was only measured 5 times. The number of measurements was reduced to maintain the room temperature as constant as possible. The estimated room temperatures are presented in Figure 6.12 where they are compared to the information presented in the thermometer. One can see that the results match very well with the information presented in the thermometer.

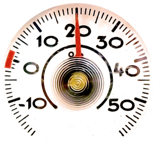
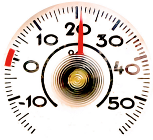
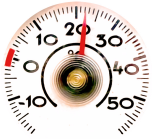
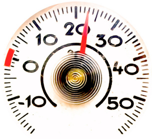
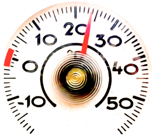
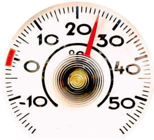
					
343.5 m/s	344.1 m/s	344.8 m/s	345.2 m/s	345.5 m/s	346.4 m/s
20.03 °C	**21.00 °C**	**22.21 °C**	**22.96 °C**	**23.50 °C**	**25.05 °**

Figure 6.12: The temperature obtained from the TOF measurements and the temperature presented in the thermometer during the second experiment.

6.2.2 Air Attenuation

In order to validate the air attenuation due to absorption a very simple experiment was carried out in a wide space. For that propose a sports pavilion was used as presented in Figure 6.13.

Figure 6.13: Sports pavilion used to measure the air attenuation due to absorption.

The microphone was placed at a fix position at 1.45 m from the floor and the transmitted signal was recorded for 46 equally spaced positions, starting at 1 m from the source and ending at 5.5 m from it as depicted in Figure 6.14.

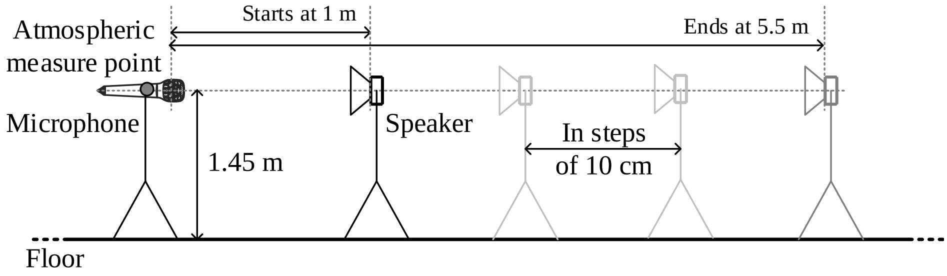


Figure 6.14: Scheme used to measure the air attenuation due to absorption.

The OFDM pulse used in this experiment had the same characteristics of the one used previously (see Section 6.2.1). The IR was obtained by correlation and the frequency response by the Fourier transform (FT) of a thousand samples window of the resultant IR. The IR windows were used to avoid the possible reflection from the floor. Each resultant frequency response was compared with the previous one, e.g. the frequency response from position 1.1 m was compared with that obtained in position 1 m and the frequency response from position 1.2 m was compared with that of position 1.1 m and so forth. Additionally the attenuation due to dispersion was estimated and removed from each comparison. To reduced the noise from the measurements, all frequency response comparisons were divided in groups of 1.5 kHz of bandwidth resulting into 55 groups each one with 270 measurements. The average for each group was computed and depicted in Figure 6.15. In the same figure the expected expression for the pavilion ambient conditions was also presented. In spite of the noisy measurements, the average results match very well the expected value for the atmospheric absorption.

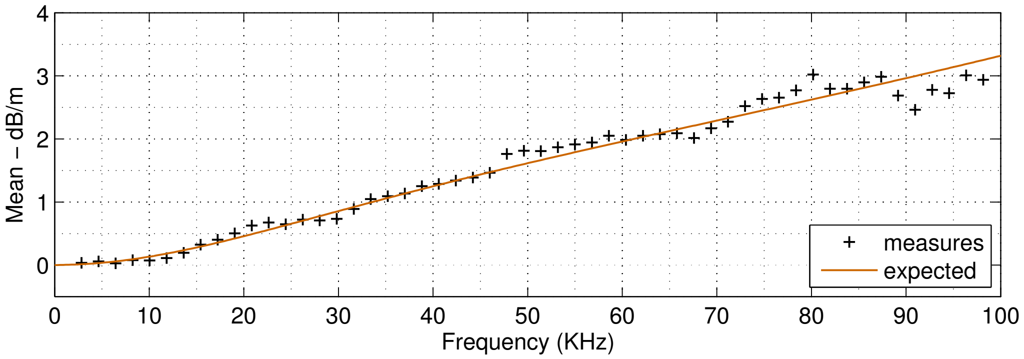


Figure 6.15: Comparison between the measured atmospheric absorption and the expected theoretical value.

The original measurements are presented in Figure 6.16. As it can be observed the attenuation measurements present considerable noise, especially above 70 kHz. This may arise from several factors, the most noteworthy are the imprecision in the distance measurements and the impossibility to keep the speaker direction constant to avoid the speaker beam pattern effect. The extra noise above 70 kHz may also arise from the very low power produced by the speaker, see Figure 6.5.

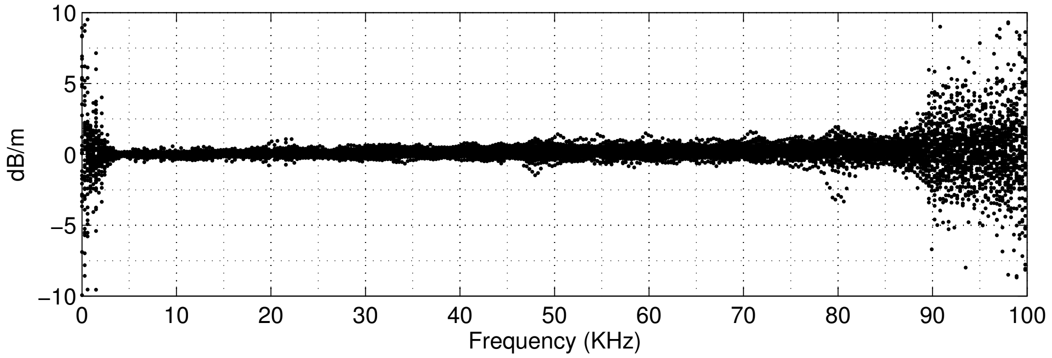


Figure 6.16: Atmospheric attenuation due absorption that results from the frequency response comparison for 46 different distances speaker-microphone.

6.2.3 Single Reflection

To test the simulator behavior a very simple example with a single reflection will be described. The simulator impulse response will be compared with the real impulse response obtained under the same circumstances. The same real environment used in the air attenuation test will be used (see Section 6.2.2), but a fix distance of 5.5 m will be considered. Figure 6.17 shows the scheme of the measurement system used. An environment as similar as possible to the real one was created in the simulator (see Figure 6.18).

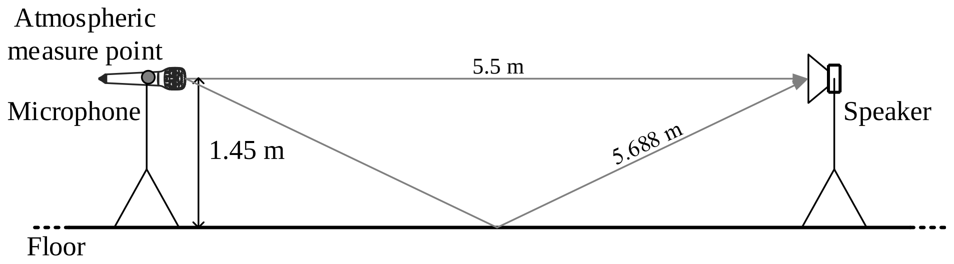


Figure 6.17: Scheme used to test the simulator for one reflection.

Additionally, it was also considered:

- the same ambient conditions (i.e.: temperature, pressure and humidity);
- an omni-directional microphone;
- the beam pattern model of a similar ultrasonic speaker;
- a reflection coefficient of 0.7 for the floor.

The real impulse response was obtained by correlation. The resultant IR is compared in Figure 6.19 with impulse response obtained by the simulator. It can be seen that the impulse responses are very similar, proving that the simulator output is very similar to the IR obtained in the field.

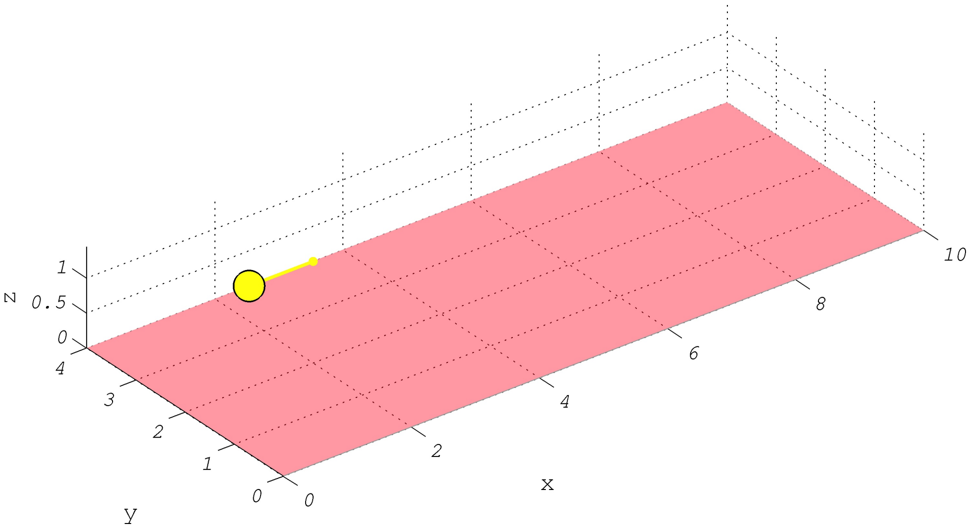
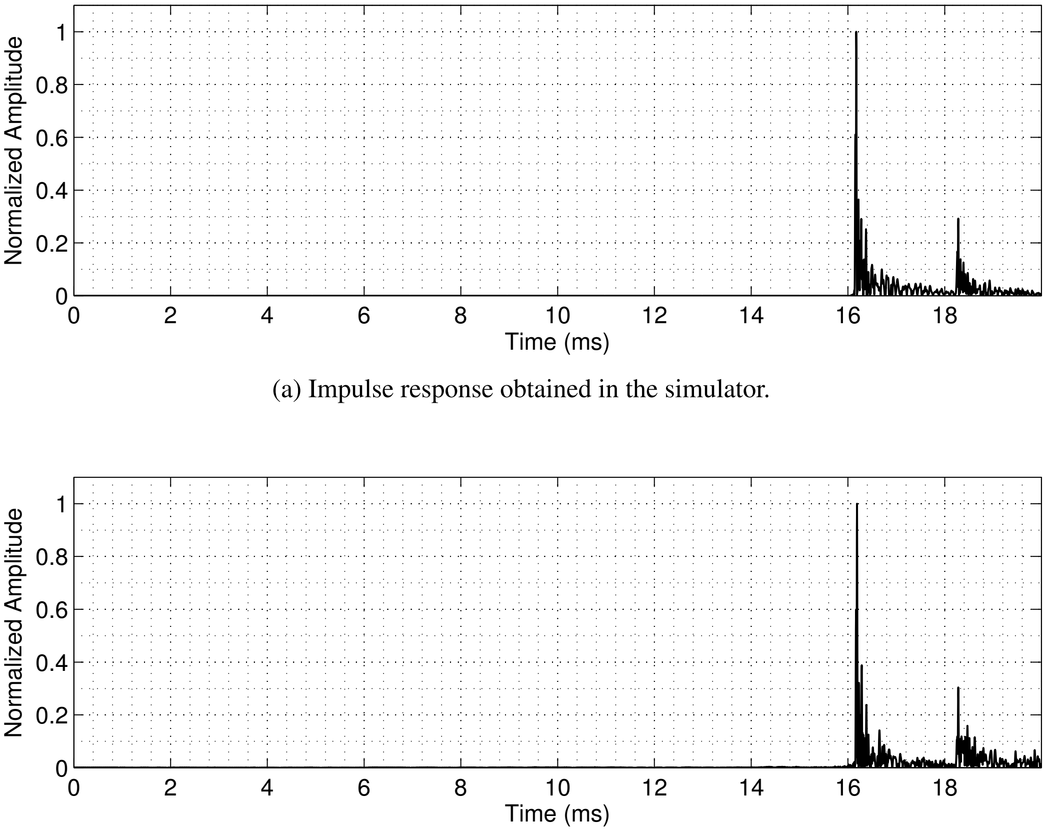


Figure 6.18: Simulation environment used to test the simulator for a single reflection case.



(a) Impulse response obtained in the simulator.

(b) Real impulse response obtained by correlation.

Figure 6.19: Impulse response comparison for a single reflection.

6.2.4 Simulation of a Complex Room

In order to test the simulator in a complex room, the room environment presented in Figure 6.20 was used. The room has the following external dimensions 6×10×3 m, in the middle of the room there is a 3 m long wall. The walls and the ceiling are made from medium-density fiberboard (MDF) and the floor from concrete. The speaker was placed in the inside wall corner, near the ceiling pointing in the direction of the room corner (3, 0, 0) as shown in Figure 6.20. The impulse response was measured in five different places in the room, as presented in Figure 6.21, at 1.5 m from the floor. A measuring tape was used to place the microphone. Therefore the microphone position presents errors of the order of a centimeter. The first four points are inside the same room division of the speaker and the last point is out of the speaker's line-of-sight (LOS). Nevertheless, this point can receive reflected waves from walls $x = 3$ m and $y = 10$ m. The results for this five points are presented in Figure 6.22, where the left column presents the impulse response, obtained from the simulator and the right column the real impulse response obtained by correlation.

In spite of some differences, especially in the impulses amplitude, it can be said that the impulse response obtained in the simulator matches very well the impulse response obtained in the field. A careful analysis to the results reveals that the main echos of simulator IRs (the ones with a considerable amplitude in comparison to the direct wave) have a similar amplitude and arrive at the same time that the ones in the real IRs. The amplitude difference could be related to some reflection coefficient difference over the wall material and the roughness of concrete floor. Another important issue that can be observed in P_5 and the simulator does not take it into account is the ultrasonic propagation through wall. In P_5 all the impulse response spikes before 25 ms are related to ultrasonic waves that cross the inside wall, because the first reflection, considered in the simulator, comes from the wall $x = 3$ m which imposes a propagation time around 8.5 m (around 25 ms of delay), 3 m from the speaker to the wall and 5.5 m from the wall to the microphone. The distance between the speaker and the microphone is around 2.55 m resulting into a 7.5 ms time delay therefore the first spike in the real impulse response arises from the direct path. Therefore, in this real scenario, in the case of NLOS, the system could measure the distance between the speaker and the microphone with a small error, whereas, the simulator result presents a worst case where the estimation for the distance between the speaker and the microphone would be completely wrong.

Nevertheless, these small differences do not compromise the realism of the simulator, on the other hand, they reinforce the idea that the simulator is a very important tool to estimate the result of the ultrasonic signal propagation in a particular room.

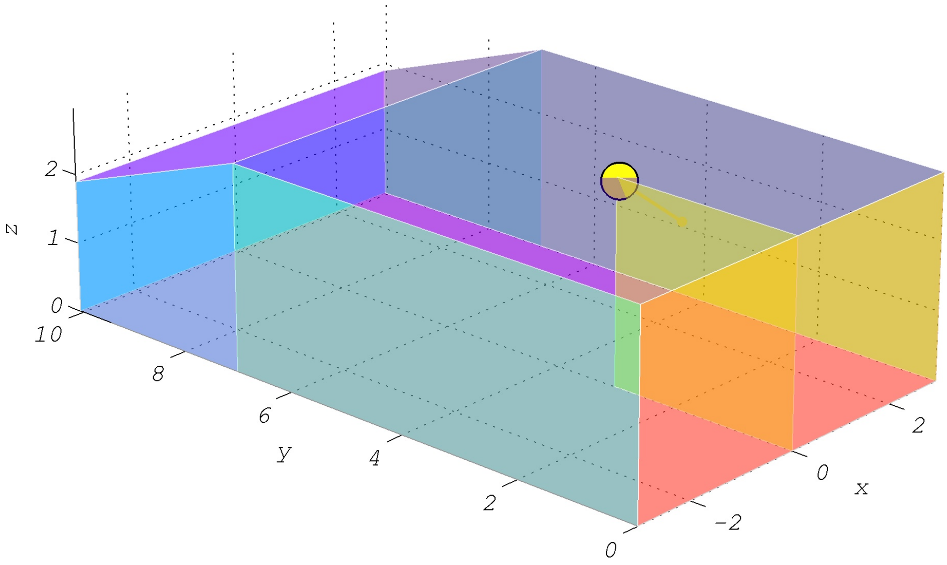


Figure 6.20: Simulation environment used to test the simulator in a complex room.

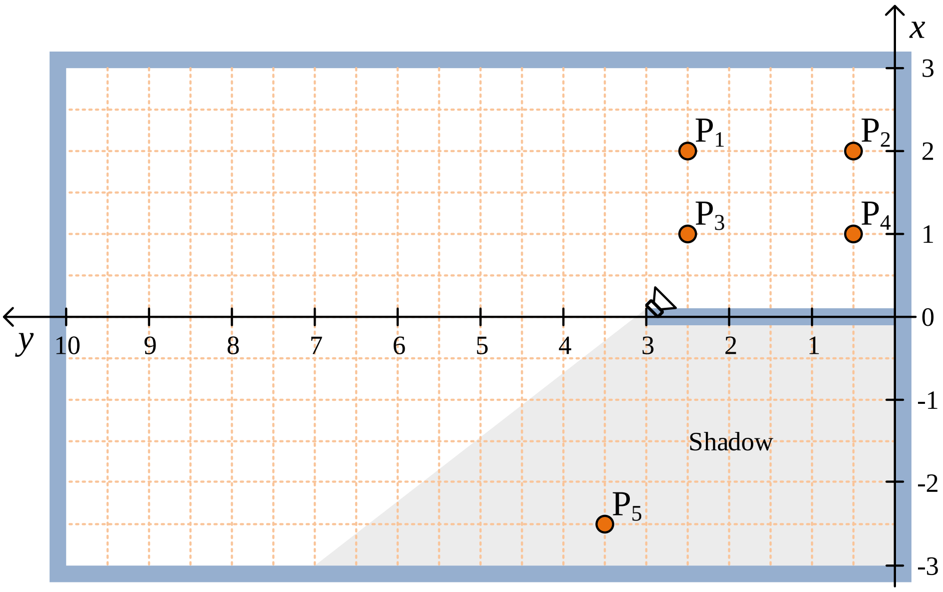


Figure 6.21: Measurement points in the test of a complex room.

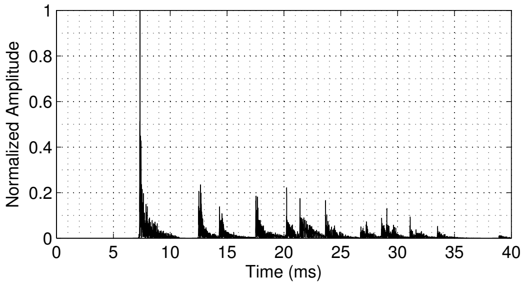
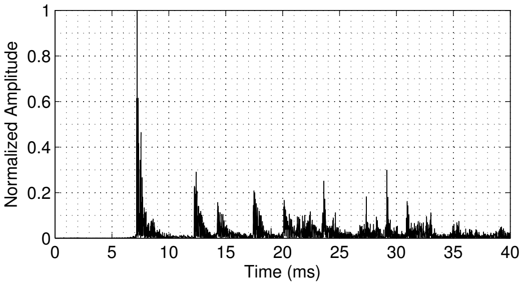
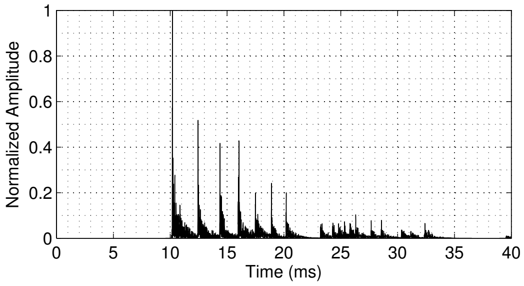
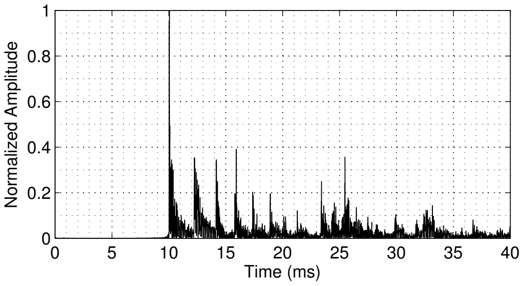
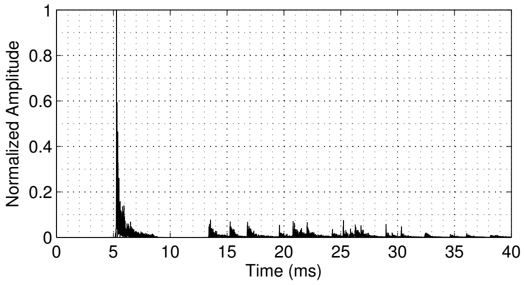
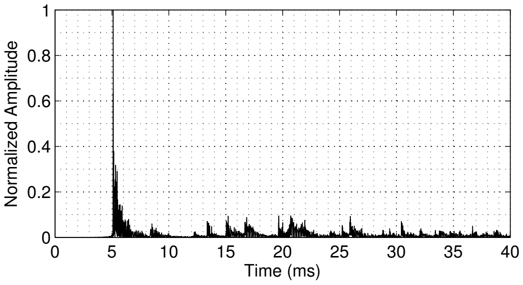
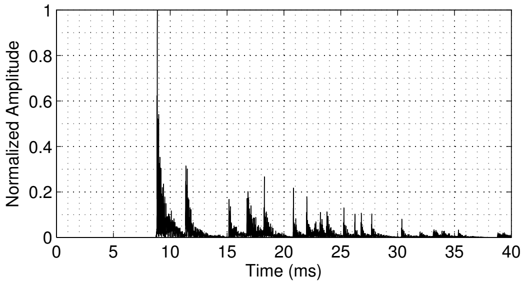
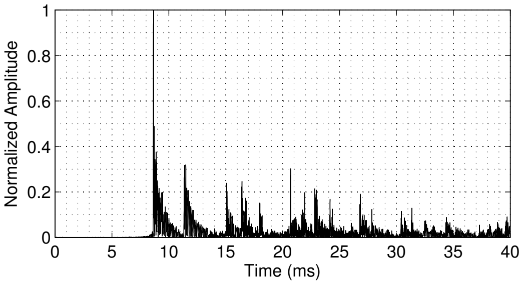
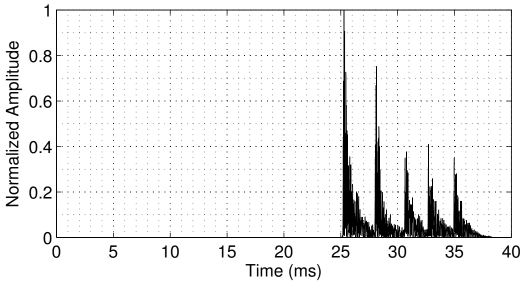
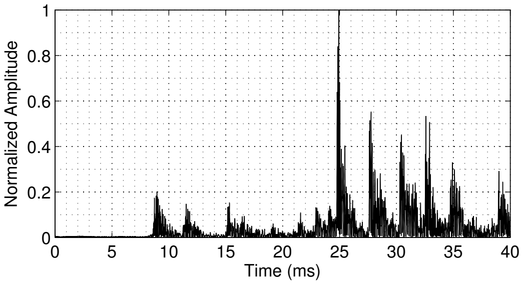
	Simulator Impulse Response	Real Impulse Response
P$_1$		
P$_2$		
P$_3$		
P$_4$		
P$_5$		

Figure 6.22: Impulse response comparison for a complex room shape.

6.3 Simulator Comparison

In this section the proposed simulator will be compared with another free available simulator also implemented in Matlab. The chosen simulator was implemented by Eric Lehmann[3] and modified by Fernando J. Álvarez Franco[4] to include atmospheric absorption, the filtering effect of real transducers, and also the Doppler shift caused by the receiver movement [84]. This simulator uses the Allen & Berkley's image source method [85] to determine all the virtual sources that produce reflections. The simulator computes the impulse response and the frequency response for a given speaker/microphone position. The algorithm was implemented in the frequency domain and thus allows fractional delays for each image source. This simulator was, further, modified to include the attenuation due to propagation losses, similar to the proposed simulator. The main differences between both simulators are summarized in Table 6.1.

Due to the limitations of the modified Lehmann's simulator the comparison between both simulators will focus only in an rectangular box shape room without objects. For that purpose it was implemented in both simulators a room with the following dimensions: 8 × 4 × 3 m (length × width × height). Moreover, the speaker was placed near the room corner in (1, 1.5, 1) as presented in Figure 6.23.

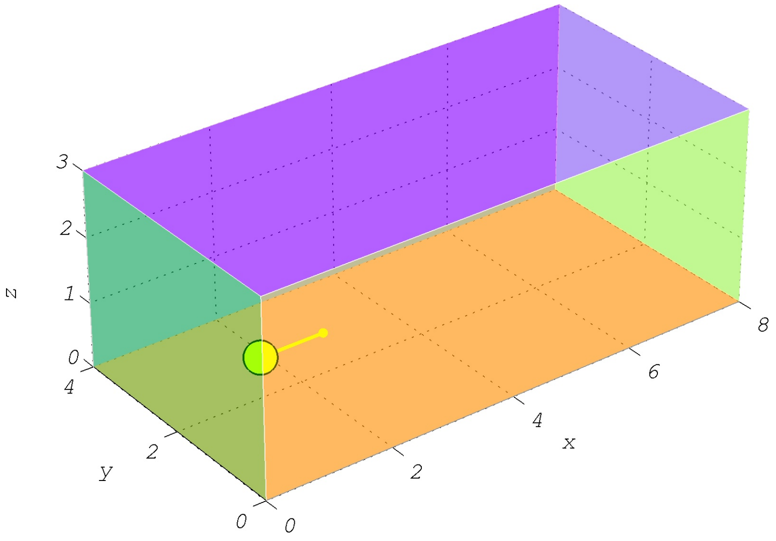


Figure 6.23: Simulation environment for simulators comparison.

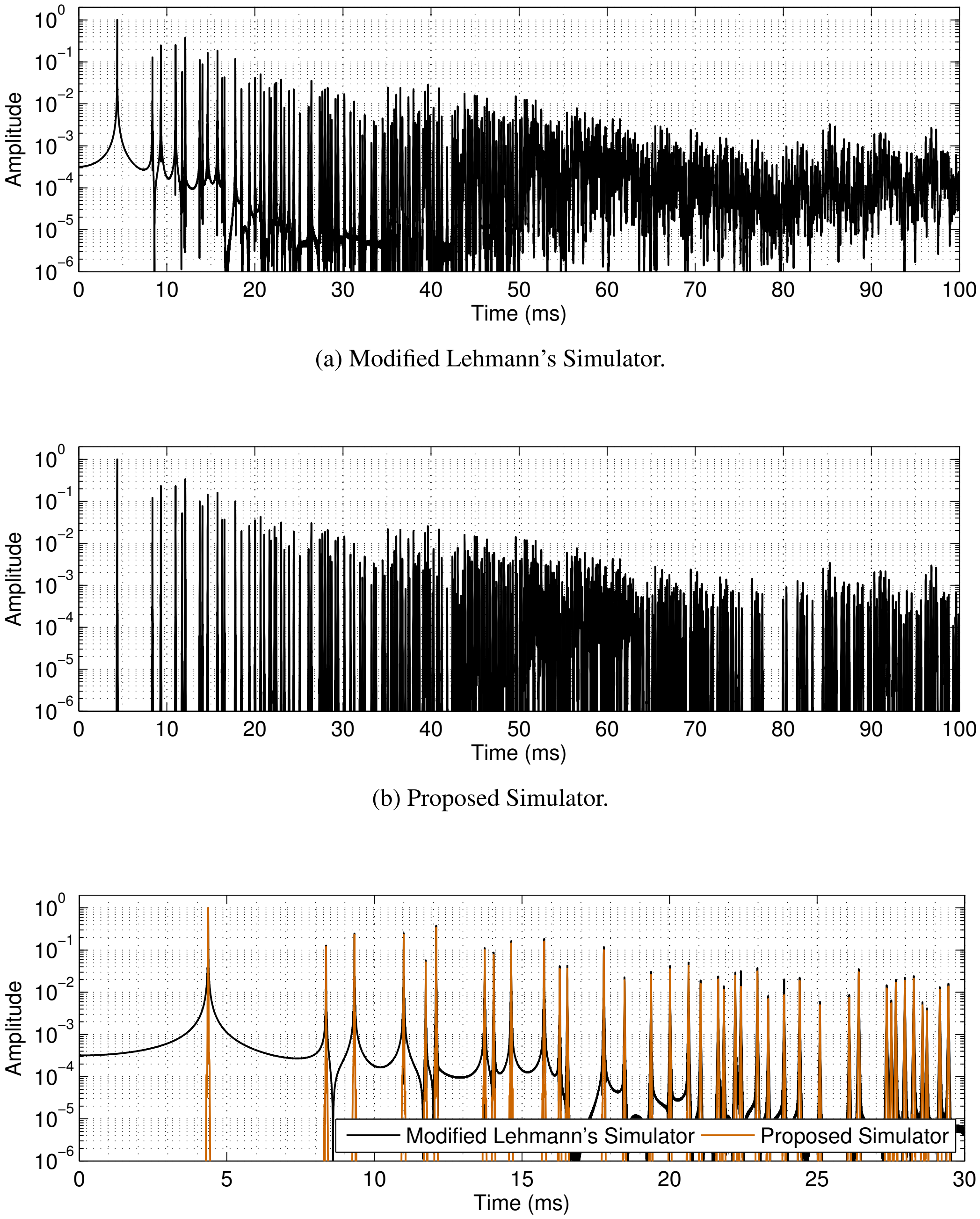
In this simulation environment, it is considered that all the room walls present a reflection coefficient of 0.7, the ceiling 0.9 and the floor 0.3. Additionally, both, the speaker and the microphone, are considered omnidirectional and the microphone is placed at 1.5 m from the speaker in position (2, 2.5, 1.5). The proposed simulator will stop when eight successive reflections are accomplished and the modified Lehmann's simulator will stop when the reflections amplitude are 60 dB bellow the direct wave amplitude. The resultant impulse responses for both simulators are presented Figure 6.24.

[3] Eric Lehmann works as a Research Scientist for the Commonwealth Scientific and Industrial Research Organisation (CSIRO) in Perth, within the division of Mathematics, Informatics and Statistics.

[4] Fernando J. Álvarez Franco is an Associate Professor at the Department of Electrical Engineering, Electronics and Automatics, Faculty of Science. University of Extremadura.

	Modified Lehmann's Simulator	*Proposed Simulator*
Outputs	Impulse response and frequency response.	Impulse Response only.
Sound Speed	Must be provided to the simulator.	Computed by the simulator, for a given room temperature.
Fractional Delay	Implemented in the frequency domain.	Implemented using fractional delay filters.
Room Shape	Only supports a unique room shape. The rectangular box shape.	Allows complex room shapes. Any room shape may be implemented as long as the walls can be modeled as quadrilaterals.
Reflection Coefficient	Possible to attribute a reflection coefficient value to each wall.	Possible to attribute a reflection coefficient impulse response to each wall, allowing frequency variability.
Room Objects	Not allowed.	Allows complex object shapes. It is possible to implement any object shape when the surfaces can be modeled as quadrilaterals.
Source Validation	Very simple but effective due to room geometry.	Necessary a ray tracing validation due to room geometry complexity and the presence of objects.
Speaker and Microphone Model	Very limited. It is only possible to apply the speaker or the microphone beam pattern in the resultant impulse response. Due to this limitation it is not possible to simulate directional beam patterns.	Possible by providing the speaker and/or microphone beam pattern to the simulator
Multiple Speakers and Microphones	Only possible to simulate a single speaker and microphone.	Possible to simulate any number of speakers. However it is only possible to simulate a single microphone.
Room Display Tools	None.	Possible to obtain the room with all the real sources or with all real and virtual sources.
Stop Criterion	The maximum amplitude for a reflection to be considered.	The maximum number of successive reflections.
Band Limited Optimization	Not implemented.	Possible to optimize the resultant impulse response for a specific frequency band, reducing the error in that particular band.
Doppler Effect	Only possible to implement a rectilinear uniform movement.	Possible to implement any movement.

Table 6.1: Main differences between the proposed simulator and the modified Lehmann’s simulator.



(a) Modified Lehmann's Simulator.

(b) Proposed Simulator.

(c) The first 30 ms zoomed.

Figure 6.24: Comparison between the impulse response of the modified Lehmann's simulator and that of the proposed simulator.

It can be seen that both impulse responses are almost identical both in amplitude and in position. Nevertheless, there are some differences especially around the delay 80 ms where the proposed simulator does not present some reflections. This is related to the stop criterion that does not take into account this higher order reflections.

Another important difference is related to the fractional delay implementation where, in the modified Lehmann's simulator, each impulse presents a tail, associated to frequency domain implementation and the use of the inverse Fourier transform to obtain the impulse response. The author had used this method to allow the implementation of fractional delays. Nevertheless, the approach used in the proposed simulator presents a better delay error than the modified Lehmann's simulator, especially in the bandwidth of interest, which was set to be between 1 kHz and 80 kHz, see Figure 6.25.

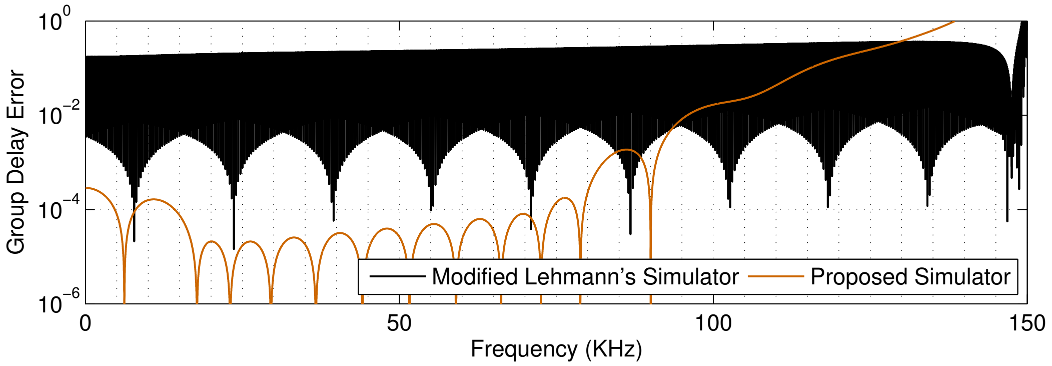


Figure 6.25: Delay error for propagation delay of 2 m (around 1747.5 samples).

Moreover, the impulse response's slow decay of the modified Lehmann's simulator creates a kind of tail around each impulse. This tail may produce a shadow effect over small impulses. Regarding Figure 6.26 the presence of the second weak reflection is almost imperceptible because its amplitude is the same order of the tail produced by the first strong reflection. This effect has a small significance if the smallest impulse is a reflection. However, if it is the other way around, the smallest pulse represents the direct path (e.g., as a result from the speaker beam) this could represent not detecting it, resulting into a wrong distance estimation by considering the reflection impulse as the direct path impulse.

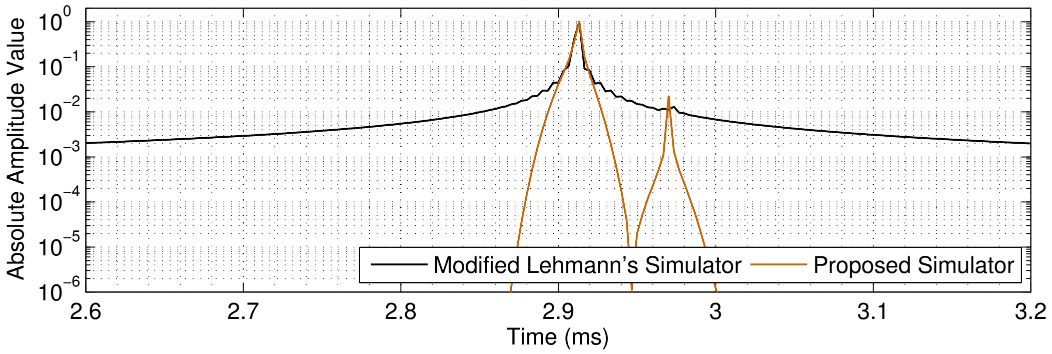


Figure 6.26: Shadowing effect produced by the slow decay of a strong impulse.

6.4 Probability of Detection

Before evaluating the probability of detection it is important to evaluate the validity of some assumptions that were presumed in the probability of detection analysis. All the theoretical analysis was based on the fact that noise is additive white Gaussian noise. Moreover, it was considered that the DFT of the noise is a complex variable where its real and imaginary part are also random variables with zero mean, and a certain variance, see Section 4.3.3. To validate this statement ten seconds of noise were recorded in a typical office environment using a sampling frequency of 300 kHz. The DFT of the recorded signal was computed and a normality test was run in the typical ultrasonic band (from 30 kHz to 50 kHz) resulting into three hundred thousand samples to test. For the normality test the well known Lilliefors test was chosen [86]. The Lilliefors test in one of the most used tests to test the null hypothesis that data come from a normally distributed population. It is based on the also well known Kolmogorov–Smirnov test [87]. In contrast to the Kolmogorov–Smirnov test, in the Lilliefors test it is not necessary to specify the mean and variance of normal distribution. The Lilliefors test results, obtained for the real and the imaginary part, are summarized in Table 6.2. The results of the Lilliefors test tell us that it is impossible to prove that the three hundred thousand samples set are not from a normal distribution for a significance level of 5%. The distribution can therefore be considered normal, within that margin. This result corroborates the empirical probability density function for the real and imaginary part presented in Figure 6.27, where it can be observed that the empirical probability density functions match very well the probability density function of a normal distribution with the same variance.

	Real Part	**Imaginary Part**
p-value:	0.126	0.071
Result:	Cannot reject normality	Cannot reject normality
Variance:	9.989×10^{-4}	9.987×10^{-4}

Table 6.2: Lilliefors test result for the 300000 frequency samples, using significance level of 5%.

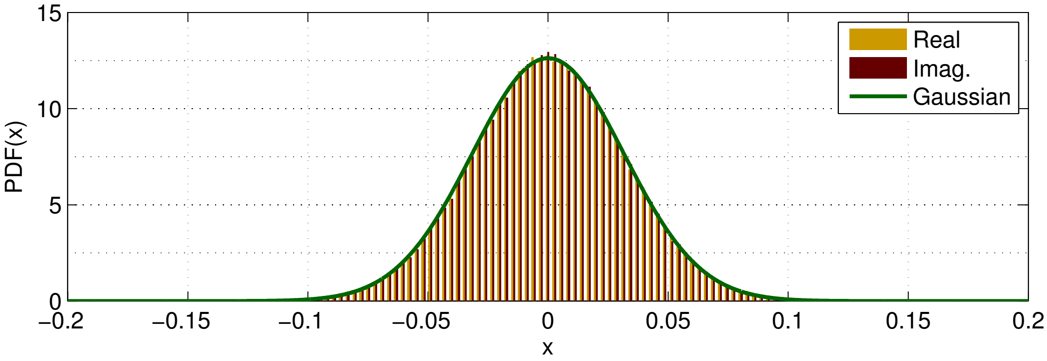


Figure 6.27: Empirical probability density function for the real and imaginary part and the true probability density function for a normal distribution with the same variance.

After this noise model validation it is possible to test the probability of detection presented in Section 4.3.3. For that purpose the scheme presented in Figure 6.28 was used where the speaker and the microphone were placed 1 m apart.

The probability of detection was evaluated for the last pulse sample (i.e. the maximum output value of the matched filter). Twelve difference SNR values around the expected probability of 0.5 were considered. Moreover, for each SNR value the noise energy was estimated and the threshold

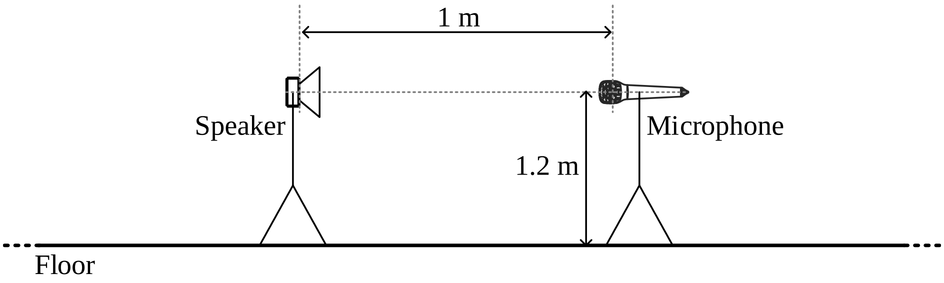


Figure 6.28: Scheme used to evaluate the probability of detection.

set to ensure a probability of false alarm of 10^{-6}, after that fifty trials were carried out. Additionally, the same procedure was implemented in the simulator, whereas a thinner SNR grid was used on a thousand trials. The results of this test are presented in Figure 6.29, where they are compared with the expected values presented in Section 4.3.3.

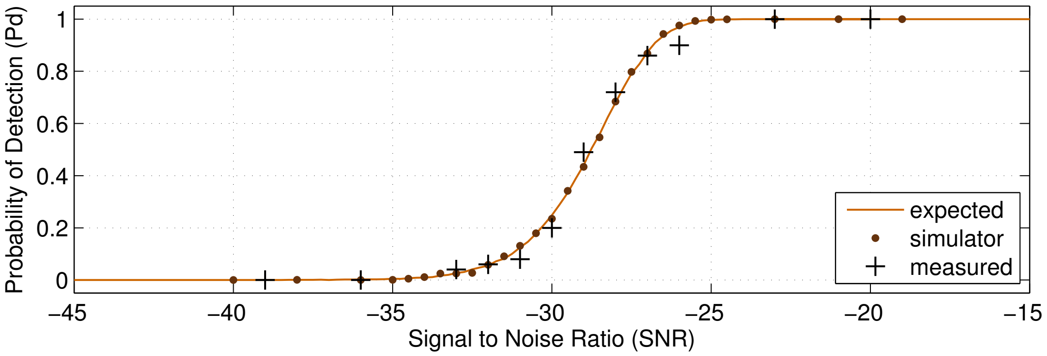


Figure 6.29: Probability of detection estimation using a real setup and the simulator for the last pulse sample considering a probability of false alarm of 10^{-6}.

From the obtained results it can be observed that the probability of detection evaluated in the simulator as well as probability of detection estimated in the real environment follow, with good precision, the expected value. This result validates the mathematical expression presented in Section 4.3 for the probability of detection. Another important result that can be retained from this experiment is that the speaker frequency response influence may be neglected, to a certain extension, from the detection point of view.

6.5 Distance Measurement

To test the accuracy of the proposed system to measure distances two different experiments were carried out:

first experiment: 25 measurements of the TOF were obtained for a fixed distance between the speaker and the microphone and considering a fixed room temperature;

second experiment: The TOF measurements were obtained for distances between the speaker and the microphone varying from 1 m to 5.5 m in steps of 10 cm.

In both experiments a 20 seconds OFDM pulse sampled at 300 kHz with two million and one carriers from 0 Hz to 100 kHz was used. It is the same pulse used to measure the sound speed in Section 6.2.1.

First Experiment

To test the accuracy and repeatability of the proposed system to measure distances, the speaker and the microphone were placed 5 m apart, as depicted in Figure 6.30.

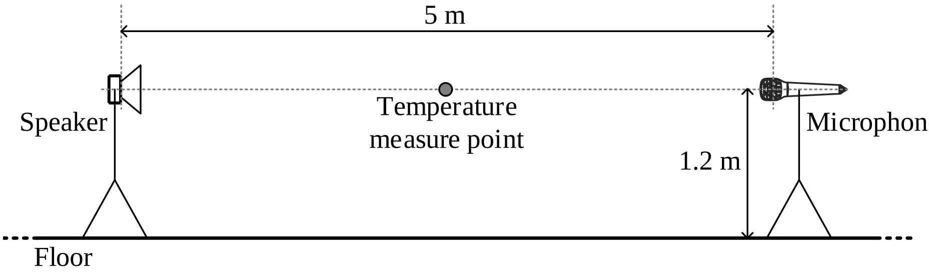


Figure 6.30: Scheme for testing the accuracy and repeatability of the distance measurement process.

At this distance, the OFDM pulse was simultaneously transmitted and recorded for 21 seconds. This procedure was carried out twenty five times with one minute of interval. This interval was used to reduce any error that could be produced by any possible echo. Moreover, the temperature was recorded and the sound speed estimated. The pulse was detected using a MF and a threshold set to one hundred times lesser than the maximum amplitude of the MF output. The results have proved to be extremely repetitive, since the same sample value was obtained for all the measurements. Due to this, it was decided to interpolate the MF output by one thousand times using cubic spline interpolation. The absolute error of the obtained measurements is presented in the box-and-whisker diagram of Figure 6.31. Additionally, the data presents an average error of 258.87 μm. The maximum obtained error is less than a millimeter and the average error is around a quarter of a millimeter. This is an excellent result and it directly arises from the pulse size and its huge bandwidth. Note that the room temperature was constant over all the trials, which contributed to a very low distance dispersion. Nevertheless, this result demonstrates all the potentiality of using ultrasonic signals for distance measurement purposes.

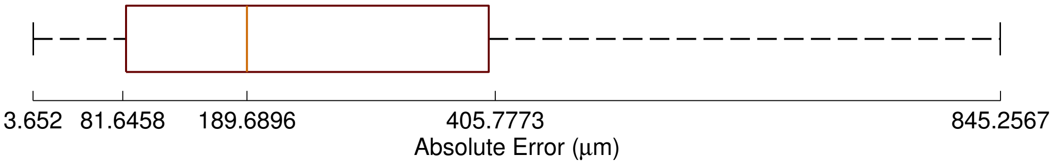


Figure 6.31: Box-and-whisker diagram of the absolute error for 25 trials in the distance measurement of 5 m. The central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points.

Second Experiment

In this second experiment, the main goal was to evaluate the error for different distance values. In order to accomplish that, the same data set of Section 6.2.2 was used, where 46 equally spaced positions were recorded, starting at 1 m from the speaker and ending at 5.5 m from it (see Figure 6.14). The pulse was detected using a MF and a threshold set to one hundred times lesser than the maximum amplitude of the match filter output. The absolute error is presented in Figure 6.32. It can be seen that the absolute error does not depend on the distance value. Moreover, the maximum error is less than 8 mm and in most cases the error is bellow 4 mm. This result is consistent with the information presented by the Box-and-whisker diagram of Figure 6.33. Additionally, it is important to mention that the absolute error average is around 2.85 mm. In order to reduce the error, the same interpolation technique of the previous example was implemented. The results are presented in Figure 6.34. It can be seen that the results do not change very much, nevertheless, the maximum error was reduced from 7.1 mm to 6.6 mm. These results are excellent considering that a measuring tape was used to position the microphone, which could imply that the error source could be the position technique and not the ultrasonic measurement system.

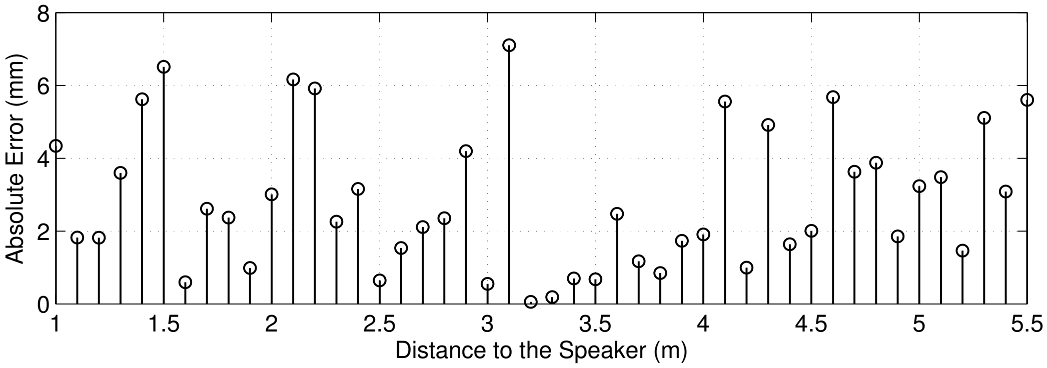


Figure 6.32: Absolute error in the distance measurement between 1 m and 5.5 m.

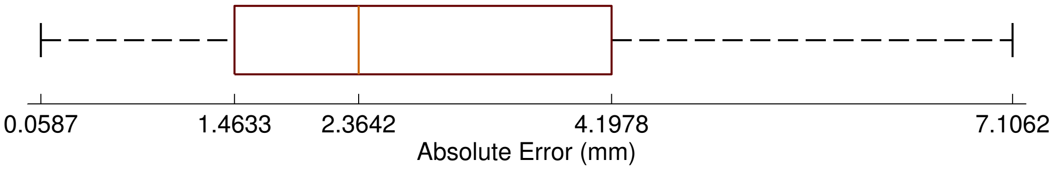


Figure 6.33: Box-and-whisker diagram of the absolute error of 46 equally spaced positions between 1 m and 5.5 m from the speaker.

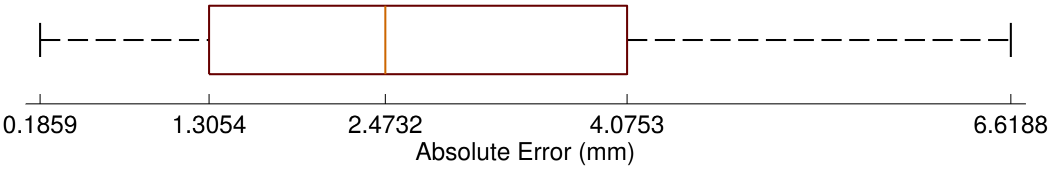


Figure 6.34: Box-and-whisker diagram of the absolute error of 46 equally spaced positions between 1 m and 5.5 m from the speaker using interpolation in the distance estimation.

6.6 Location System

To present the results for the location system the example used in Chapter 5 for the mobile location (see Section 5.8.1) will be used. The simulation environment is presented in Figure 5.32 where twenty beacons were used, twelve in the wall at 2 m from the floor and eight in the ceiling. In this example only four anchor beacons were considered two in the walls, beacons 1 and 10, and two in the ceiling, beacons 16 and 17, as presented in Figure 6.35.

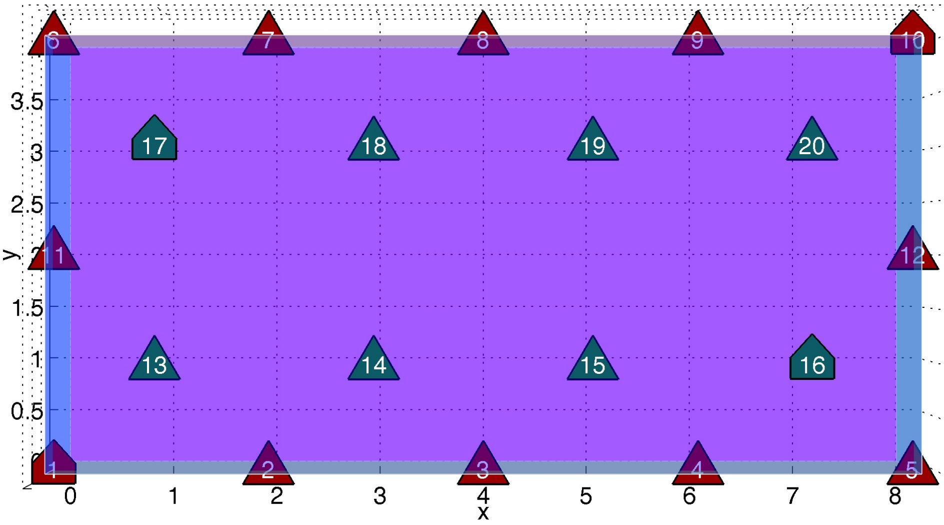


Figure 6.35: Simulation environment to test the location system.

During all the process it is considered that each beacon has a maximum range of 5 m during the synchronization process. Moreover, beacon 1 is considered to be the master beacon. Under this constraints four different levels in the hierarchy were obtained, as presented in Figure 6.36.

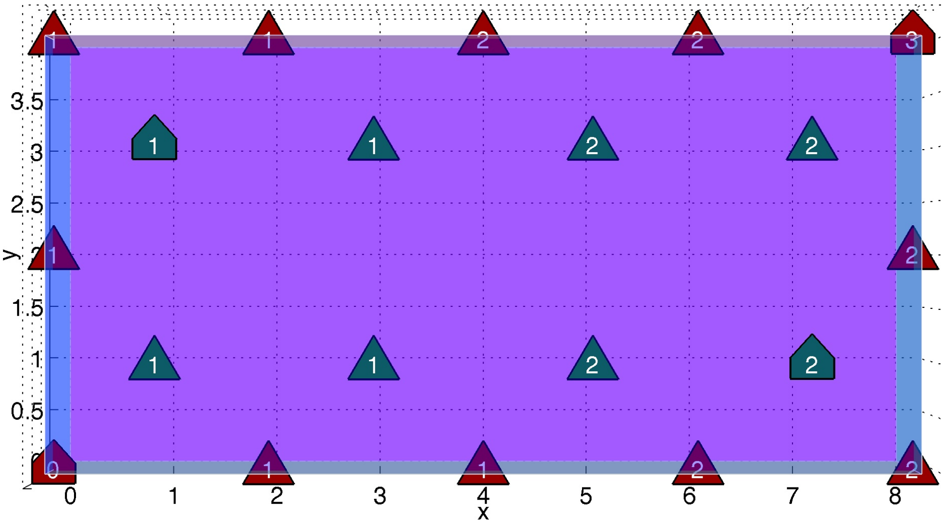


Figure 6.36: Hierarchy level of each beacon in the network considering beacon 1 as reference beacon and a maximum communication range of 5 m.

Apart from the anchor beacons, with well known location, all the remaining beacons' coordinates are initially randomly set between 0 and 100. Due to the initial slow system convergence it will be considered that the system operates in two different stages:

Initial Stage The synchronization takes place every 30 seconds and beacons do not transmit any information to the mobiles nodes;

Established Stage The synchronization takes place every 15 minutes and beacons transmit information to the mobiles nodes.

This division into two different stages allows to speed up the convergence of the beacons coordinates to their real position. After the network setup, in the *initial stage*, beacons are configured to exchange synchronization messages at a higher rate that they normally do. As a result of this, it will be impossible to send information to the mobile nodes during this stage. Nevertheless, this is not considered a problem or a disadvantage, once the information that beacons could provide is completely useless due to the error that they present. For this example the location system is configured to be in this stage for a duration of one hour. During this period of time, 120 synchronization processes are performed. After this stage, the location system changes to the *established stage* where beacons operate normally, namely, they rarely exchange synchronization messages and periodically transmit information to the mobile nodes. At this stage the synchronization process only takes place every 15 minutes.

During both stages it is considered that each beacon takes one second to perform a single iteration of the location algorithm and for each synchronization process only ten iterations are performed. This number of iterations shows, by simulation under this beacons displacement, to be the threshold value whereupon the cost function is only reduced by a small amount for each extra iteration. Figures 6.37 and 6.38 present the time-lines for the *initial stage* and the *established stage* respectively.

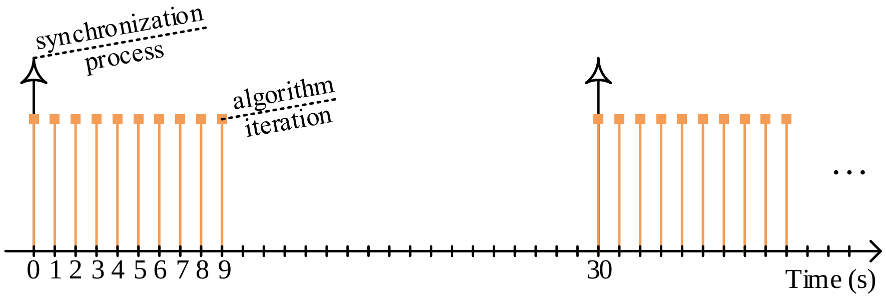


Figure 6.37: Time-line for the *initial stage*.

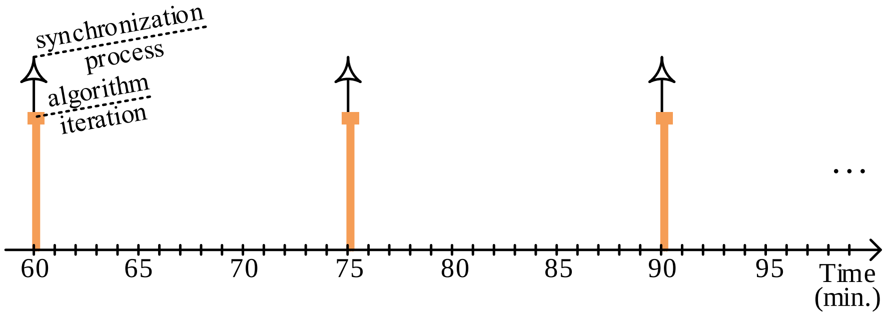


Figure 6.38: Time-line for the *established stage*.

6.6.1 Beacons Location

During the beacon location it is considered that the distance measurement presents an error produced by additive white Gaussian noise with a standard deviation of 1 cm. The average position error is depicted in Figure 6.39.

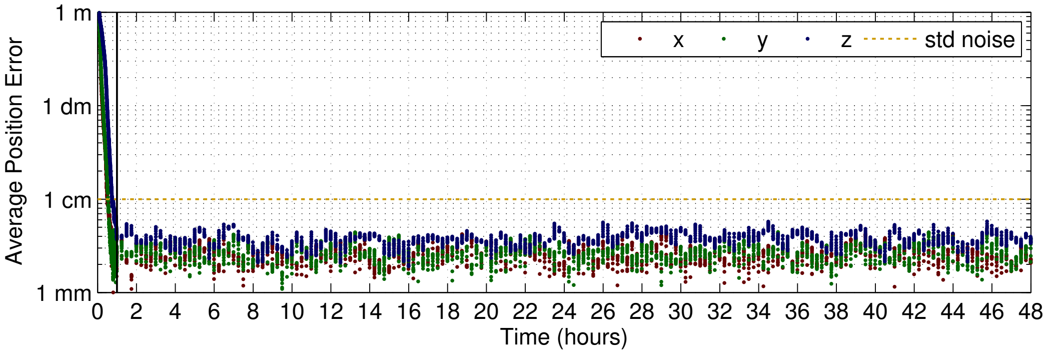


Figure 6.39: The average beacon's position error.

It can be seen that the beacon's position error is reduced to the minimum error during the first hour, namely, during the *initial stage*. Thereafter, the average error remains always between 2 and 5 mm. Additionally, the z coordinate presents larger error than the x and y coordinates, this is more or less expected due to the low diversity in the z coordinate. Nevertheless, the resultant error is not meaningful comparatively to what succeed in co-planar beacon's network, as presented in Section 5.8.2, where the error in the z coordinate is an order of magnitude above, as can be seen in the example of Figure 5.43a.

Another important conclusion that may be drawn from Figure 6.39 and especially from Figure 6.40 is that the continuous beacons relocation will not reduce the average beacon's position error. Nevertheless, this continuous relocation will be important to deal with failure and/or new beacons in the network.

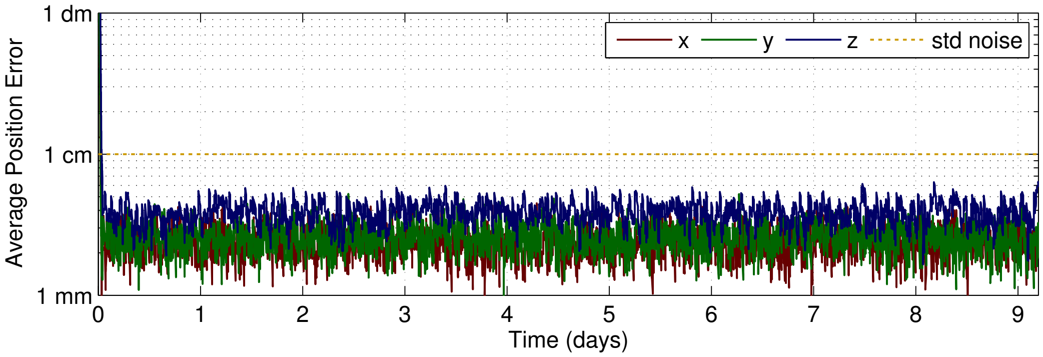


Figure 6.40: The average beacon's position error during 9 days long.

As mentioned above the beacons present a maximum communication range of 5 m, therefore, each beacon has at least two anchor beacons in the range. This reduces error propagation and keeps the error within boundaries over all the network, as it can be seen in Figure 6.41.

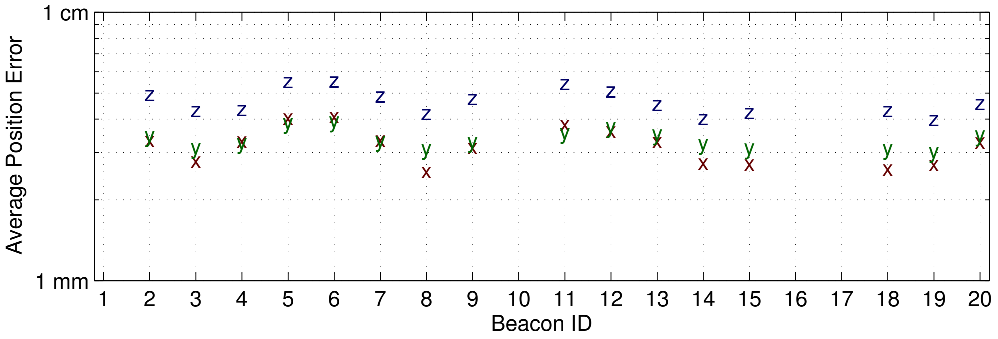


Figure 6.41: The average beacon's position error for each beacon during 9 days long.

6.6.2 Mobile Location

For the mobile location two different cases will be considered:

1. The mobile can receive all the information from the beacons network;

2. The mobile only receives information from the beacons at the 5 m range.

In both cases, it will be considered that the distance measurement presents an error produced by additive white Gaussian noise with a standard deviation of 1 cm.

Information from All Beacons

In this example it is considered that the mobile node can obtain all the information sent by all the beacons. Figure 6.43 presents the average position error, in the *x* and *z* coordinate for nine consecutive days after the network setup. The algorithm in the receiver is performed in a grid of 5 cm in the *z-plane* at 1.5 m from the floor.

The average position error in the *x* coordinate is between 4 and 8 mm in almost all the room. Although in the room corners this error increases up to 2 cm. On the other hand, the average position error in the *z* coordinate is between 1 and 4 cm increasing from the room center to the room corners, similar to what happens with the *x* coordinate. The *y* coordinate error shows a pattern very similar to that of the *x* coordinate and therefore it was decided to exclude it from this analysis.

The resultant error order may be considered acceptable for typical mobile location proposes where the most important position error belongs to the *x* and *y* coordinates.

Information from Some Beacons

In this example it is considered that the mobile node cannot obtain the information sent by all the beacons. It only receives the information from beacons in the 5 m range. Figure 6.42 shows the number of beacons that the mobile could receive information from at 1.5 m from the floor. As it can be seen, only in the room center is possible to receive information from all the 20 beacons. This number starts to reduce as the mobile moves into the room corners reaching only ten beacons.

Figure 6.44 presents the average position error, in the x and z coordinates for nine consecutive days after the network setup, similar to the previous example.

It can be observed that this range limitation does not introduce a significant impact in the average position error of the mobile node. Furthermore, at some positions this error is reduced instead of increased. For example, in the room corner, near beacon 10, the error is reduced a little bit. This effect could be explained by the fact that the mobile location algorithm considers the reference beacon as the beacon with the smallest ID. Therefore, in this example the reference beacon depends on the mobile position. For example, near beacon 10 the beacon with the smallest ID in the 5 m range is beacon 4, as a result of this the reference beacon will be beacon 4 and not beacon 1 as in the first example.

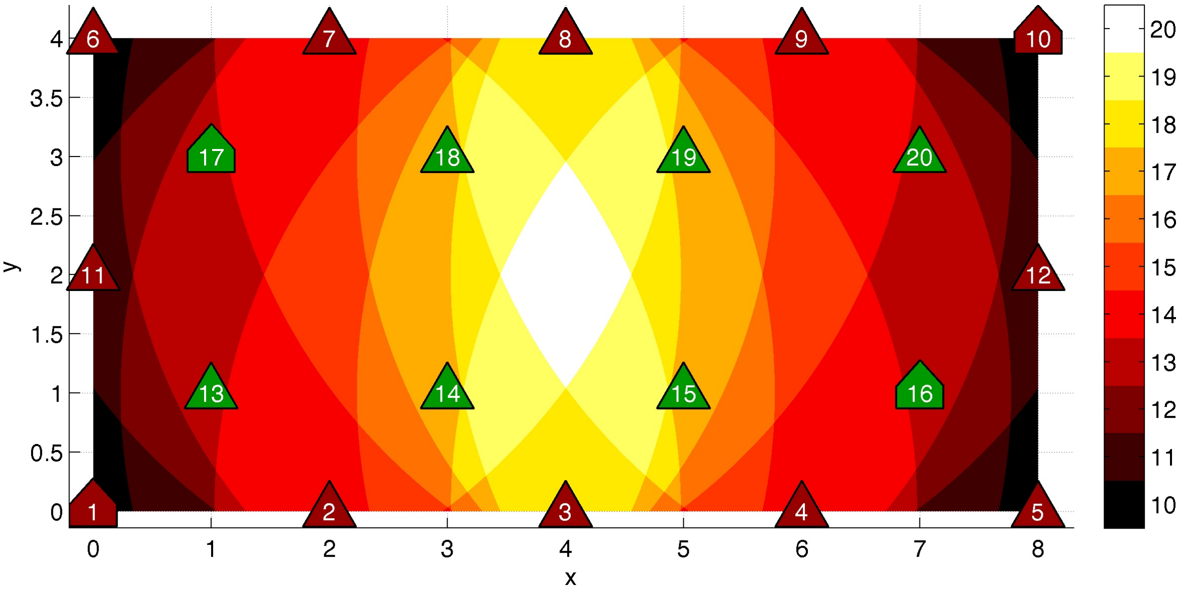
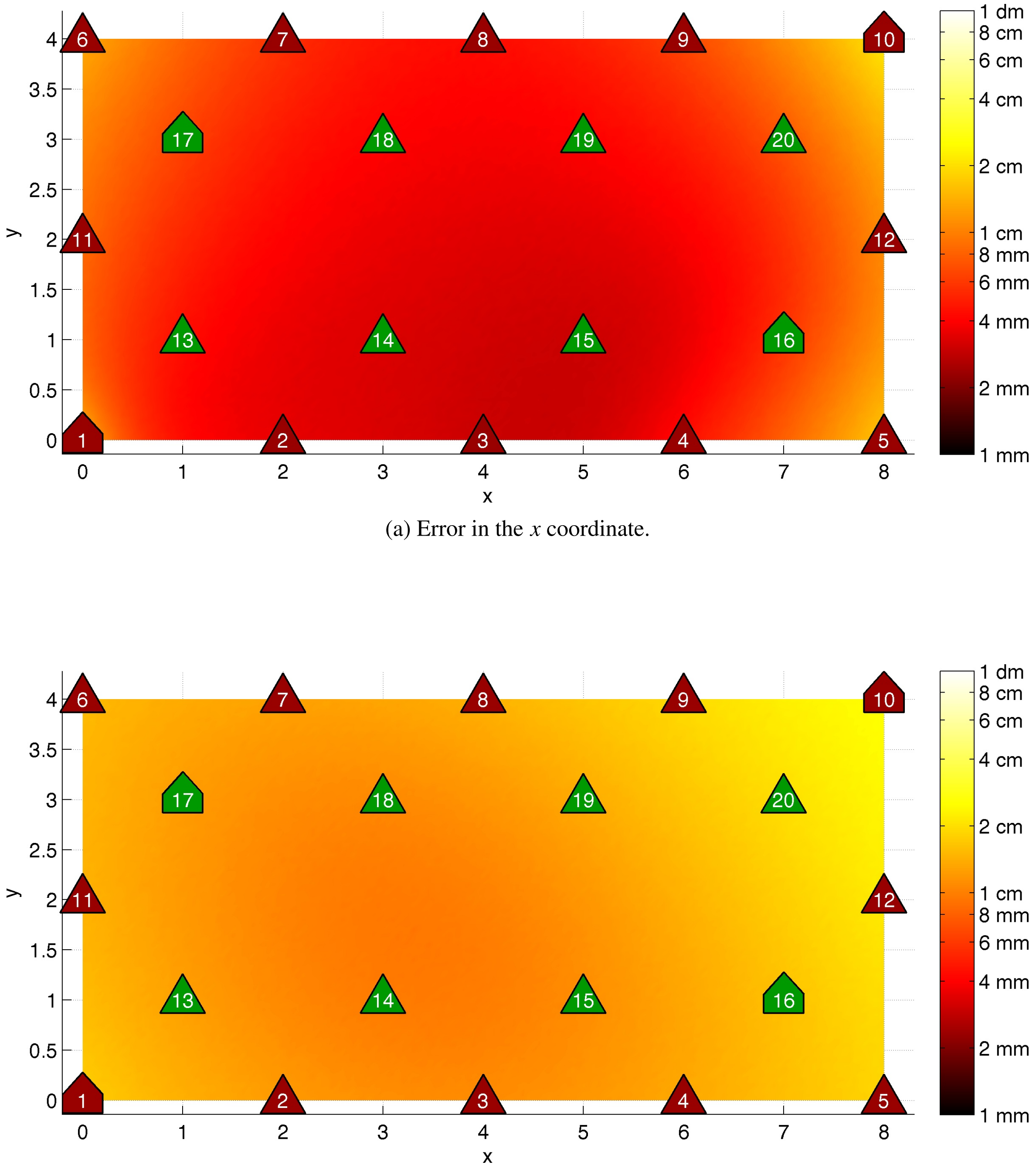


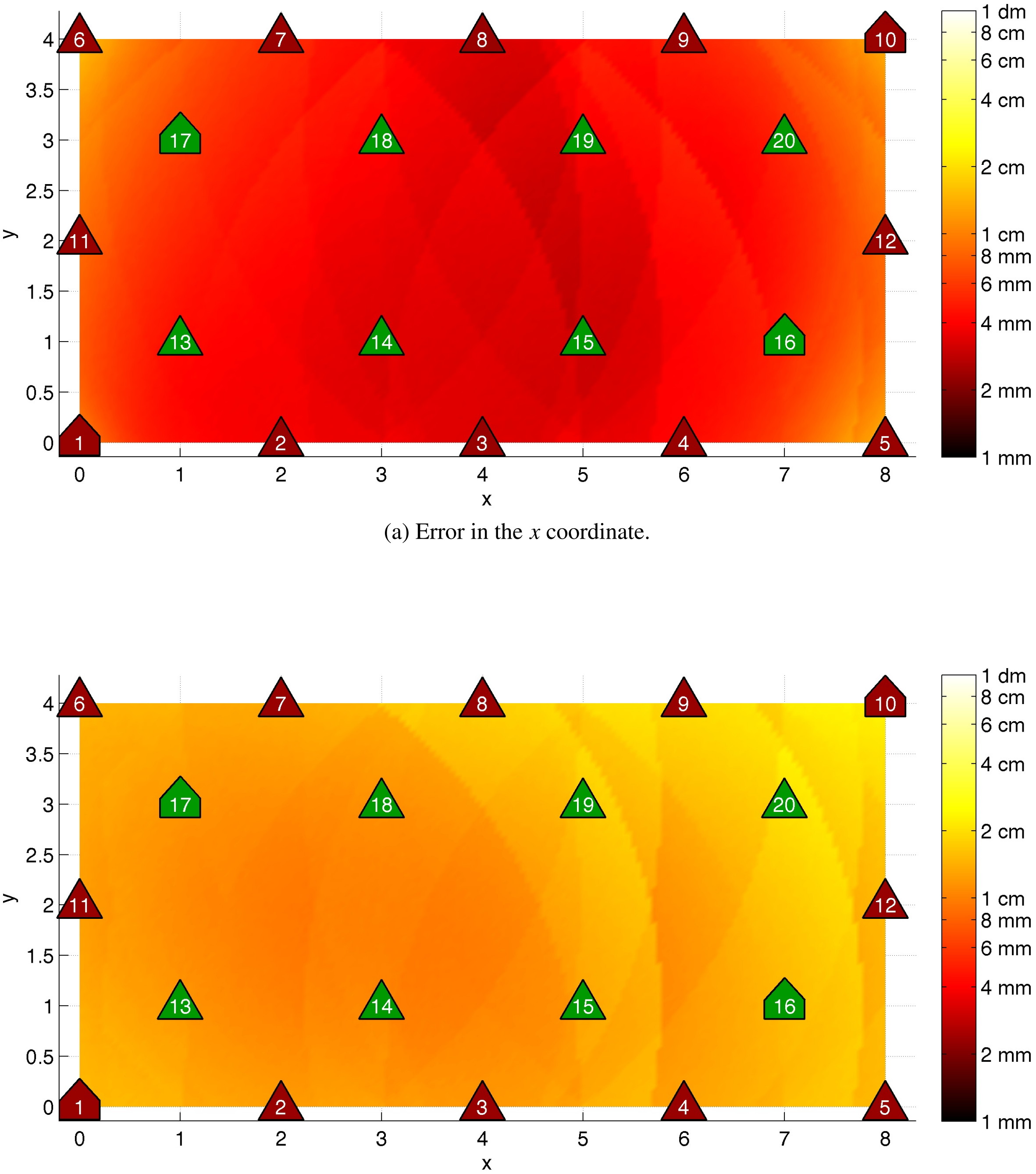
Figure 6.42: Number of beacons that a mobile could receive information from at 1.5 m from the floor, considering a 5 m range.



(a) Error in the *x* coordinate.

(b) Error in the *z* coordinate.

Figure 6.43: Average error for *x* and *z* coordinates during 9 days after the *initial stage*.



(a) Error in the *x* coordinate.

(b) Error in the *z* coordinate.

Figure 6.44: Average error for *x* and *z* coordinates during 9 days after the *initial stage* considering a maximum range of 5 m.

This page intentionally contains only this sentence.

CHAPTER 7

CONCLUSION

This dissertation describes the project and development of an indoor location system based exclusively in ultrasonic signals. A completely RF-free location system that could be implemented in any environment where RF communications are not desirable or not possible. Moreover, the proposed location system has an huge potential to be portable to other environments beyond the indoor environment. One of the best examples is the underwater environment where the RF communications are strong attenuated.

The proposed location system is very easy to setup and expand. It only needs to know the exact coordinates of a small number of beacons (anchors), and the proposed algorithm will determine the position of the remaining beacons. Even when a beacon is added or its position has changed, the same algorithm will determine the new beacon's position. Moreover, the proposed location system provides location information to the users without compromising their privacy, since the users do not need to transmit any information, resulting into a very high degree of scalability, since the location system performance does not depend on the number of users.

This chapter summarizes all the results achieved during this thesis work as well as the most important contributions made in this dissertation (Section 7.1). At the end of this chapter, in Section 7.2, some future research directions in order to improve the proposed location system will be presented and discussed.

7.1 Results Achieved

This section summarizes the most important results achieved during the thesis work. These results will be divided in three main sub-sections. In the first one we will summarize the results obtained in Chapters 3 and 6 about the ultrasonic room simulator. In the second sub-section we will summarize the results obtained in Chapters 4 and 6 about the pulse design and its advantages and performance. Finally, we will summarize the results obtained in Chapters 5 and 6 detailing all the results obtained in the proposed location system and highlighting its performance and advantages.

7.1.1 Ultrasonic Room Simulator

In this dissertation an ultrasonic room simulator was designed and implemented. It has the following characteristics:

- Designed for wideband ultrasonic signals;
- The signal bandwidth can be specified in order to improve its accuracy within that band;
- Takes into account the air attenuation and losses by providing the desired room ambient conditions, such as, temperature and humidity;

- Any type of flat surface can be implemented and its reflections can be modeled through an IR;
- Makes use of the speaker and/or microphone beam patterns;
- Simulates multiple signal sources at the same time;
- Simulates the Doppler effect for any type of movement.

On the other hand it presents the following limitations:

- Only computes the reflections up to a certain order, initially specified by the user;
- Does not take into account gradients of temperature nor wind effect;
- Discards the wave propagation through the walls (transmitted wave);
- Only simulates a single receiver at a time;
- Rounded surfaces are difficult to model.

The results presented in Chapter 6 show that the models used in the simulator are valid. The simulation presented in Section 6.2.4 shows very accurate and realist results. Additionally, the simulator compared well to another similar simulator producing similar results, however our simulator can be used in more complex rooms as well as to simulate the Doppler effect for any movement, as one can see in Section 6.3. Nevertheless, some improvements may be made, as it was pointed out in Section 3.8.

7.1.2 Pulse Design

The pulse design can be considered the core of this thesis work, since it avoids the need of an auxiliary RF channel, commonly used in ultrasonic location for small data communication (see Chapter 2). The pulse design combines a very accurate synchronization pulse with a data communication pulse robust to the indoor acoustic environment. The proposed pulse is composed by two OFDM pulses with DBPSK modulated carriers. The first pulse has optimized PAPR characteristics and has dual purpose: for measuring the time of flight and to be the phase reference for the second OFDM pulse. The second pulse carries the data.

On the other hand, the proposed OFDM pulses proved to be very robust to strong wall's reflection presenting low BER, recovering very well the sent data, where a typical modulation presents errors that could reach 40%, see the simulations presented in the end of Section 4.4.

Furthermore, the probability of detection and the probability of false alarm under additive Gaussian noise was evaluated and a simple theoretical expression was obtained. The results obtained in the field have shown that the theoretical expressions are very close to the observations as it can be seen in Section 6.4.

Finally, Section 6.5 shows that with the proposed pulse it is possible to reach a sub-millimeter precision. This result was obtained with calibrated equipment in a quite room, presenting all the potential of the proposed pulse for TOF measurement purposes.

7.1.3 Ultrasonic Location without RF Channel

Ultrasonic location without RF channel is the key point for this thesis work and it is only possible because it is possible to communicate using ultrasounds. The proposed location system is composed by two distinctive devices the beacons and the mobiles. The beacons are used as a support network allowing the mobiles to obtain their location by collecting the information sent from the beacons. One the other hand, the beacons are distributed in a network where only a few number of them know their position. By applying the proposed modified version of the distributed weighted-multidimensional scaling algorithm the remaining beacons can obtain or improve their position over time. The results have shown that, even with a co-planar network and short range communication, it is possible to obtain the beacon's position with a very small error after some messages are exchanged between beacons, except for the error related to the coordinate normal to the plane, that decreases very slowly.

This location algorithm is a distributed location algorithm that runs in each beacon and computes the beacon's position by knowing the distance to its neighbors and, if it is available, the distance between the neighbors. These distances are obtained, by employing a proposed clock synchronization mechanism that is based on the well known TPSN. The original synchronization protocol was extended allowing to measure the distance between beacons apart from the clock synchronization. This was accomplished by introducing an extra broadcast message (resulting in a total of three messages exchanged per beacon) that allows each beacon to collect and compute a significant number of distance measurements. The results showed that in a regular beacon network it is possible to obtain, not only, the distance to all the neighbors in the beacon range, but also, some distances between them. This is a good result considering that only three messages are exchanged per beacon (see Section 5.4).

In contrast to the beacon case, the mobile location is accomplish by collecting a set of periodic messages sent from the beacons network. These messages contain the beacons coordinate and the time instant at which the message was sent. With this information the mobile node computes its location by applying a TDOA algorithm. Using this approach, the mobiles do not need to send any information, resulting in two important benefits: the system is independent of the number of mobiles and it ensures privacy to the users.

The simulation results have shown that for a typical beacon displacement (in the ceiling and in the walls near the ceiling) using only four anchor beacons the remaining beacons' position can be computed with an average error around some millimeters (2 to 5 mm) for distance measurements with an average error of 1 cm. Additionally, the mobiles may determine their position with an average error in the order of some millimeters in the x and y coordinates (parallel to the ceiling plane) and in the order of some centimeters in the z coordinate (normal to the ceiling plane). Another important result, somehow expected, is that the mobile position error increases in the corners of the beacon's network. This is related with the reduced number of beacons from which it is possible to receive information. However, the results could be considered very good regarding that the distance measurements presents an average error of 1 cm.

7.2 Future Research Directions

The next important big step for the current location system is to implement it in a real environment, since, it is only tested under simulated environments. This will certainly bring new challenges and research directions. Nevertheless, there are several research directions that may be proposed to improve the current location system.

Regarding the simulator, it is important to recall the end of Chapter 3 where some future research directions were detailed. Apart from the performance and user interface, one important direction is to turn the simulator compatible with other drawing tools and room design software taking advantage of all their simplicity and user knowledge. Another important factor that could be included was the propagation model of sound waves through the walls and some of the environment conditions that were not taken into account (e.g.: room temperature gradient and wind).

Another very important research direction could be the design of a pulse that presents not only a low PAPR for synchronization but also for data communication.

In the proposed pulse design the second OFDM pulse is used only for data transmission, however, if the first pulse is correctly detected it is possible to take advantage of the timing information of the second OFDM pulse to improve the TOF estimation accuracy.

Another research topic can be the design of a single OFDM pulse with the double function of data transmission and TOF measurement.

Concerning the location algorithm, several research directions can be pointed out:

- find the best beacon arrangement that improves the clock synchronization accuracy and beacon position estimation at the same time;
- study the impact of the number and position of the anchor beacons into the position estimation of the remaining beacons;
- characterize the probability density function of the location error due to the TOF error measurements;
- study the more efficient way to share the acoustic channel between the beacons;
- study the influence of the beacons movement, beacon failures (especially for the master beacon);
- verify and improve the network security and the robustness to attacks.

The last research direction that it is important to mention is related to the portability of the location systems to underwater applications. This step will rise several challenges into the location system especially into the pulse design, without forgetting the simulator that must be redesigned to this entirely new environment.

BIBLIOGRAPHY

[1] Maria Rosália Mendes, "Os Animais," in *Enciclopédia do Conhecimento Ciencia e Tecnologia*, Mike Janson and Joyce Pope, Eds., vol. V. Resomnia, 1990.

[2] Ana Pinto, "A Época dos Descobrimentos," in *Grande História Universal*, Pedro Cardim, Ed., vol. XII. Ediclube, 2006.

[3] Mike Hazas, James Scott, and John Krumm, "Location-aware computing comes of age," *Computer*, vol. 37, no. 2, pp. 95–97, 2004.

[4] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, 2007.

[5] Simeon Lisovski, Chris M. Hewson, Raymond H. G. Klaassen, Fränzi Korner-Nievergelt, Mikkel W. Kristensen, and Steffen Hahn, "Geolocation by light: accuracy and precision affected by environmental factors," *Methods in Ecology and Evolution*, vol. 3, no. 3, pp. 603–612, 2012.

[6] Guoqiang Mao and Baris Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks*, Information Science Reference, 2009.

[7] M. J. Tucker, "Conduction signalling in the sea," *Radio and Electronic Engineer*, vol. 42, no. 10, pp. 453–456, 1972.

[8] Jose A. Costa, Neal Patwari, and Alfred O. Hero, III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, pp. 39–64, 2006.

[9] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava, "Timing-sync protocol for sensor networks," in *1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, California, USA, 2003 of Conference, pp. 138–149, ACM.

[10] Ali H. Sayed, Alireza Tarighat, and Nima Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *Signal Processing Magazine, IEEE*, vol. 22, pp. 24–40, 2005.

[11] Shashika Lakmali and Dileeka Dias, "Database Correlation for GSM Location in Outdoor & Indoor Environments," in *4th International Conference on Information and Automation for Sustainability*, 2008, pp. 42–47.

[12] Fredrik Gustafsson and Fredrik Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *Signal Processing Magazine, IEEE*, vol. 22, pp. 41–53, 2005.

[13] Roy Want, Andy Hopper, Veronica Falc, and Jonathan Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, pp. 91–102, 1992.

[14] Erwin Aitenbichler and Max Muhlhauser, "An IR local positioning system for smart items and devices," in *23rd International Conference on Distributed Computing Systems Workshops*, 2003, pp. 334–339.

[15] Andy Ward, Alan Jones, and Andy Hopper, "A new location technique for the active office," *Personal Communications, IEEE*, vol. 4, no. 5, pp. 42–47, 1997.

[16] Nissanka Bodhi Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The Cricket location-support system," in *The 6th Annual International Conference on Mobile Computing and Networking*, Boston, Massachusetts, United States, 2000, pp. 32–43, ACM Press.

[17] Nissanka Bodhi Priyantha, *The Cricket Indoor Location System*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

[18] Frédéric Rivard, Jonathan Bisson, François Michaud, and Dominic Létourneau, "Ultrasonic relative positioning for multi-robot systems," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 323–328.

[19] Wei Zhang, Joseph Djugash, and Sanjiv Singh, "Parrots: A range measuring sensor network," Tech. Rep., Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, CMU-RI-TR-06-05, 2006.

[20] Masateru Minami, Yasuhiro Fukuju, Kazuki Hirasawa, Shigeaki Yokoyama, Moriyuki Mizumachi, Hiroyuki Morikawa, and Tomonori Aoyama, "DOLPHIN: A Practical Approach for Implementing a Fully Distributed Indoor Ultrasonic Positioning System," in *Conference on Ubiquitous Computing*, Nottingham, England, 2004.

[21] José Carlos Prieto, Antonio R Jiménez, and Jorge I Guevara, "Subcentimeter-accuracy localization through broadband acoustic transducers," in *IEEE International Symposium on Intelligent Signal Processing*, 2007, pp. 1–6.

[22] Mike Hazas and Andy Hopper, "Broadband ultrasonic location systems for improved indoor positioning," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 536–547, 2006.

[23] Juan Gonzalez and Chris Bleakley, "High-Precision Robust Broadband Ultrasonic Location and Orientation Estimation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, pp. 832–844, 2009.

[24] Erich P. Stuntebeck, Shwetak N. Patel, Thomas Robertson, Matthew S. Reynolds, and Gregory D. Abowd, "Wideband powerline positioning for indoor localization," in *10th International Conference on Ubiquitous Computing*, Seoul, Korea, 2008, pp. 94–103, ACM.

[25] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," *Wireless Networks*, vol. 10, pp. 701–710, 2004.

[26] Paramvir Bahl and Venkata N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in *19th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000, vol. 2, pp. 775–784 vol.2.

[27] Thomas King, Stephan Kopf, Thomas Haenselmann, Christian Lubberger, and Wolfgang Effelsberg, “COMPASS: A probabilistic indoor positioning system based on 802.11 and digital compasses,” in *1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, Los Angeles, CA, USA, 2006, pp. 34–40, ACM.

[28] Pedro Claro and Nuno Borges Carvalho, “Local Positioning System Based on Artificial Neural Networks,” in *Artificial Neural Networks*, vol. 4669/2007, pp. 699–708. Springer Berlin / Heidelberg, 2007.

[29] Julian Randall, Oliver Amft, Jürgen Bohn, and Martin Burri, “LuxTrace: indoor positioning using building illumination,” *Personal and Ubiquitous Computing*, vol. 11, pp. 417–428, 2007.

[30] Carlos De Marziani, Jesús Ureña, A. Hernandez, Juan Jesús García, Fernando J. Álvarez, Ana Jiménez, Ma Carmen Pérez, José Manuel Villadangos Carrizo, Joaquín Aparicio, and Rómulo Alcoleas, “Simultaneous Round-Trip Time-of-Flight Measurements With Encoded Acoustic Signals,” *Sensors Journal, IEEE*, vol. 12, no. 10, pp. 2931–2940, 2012.

[31] Jens Holger Rindel, “The Use of Computer Modeling in Room Acoustics,” *Journal of Vibroengineering*, vol. 3, no. 4, pp. 41–72, 2000.

[32] Severe Holm, “Simulation of Acoustic Fields from Medical Ultrasound Transducers of Arbitrary Shape,” in *Nordic Symposium in Physical Acoustics*, Ustaoset, Norway, 1995.

[33] Sverre Holm, “Ultrasim - a toolbox for ultrasound field simulation,” in *Nordic Matlab conference*, 2001.

[34] Andrew Wabnitz, Nicolas Epain, Craig Jin, and André van Schaik, “Room acoustics simulation for multichannel microphone arrays,” in *International Symposium on Room Acoustics*, Melbourne, Australia, 2010.

[35] Peter Sonnek and Stephen V. Rice, “Synthesizing the acoustic Doppler effect in software,” in *48th Annual Southeast Regional Conference*, Oxford, Mississippi, 2010, pp. 1–5, ACM.

[36] Milica Stojanovic and James Preisig, “Underwater acoustic communication channels: Propagation models and statistical characterization,” *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 84–89, 2009.

[37] Fernando Pérez Fontán and Perfecto Mariño Espiñeira, *Modeling the Wireless Propagation Channel: A Simulation Approach with MATLAB*, vol. 5, Wiley, 2008.

[38] Lawrence E. Kinsler, Austin R. Frey, Alan B. Coppens, and James V. Sandersq, *Fundamentals of Acoustics*, John Wiley & Sons, 1982.

[39] Vesa Valimaki and Azadeh Haghparast, “Fractional Delay Filter Design Based on Truncated Lagrange Interpolation,” *IEEE Signal Processing Letters*, vol. 14, pp. 816–819, 2007.

[40] Scott Phillips, Yefim Dain, and Richard M. Lueptow, “Theory for a gas composition sensor based on acoustic properties,” *Measurement Science and Technology*, vol. 14, no. 1, pp. 70–75, 2003.

[41] ISO 9613-1, “Acoustics - Attenuation of sound during propagation outdoors - Part 1: Calculation of the Absorption of Sound By The Atmosphere,” 1993.

[42] Christopher C. Burt and Mark Stroud, *Extreme Weather: A Guide & Record Book*, W.W. Norton & Company, New York, 2007.

[43] Athanasios Papoulis, “A new algorithm in spectral analysis and band-limited extrapolation,” *Circuits and Systems*, vol. 22, no. 9, pp. 735–742, 1975.

[44] R. W. Gerchberg, “Super-resolution through Error Energy Reduction,” *Optica Acta: International Journal of Optics*, vol. 21, no. 9, pp. 709–720, 1974.

[45] Max K. Agoston, *Computer graphics and geometric modeling: implementation and algorithms*, Springer, London, 2005.

[46] Frederick Alton Everest, *Master Handbook of Acoustics*, Tab Electronics, fourth edition, 2000.

[47] Phillip John MCKerrow and Shao min Zhu, “Modelling multiple reflection paths in ultrasonic sensing,” in *International Conference on Intelligent Robots and Systems*, 1996, vol. 1, pp. 284–291.

[48] Henrik Schulze and Christian Luders, *Theory and Applications of OFDM and CDMA*, John Wiley & Sons, first edition, 2005.

[49] Simon Haykin, *Communication Systems*, John Wiley & Sons, 4th edition, 2001.

[50] John G. Proakis and Masound Salehi, *Digital Communications*, McGraw-Hill, fifth edition, 2008.

[51] Ahmad R. S. Bahai, Burton R. Saltzberg, and Mustafa Ergen, *Multi-Carrier Digital Communications: Theory and Applications of OFDM*, Springer, 2nd editio edition, 2004.

[52] Joseph Boccuzzi, *Signal Processing for Wireless Communications*, McGraw-Hill Professional, 2007.

[53] Sean Mason, Robert Anstett, Nicoletti Anicette, and Shengli Zhou, “A Broadband Underwater Acoustic Modem Implementation Using Coherent OFDM,” in *The National Conference On Undergraduate Research*, California, 2007.

[54] Yusuke Nakashima, Hosei Matsuoka, and Takeshi Yoshimura, “Evaluation and Demonstration of Acoustic OFDM,” in *14th Asilomar Conference on Signals, Systems and Computers*, 2006, pp. 1747–1751.

[55] Nadav Levanon and Eli Mozeson, *Radar Signals*, JOHN WILEY & SONS, 2004.

[56] Steven M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection theory*, Prentice Hall, Englewood Cliffs, N.J, 1998.

[57] Steven M. Kay, *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*, Prentice-Hall, Inc., 1993.

[58] Charles E. Cook and Marvin Bernfeld, *Radar Signals: An Introduction to Theory and Application* , Artech House Publishers, 1993.

[59] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing*, Pearson Prentice Hall, Upper Saddle River, N.J., 2007.

[60] Simon Haykin, *Adaptive Radar Signal Processing*, Wiley-Interscience, 2006.

[61] Seung Hee Han and Jae Hong Lee, "An overview of peak-to-average power ratio reduction techniques for multicarrier transmission," *IEEE Wireless Communications*, vol. 12, pp. 56–65, 2005.

[62] Jiang Tao and Wu Yiyan, "An Overview: Peak-to-Average Power Ratio Reduction Techniques for OFDM Signals," *Broadcasting*, vol. 54, no. 2, pp. 257–268, 2008.

[63] D. J. Newman, "An L1 Extremal Problem for Polynomials," *Proceedings of the American Mathematical Society*, vol. 16, no. 6, pp. 1287–1290, 1965.

[64] S. Narahashi and T. Nojima, "New phasing scheme of N-multiple carriers for reducing peak-to-average power ratio," *Electronics Letters*, vol. 30, no. 17, pp. 1382–1383, 1994.

[65] Edwin Van der Ouderaa, Johan Schoukens, and Jean Renneboog, "Peak factor minimization using a time-frequency domain swapping algorithm," *Instrumentation and Measurement*, vol. 37, no. 1, pp. 145–147, 1988.

[66] Meyr Heinrich, Moeneclaey Marc, and Fechtel Stefan, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, John Wiley &amp; Sons, Inc., 1997.

[67] Yik-Chung Wu, Qasim Chaudhari, and Erchin Serpedin, "Clock Synchronization of Wireless Sensor Networks," *Signal Processing Magazine, IEEE*, vol. 28, pp. 124–138, 2011.

[68] Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[69] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi, "The flooding time synchronization protocol," in *2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 2004 of Conference, pp. 39–49, ACM.

[70] Jeremy Elson, Lewis Girod, and Deborah Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147–163, 2002.

[71] Kyoung-Lae Noh, Qasim Mahmood Chaudhari, Erchin Serpedin, and Bruce W Suter, "Novel Clock Phase Offset and Skew Estimation Using Two-Way Timing Message Exchanges for Wireless Sensor Networks," *Communications, IEEE*, vol. 55, no. 4, pp. 766–777, 2007.

[72] Saurabh Ganeriwal, Ram Kumar, Sachin Adlakha, and Mani Srivastava, "Network-wide time synchronization in sensor networks," Tech. Rep., Technical report, University of California, Dept. of Electrical Engineering, 2002.

[73] Jana van Greunen and Jan Rabaey, "Lightweight time synchronization for sensor networks," in *2nd ACM international conference on Wireless sensor networks and applications*, San Diego, CA, USA, 2003, pp. 11–19, ACM.

[74] Kun Sun, Peng Ning, and Cliff Wang, "TinySeRSync: secure and resilient time synchronization in wireless sensor networks," in *13th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, 2006, pp. 264–277, ACM.

[75] Ali Burak Kulakli and Kayhan Erciyes, "Time synchronization algorithms based on Timing-sync Protocol in Wireless Sensor Networks," in *23rd International Symposium on Computer and Information Sciences*, 2008, pp. 1–5.

[76] Surendra Rahamatkar, Ajay Agarwal, Vineet Sharma, and Pankaj Gupta, "Tree Structured Time Synchronization Protocol in Wireless Sensor Network," *parameters*, vol. 12, pp. 2, 2009.

[77] Surendra Rahamatkar and Ajay Agarwal, "A Reference Based, Tree Structured Time Synchronization Approach and its Analysis in WSN," *International Journal of Ad Hoc, Sensor & Ubiquitous Computing*, vol. 2, no. 1, 2011.

[78] Fernando Seco, Antonio R. Jiménez, Carlos Prieto, Javier Roa, and Katerina Koutsou, "A survey of mathematical methods for indoor localization," in *IEEE International Symposium on Intelligent Signal Processing*, 2009, pp. 9–14.

[79] James J Caffery Jr, "A new approach to the geometry of TOA location," in *52nd Vehicular Technology Conference*, 2000, vol. 4, pp. 1943–1949 vol.4.

[80] Jasmin Desai and Uf Tureli, "Evaluating Performance of Various Localization Algorithms in Wireless and Sensor Networks," in *18th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007, pp. 1–5.

[81] Trevor F. Cox and Michael A. A. Cox, *Multidimensional Scaling*, Chapman & Hall/CRC, 2nd edition, 2001.

[82] Patrick J. F. Groenen, Rudolf Mathar, and Willem J. Heiser, "The majorization approach to multidimensional scaling for Minkowski distances," *Journal of Classification*, vol. 12, pp. 3–19, 1995.

[83] A. Andrei Dmitrievich Polianin and A. Aleksandr Vladimirovich Manzhirov, *Handbook of mathematics for engineers and scientists*, Chapman & Hall/CRC, 2007.

[84] Fernando J. Álvarez, Teodoro Aguilera, Juan A. Fernández, José A. Moreno, and Antonio Gordillo, "Analysis of the performance of an ultrasonic local positioning system based on the emission of Kasami codes," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*. 2010, pp. 1–5, IEEE.

[85] Jont B. Allen and David A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[86] Hubert W. Lilliefors, "On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.

[87] Jean Dickinson Gibbons and Subhabrata Chakraborti, *Nonparametric Statistical Inference*, Chapman & Hall/Taylor & Francis, Boca Raton, 2011.

[88] Saeed V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, Wiley, 3rd edition, 2006.

This page intentionally contains only this sentence.

Appendix A

Microphone Amplifier

The microphone amplifier was built with the single propose of connecting the ultrasonic microphone (Brüel & Kjær 4954-A) to the data acquisition system (Data Translation 9836S), as Figure A.1 shows. The microphone amplifier was designed to present a bandwidth up to 100 kHz, an adjustable gain between 10 and 10000 and very low noise. For that proposed it was decided to use two stages of amplification using two different amplifiers components. For the first amplification stage it was used a low-noise audio pre-amplifier with a fix gain and for the second stage an instrumentation amplifier with selectable gain. Accordingly to the specifications, we select, for the first stage, the THAT 1510 (from THAT Corporation) with a fixed gain of 10 and, for second stage, the INA110 (from Texas Instruments). The use of this two components allow us to design a excellent microphone amplifier with a very few extra components, reducing this way the circuit complexity and the possibility of error and noise sources. The main characteristics of those components are presented in Table A.1.

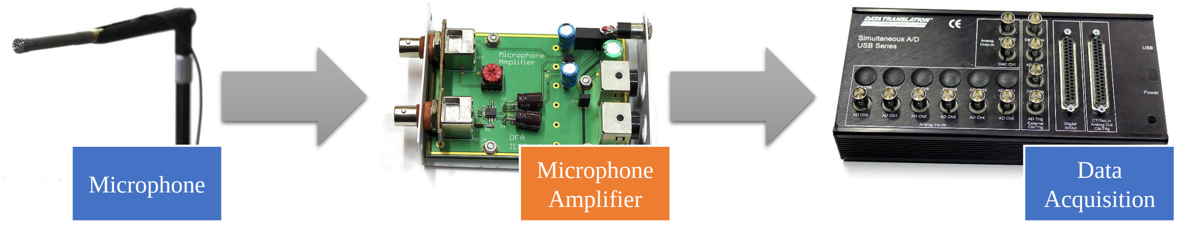


Figure A.1: Microphone amplifier propose.

	THAT 1510	INA110
Gain	10	1, 10, 100, 200, 300, 500, 600, 700 and 800
Input Noise	7 nV/ $\sqrt{\text{Hz}}$	10 nV/ $\sqrt{\text{Hz}}$
Bandwidth (-3dB)	8 MHz	100 kHz

Table A.1: Main characteristics of THAT 1510 and INA110, components used in the microphone amplifier.

The second stage gain selection was provided by a rotary switch that presents nine different positions. By combining the gain of the first stage with the possible gains of the second amplification stage it possible to select nine possible gains between 10 and 8000 as presented in Table A.2,

In order to reduce the number of components to a minimum it was decided to use the overdimensioned power adapter from the data acquisition hardware. The power adapter provides only 5 V which is not suitable for the microphone amplifier components. Therefore, it was decided to include a DC/DC converter that converts this 5 V to an isolated ±15 V. For that proposed it was used the TMA 0515D (from TRACO Electronic) which is able to provide 35 mA of current (more than enough for

	Pos. 0	Pos. 1	Pos. 2	Pos. 3	Pos. 4	Pos. 5	Pos. 6	Pos. 7	Pos. 8
Gain	10	1000	2000	3000	5000	6000	7000	8000	100

Table A.2: All possible gains provided by the microphone amplifier.

both components). This component was chosen because it does not only need any extra component, but it also has a switch frequency of around 100 kHz whereas other similar components presents a much lower frequency (for example the RD-0515D from RECOM Power presents a switch frequency around 50 kHz). Nevertheless, it was implemented a passive second order filter to remove any possible noise produced by the switch frequency. Apart from the ground shield around the components, the circuit was surrounded by the power source Earth providing an extra shield to prevent that any electromagnetic noise could interfere. The resultant circuit is depicted in Figure A.2.

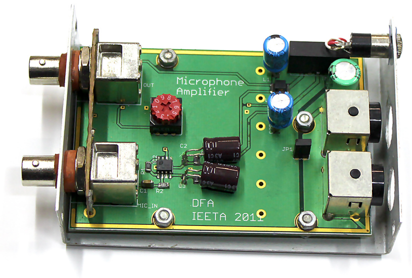


Figure A.2: Implemented microphone amplifier.

Section A.1 presents the frequency response of the microphone and sections A.2 and A.3 the schematics and the layout respectively.

A.1 Frequency Response

In order to evaluate the microphone amplifier frequency response it was used the Audio Precision – Portable one plus (see Figure A.3). During the measuring process the output signal voltage was kept between 3 and 8 V, avoiding as much as it is possible the noise and the amplitude distortion.

Figure A.3: Measuring tool used to evaluate the frequency response of the microphone amplifier.

The microphone amplifier response was measured for 41 different frequencies for each gain position and the obtained results were registered in Table A.3 and presented in Figure A.4.

F. (kHz)	Pos. 0	Pos. 1	Pos. 2	Pos. 3	Pos. 4	Pos. 5	Pos. 6	Pos. 7	Pos. 8
0.01	8.8325	875.25	1743	2624	4378	5271.4	6240	7160	87.2
3	10.98	1096.7	2183.5	3290	5491	6606.3	7824	8972	109.55
6	10.973	1097.2	2183.5	3289	5485	6595.2	7808	8948	109.45
9	10.967	1100.2	2188.5	3293	5486	6592.1	7798	8926	109.35
12	10.957	1105.5	2198	3305	5497	6600	7796	8914	109.22
15	10.95	1113.8	2212	3323	5517	6615.9	7804	8914	109.08
18	10.942	1124.8	2230.5	3349	5547	6644.4	7824	8922	108.95
21	10.935	1137.8	2255	3379	5584	6676.2	7850	8930	108.8
24	10.928	1153.2	2283	3418	5632	6722.2	7886	8952	108.65
27	10.922	1171.2	2316	3463	5687	6774.6	7932	8980	108.5
30	10.915	1191.2	2353.5	3513	5748	6833.3	7982	9014	108.33
33	10.905	1213.8	2393	3569	5817	6900	8040	9054	108.17
36	10.9	1237	2435	3627	5889	6973	8102	9094	108
39	10.89	1263	2483	3691	5968	7047.6	8162	9140	107.8
42	10.88	1289.5	2532	3759	6047	7125.4	8232	9186	107.6
45	10.87	1318	2583	3831	6129	7204.8	8302	9236	107.42
48	10.862	1347	2638.5	3904	6213	7288.9	8370	9282	107.2
51	10.852	1377.8	2694	3980	6299	7368.3	8440	9330	107
54	10.842	1410.8	2750.5	4059	6386	7452.4	8508	9374	106.8
57	10.832	1442.2	2808.5	4140	6474	7533.3	8576	9422	106.57
60	10.825	1475	2868	4220	6557	7614.3	8644	9464	106.35
63	10.815	1508.8	2928	4303	6641	7693.7	8706	9504	106.12
66	10.805	1553	2989	4383	6726	7771.4	8768	9544	105.92
69	10.795	1587.5	3045	4465	6807	7844.4	8826	9578	105.7
72	10.785	1624.5	3126	4547	6884	7914.3	8884	9610	105.5
75	10.777	1660.5	3192	4630	6959	7981	8934	9642	105.28
78	10.768	1698	3257	4710	7027	8044.4	8988	9664	105.08
81	10.757	1730	3321	4790	7099	8109.5	9030	9692	104.82
84	10.75	1768	3386	4868	7165	8166.7	9074	9712	104.62
87	10.74	1805	3453	4947	7227	8222.2	9112	9730	104.4
90	10.73	1840	3516	5023	7285	8271.4	9148	9746	104.2
93	10.722	1878	3579	5097	7339	8319	9182	9760	104
96	10.715	1915	3642	5172	7390	8363.5	9210	9768	103.8
99	10.707	1951.5	3707	5244	7437	8404.8	9238	9778	103.6
102	10.7	1988	3770	5313	7481	8442.9	9264	9784	103.45
105	10.695	2024.5	3831	5382	7522	8479.4	9288	9792	103.3
108	10.692	2060	3895	5447	7561	8511.1	9310	9796	103.17
111	10.69	2097	3950	5515	7596	8544.4	9330	9800	103.05
114	10.69	2134	4015	5579	7685	8571.4	9346	9800	102.95
117	10.69	2168.5	4071	5642	7733	8592.1	9358	9794	102.8
120	10.688	2201.5	4125	5697	7757	8607.9	9364	9660	102.65

Table A.3: Microphone amplifier gain for each gain position and frequency.

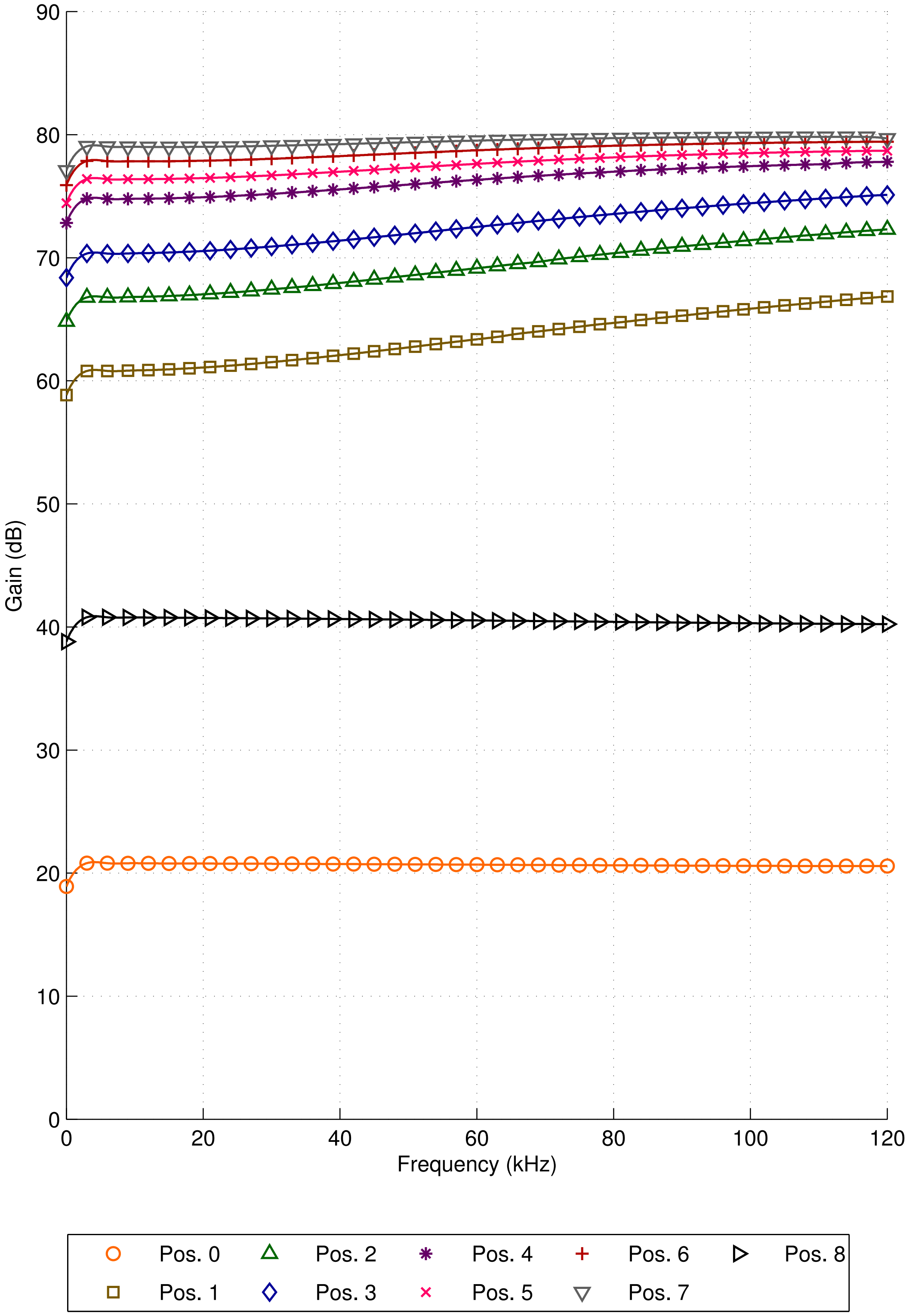


Figure A.4: Microphone amplifier frequency response for each gain position.

A.2 Schematics

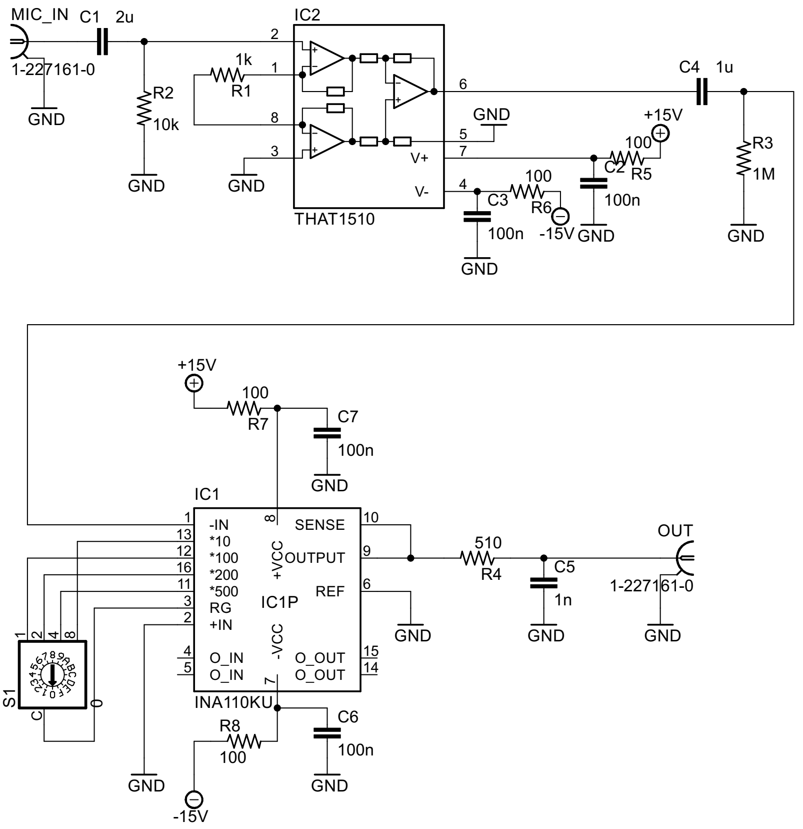


Figure A.5: Microphone Amplifier Schematic – Audio Amplifier.

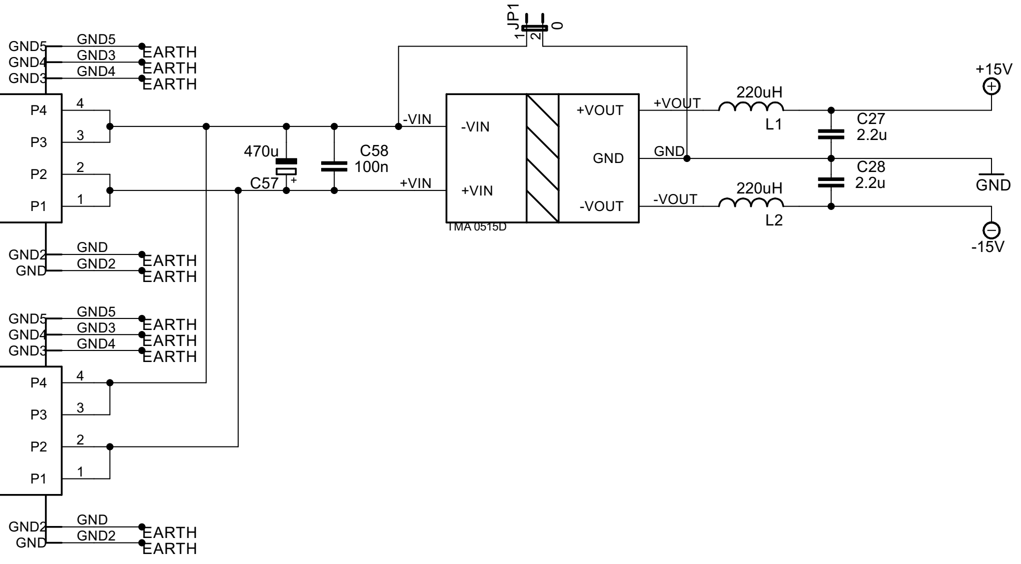


Figure A.6: Microphone Amplifier Schematic – Power Supply.

A.3 Board Layout

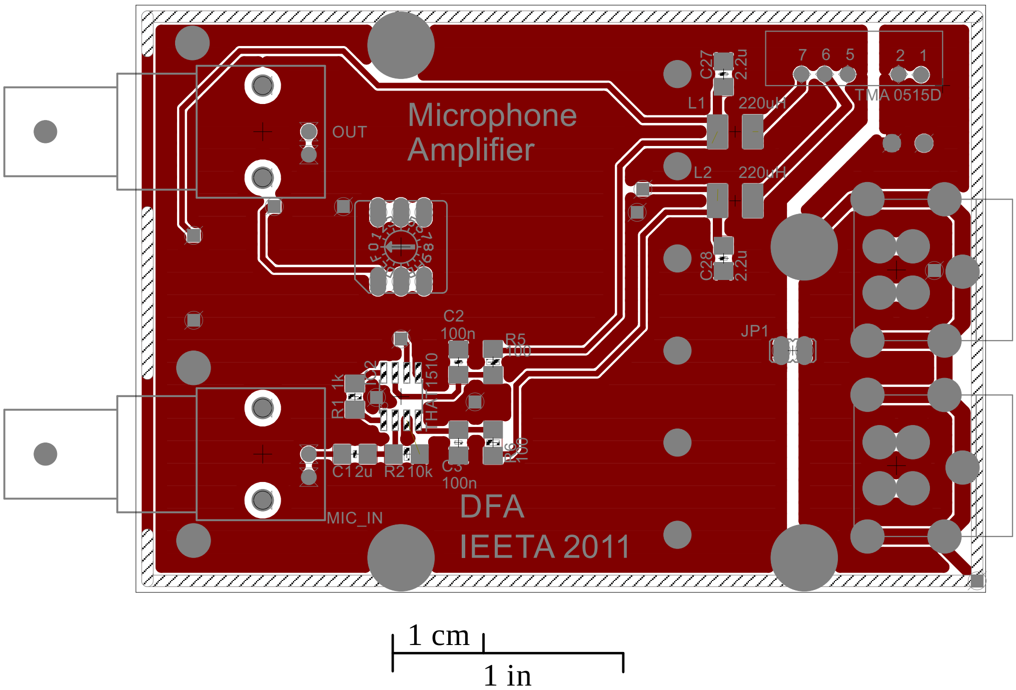


Figure A.7: Microphone Amplifier Board Layout – Top Layer.

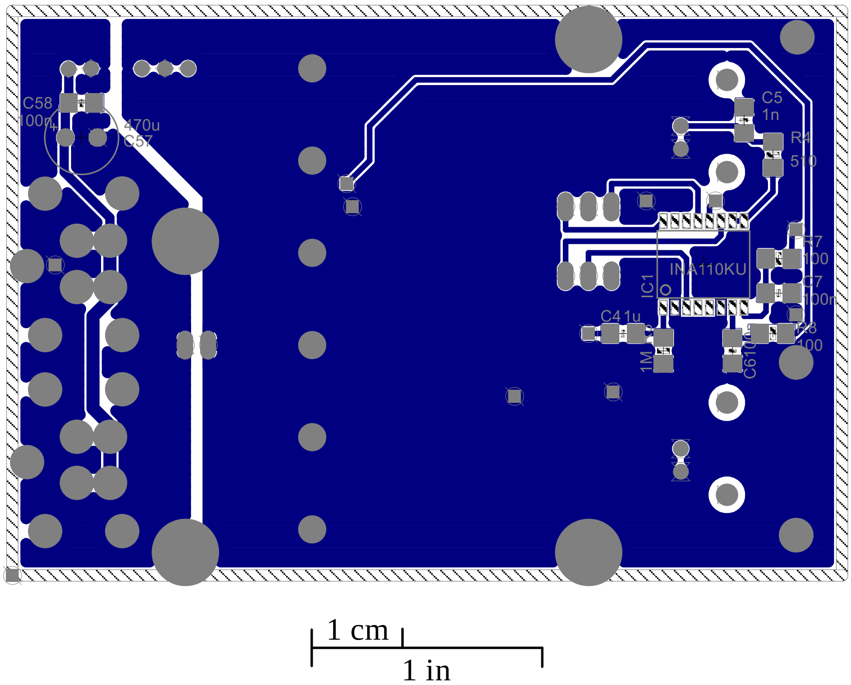


Figure A.8: Microphone Amplifier Board Layout – Bottom Layer.

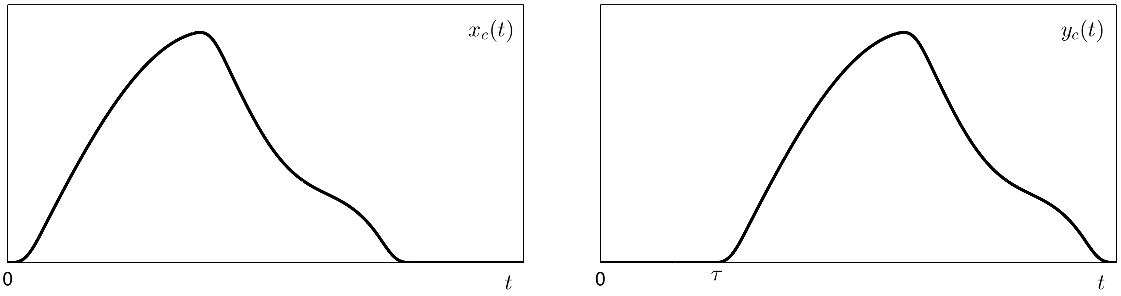
Appendix B

Fractional Delay Filters

Delaying a continuous time signal, $x_c(t)$, by an amount τ is conceptually simple. An ideal delay could be defined as a linear operator, L, defined as follow:

$$y_c(t) = L\{x_c(t)\} = x_c(t - \tau). \tag{B.1}$$

Figure B.1 presents a continuous signal time delay example, where the output signal $y_c(t)$ is a perfect delay version of the input signal $x_c(t)$.



(a) The original signal $x_c(t)$ that must be delayed.

(b) The output $y_c(t)$ of the linear operator which delays the input signal by τ.

Figure B.1: The time delay of a continuous signal.

However, the time delay of a band limited digital signal uniformly sampled is not as straight forward as the continuous signal is. It is necessary to take into account some limitations. To address this problem we start by the conversion of a continuous signal to a discrete signal by uniform time sampling:

$$t = nT, \tag{B.2}$$

where n is an integer and T is the sampling period. Using this result, equation B.1 can be rewritten to a discrete signal:

$$y(n) = L\{x(n)\} = x(n - n_\tau), \tag{B.3}$$

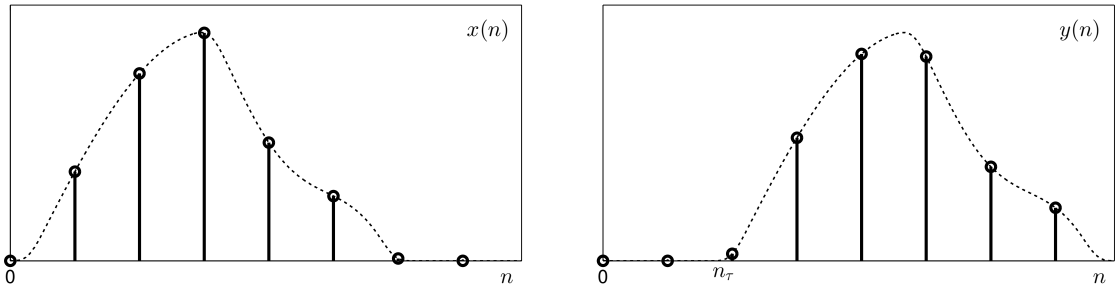
where n_τ is the delay in samples and it is a real number defined as:

$$n_\tau = \frac{\tau}{T}. \tag{B.4}$$

n_τ can be divided into the integer part and fractional part:

$$n_\tau = n_I + n_f \tag{B.5}$$

However, equation B.3 only has a meaning if n_f is zero. In this particular case $y(n)$ will be a previous sample of $x(n)$. On the other hand, if n_f is different of zero, $y(n)$ will be a sample between two previous and consecutive samples of $x(n)$, which it is impossible to obtain directly from the $x(n)$ signal. Figure B.2 presents an example of a non integer delay, and as the figure shows the samples in the delayed signal are not present in the original one.



(a) The original signal $x(n)$ that must be delayed.

(b) The $y(n)$ a delay version of the original signal by $n_\tau = 1.7$.

Figure B.2: The time delay of a discrete signal.

Nevertheless, the value for each $y(n)$ sample can be determined using bandlimited signal interpolation [88]. The problem of the fractional delay can be solved by interpolating the original signal by a factor that turns the delay integer. Note that, during this process the sampling frequency could increase considerably, especially if the delay is not a simple rational number, which brings an huge computational problem. As a result of this, in practice this procedure is not implemented, usually it is used a linear filter that produces a similar effect, resulting into a much simple implementation [39].

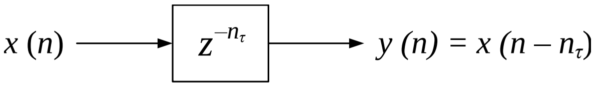


Figure B.3: Ideal delay filter.

Figure B.3 presents the ideal delay filter. As any linear and time invariant operation the delay filter can be analysed in the Z-transform domain:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-n_\tau}X(z)}{X(z)} = z^{-n_\tau} \tag{B.6}$$

where $X(z)$ and $Y(z)$ are the Z-transform of $x(n)$ and $y(n)$ respectively and H_{id} is the transfer function of the ideal delay filter. One way to deal with fractional delays is to consider the Fourier transform and the filter in the frequency domain resulting into:

$$H\left(e^{j\omega}\right) = e^{-j\omega n_\tau}. \tag{B.7}$$

This resultant filter is a linear-phase and all-pass filter with unit gain and constant group delay:

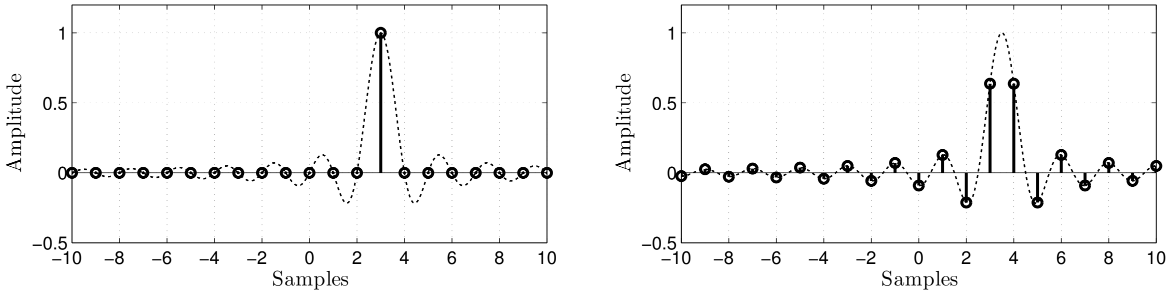
$$\left|H\left(e^{j\omega}\right)\right| \equiv 1, \qquad \forall\omega, \tag{B.8}$$

$$\tau_g\left(\omega\right) = -\frac{\partial\left(\angle\left\{H\left(e^{j\omega}\right)\right\}\right)}{\partial\omega} = -\frac{\partial\left(-\omega n_\tau\right)}{\partial\omega} = n_\tau. \tag{B.9}$$

The resultant impulse response for this filter is given by the inverse Fourier transform:

$$h(n) = \int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega = \mathrm{sinc}(n - n_\tau). \tag{B.10}$$

When the delay n_τ is an integer value, the impulse response is always zero except for $n = n_\tau$ where it is one. However, for non integer delay the impulse response will be different from zero for any n as the example of Figure B.4 shows.



(a) Delay of 3 samples.

(b) Delay of 3.5 samples.

Figure B.4: Example of an impulse response for an integer and a non integer delay.

The fractional delay results not only into an infinite impulse response but also into a non causal impulse response, which is impossible to implement in real-time applications.

Nevertheless, this ideal impulse response could be approximated by a finite impulse response. In order to obtain this filter usually the desired response is only imposed to certain frequencies or frequency band instead to be imposed to all signal bandwidth. One example of this type of filters are the Lagrange fractional delay filter presented in [39]. This filter are based on the Lagrange interpolation and they impose the ideal frequency response for the frequency $\omega = 0$. The filter coefficients for a N-order filter are obtained explicit by the following equation:

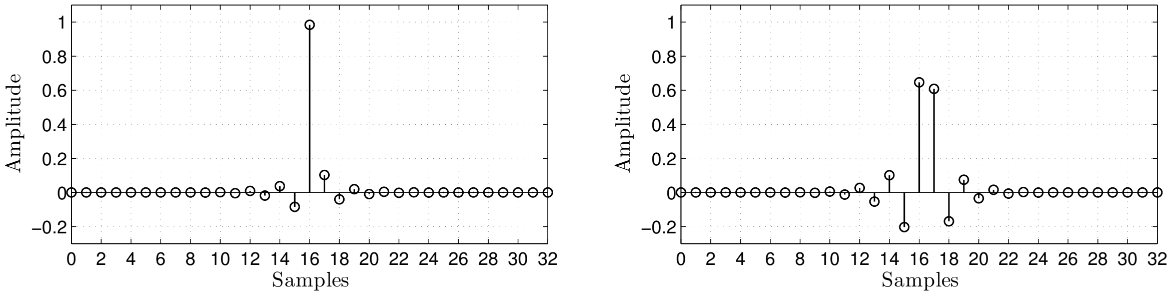
$$h(n) = \prod_{\substack{k=0 \\ k \neq n}}^{N} \frac{n_\tau - k}{n - k} \qquad \forall n_\tau \in [0,\ N] \tag{B.11}$$

Note that for $N = 1$ the filter implements a linear interpolation between the two samples, and the filter coefficients are given by:

$$h(0) = 1 - n_\tau \qquad \text{and} \qquad h(1) = n_\tau.$$

Figure B.5 presents two examples of the obtained impulse response for Lagrange fractional delay filters with 16.1 and 16.5 samples of delay using an order of 32. The correspondent frequency response and group delay is presented in Figure B.6. One can see that the frequency response and the group delay are very near to the ideal for frequencies up to 0.7π, above this frequency the filter deviates considerably from the ideal response.

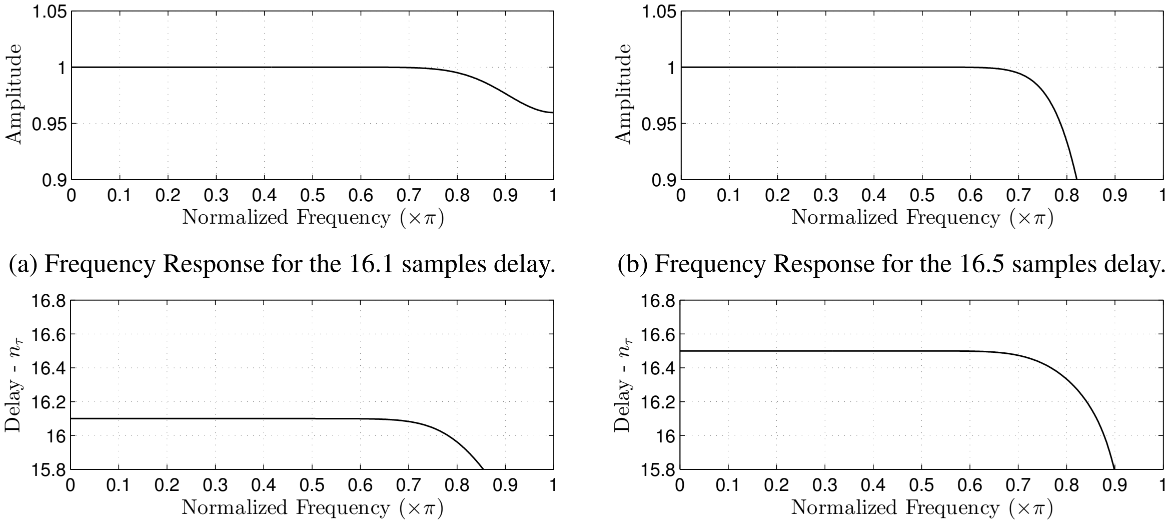
The Lagrange fractional filters only present an optimal performance for delays around half of the filter length. Figure B.7 presents an example of a delay of 0.5, as one can see not only the ideal



(a) Delay of 16.1 samples.

(b) Delay of 16.5 samples.

Figure B.5: Example of two impulse responses for Lagrange fractional delay filter with 16.1 and 16.5 samples delay and order 32.



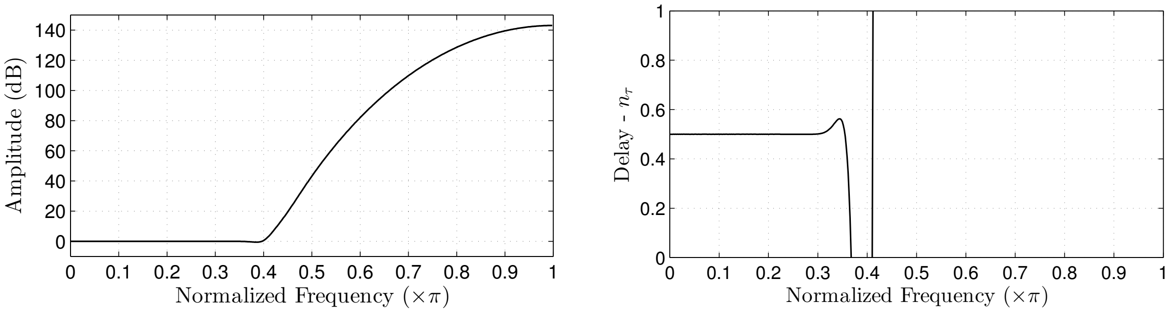
(a) Frequency Response for the 16.1 samples delay.

(b) Frequency Response for the 16.5 samples delay.

(c) Group Delay for the 16.1 samples delay.

(d) Group Delay for the 16.5 samples delay.

Figure B.6: Frequency Response and Group Delay for the examples of Figure B.5.



(a) Frequency Response.

(b) Group Delay.

Figure B.7: Frequency Response and Group Delay for a Lagrange fractional delay filter with delay of 0.5 samples and order 32.

response is narrow but also the frequency response above 0.4π presents an huge gain. This results turns the filter useless due to the fact than any noise in this frequency band will be strongly amplified. Figure B.8 presents the maximum frequency response gain as function of the delay for a 32nd order filter example. As one can see the filter only presents a maximum unite gain between 14 and 18 samples delay. Therefore, for optimal results the filter must present a delay of:

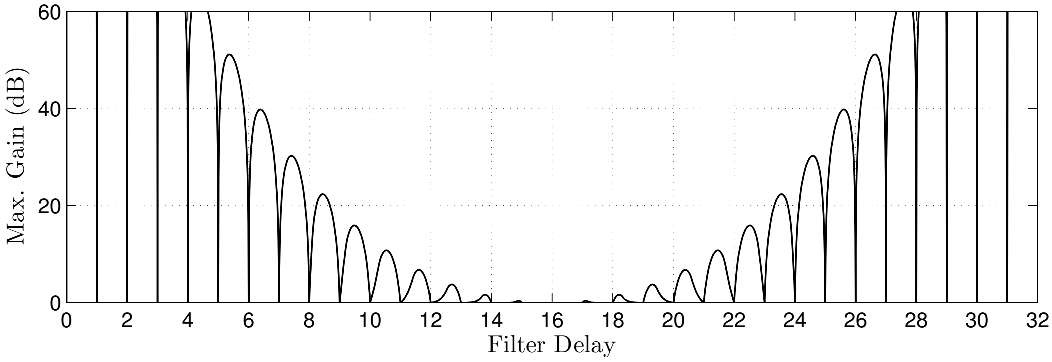


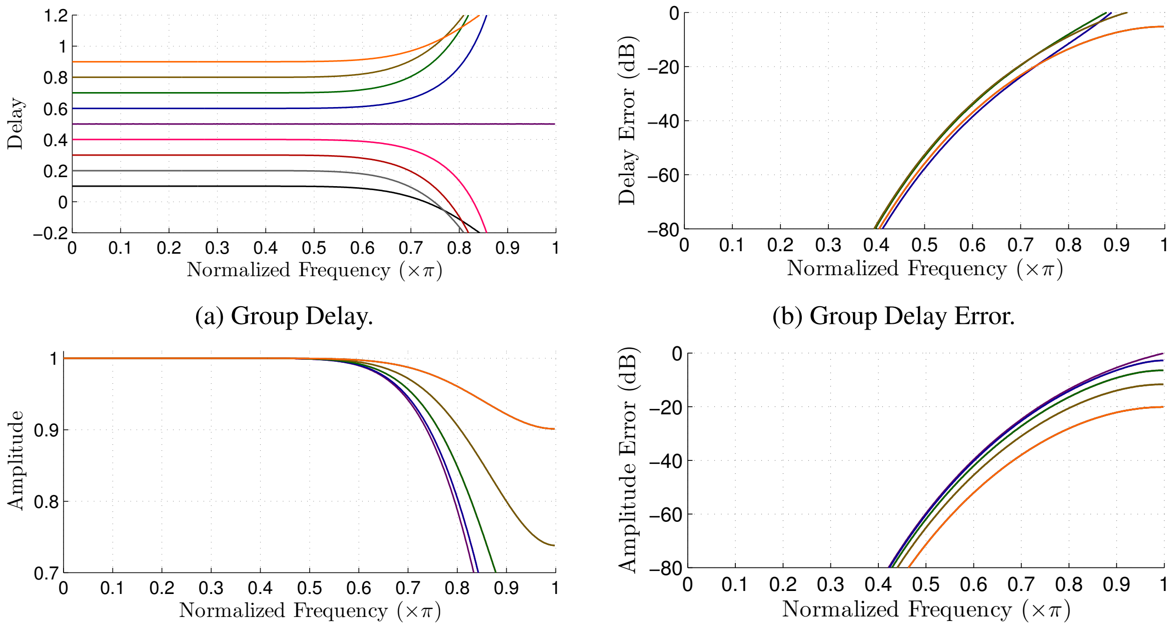
Figure B.8: Maximum frequency response gain for different delays, considering a 32 order filter.

$$n_\tau = \left\lfloor \frac{N}{2} \right\rfloor + n_f, \tag{B.12}$$

where n_f is the desired fractional delay and $\lfloor \bullet \rfloor$ represents the the nearest integers towards minus infinity. As a result of this, after applying the fractional delay filter the output must be delayed by an integer amount to obtain the desired delay. For example, if the desired delay is 45.3 samples and if it is going to be used a 32nd order filter the delay to be used in the filter must be 16.3 samples (half of the filter order plus the fractional delay part). Thereafter, the filter's output signal must be delay by 29 samples to obtain the desired delay of 45.3 samples. The main disadvantage of this process, for real-time applications, is that it imposes a lower bound to the possible delay, given by half of the filter order. Nevertheless, for most of the applications this could not be considered a meaningful problem. For example, for ultrasonic applications considering a sampling rate of 80 kHz and a 32nd order delay filter, the filter imposes a minimum delay of 200 μs which represents a minimum distance of 7 cm.

Figure B.9, B.10, B.11 and B.12 present the frequency response, the group delay and their errors for a fractional delay filter with order 15, 16, 31 and 32 respectively. By comparing the results, one can see that the odd and even order filters produce similar results. However, an odd order filter presents a slightly better delay response and slightly worst frequency response in comparison with even order filters. Additionally, an odd order filter presents an excellent delay response for the 0.5 delay case, as one can see by the purple line in Figure B.9a and Figure B.11a. On the other hand it presents similar frequency response of the remaining ones. Another important conclusion is that both errors (frequency response and group delay) are reduced by increasing the filter order.

Figure B.13 presents the impact of the fractional delay into the frequency response and the group delay performance. As one can see, the results are in agreement with those obtained in Figure B.12. The available bandwidth is maximum near the integer delays (zero and one) and decreases rapidly to around 0.6π for a fractional delay between 0.1 and 0.9. From this results, a 32nd order Lagrange



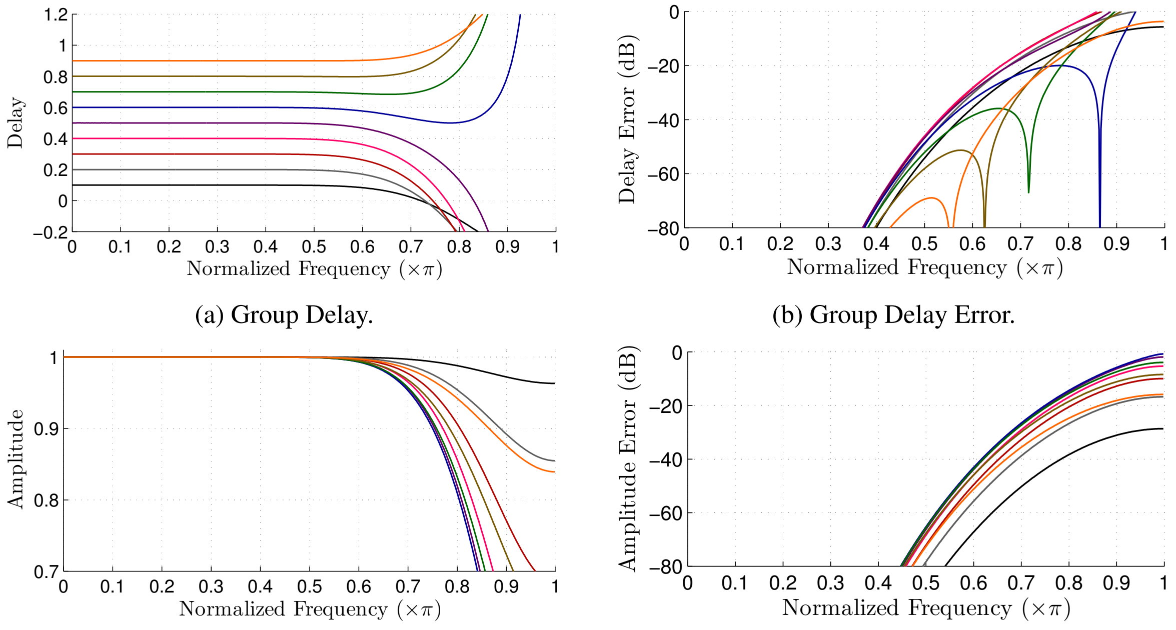
(a) Group Delay.

(b) Group Delay Error.

(c) Frequency Response.

(d) Frequency Response Error.

Figure B.9: Frequency Response and Group Delay of 15th order Lagrange fractional delay filters for different delays.



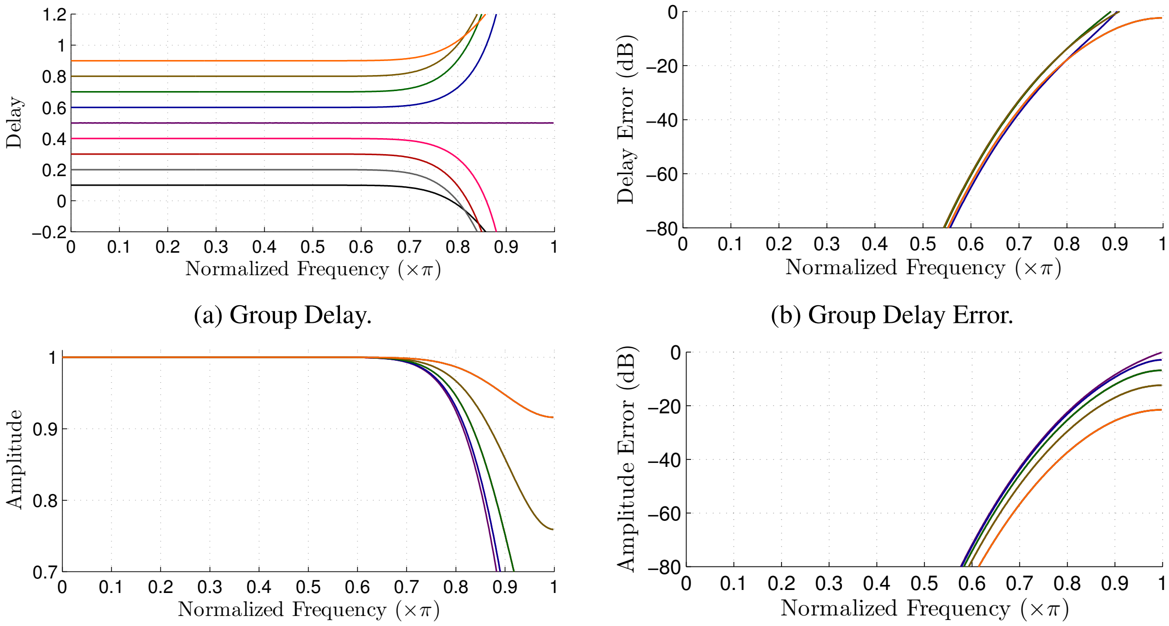
(a) Group Delay.

(b) Group Delay Error.

(c) Frequency Response.

(d) Frequency Response Error.

Figure B.10: Frequency Response and Group Delay of 16th order Lagrange fractional delay filters for different delays.



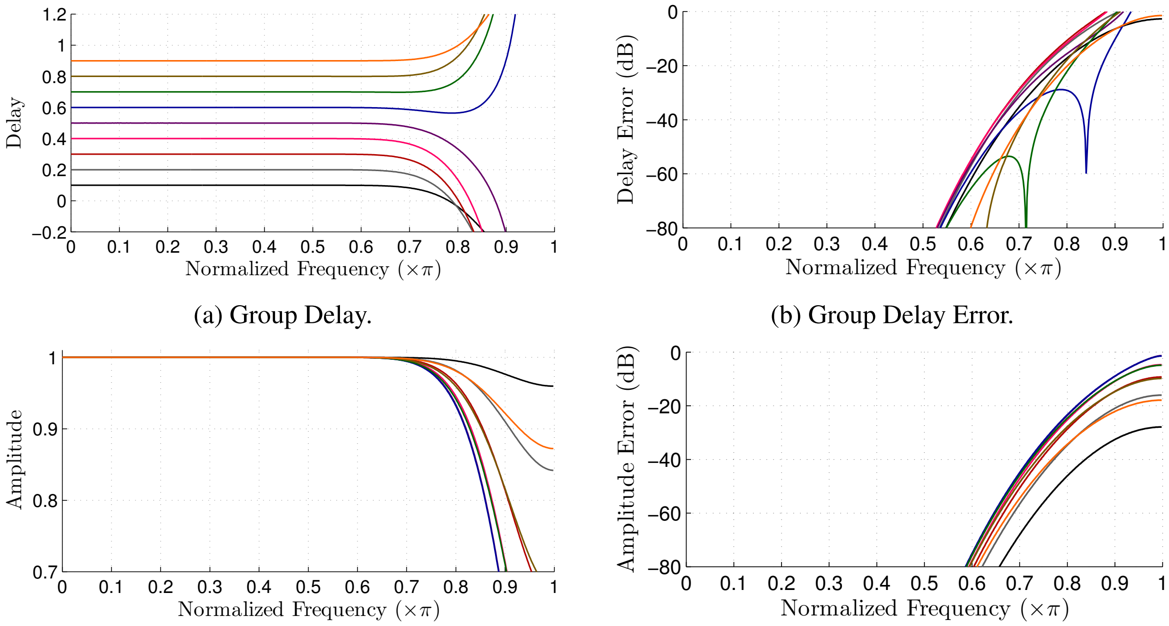
(a) Group Delay.

(b) Group Delay Error.

(c) Frequency Response.

(d) Frequency Response Error.

Figure B.11: Frequency Response and Group Delay of 31st order Lagrange fractional delay filters for different delays.



(a) Group Delay.

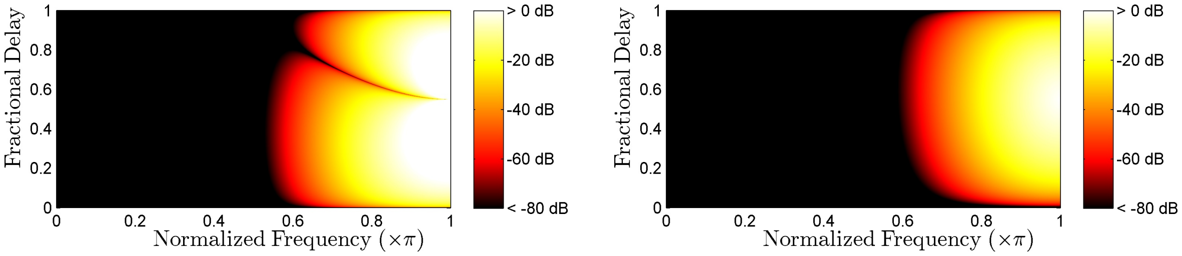
(b) Group Delay Error.

(c) Frequency Response.

(d) Frequency Response Error.

Figure B.12: Frequency Response and Group Delay of 32nd order Lagrange fractional delay filters for different delays.

fractional delay filter presents an optimum performance for a signal bandwidth of 0.6π.

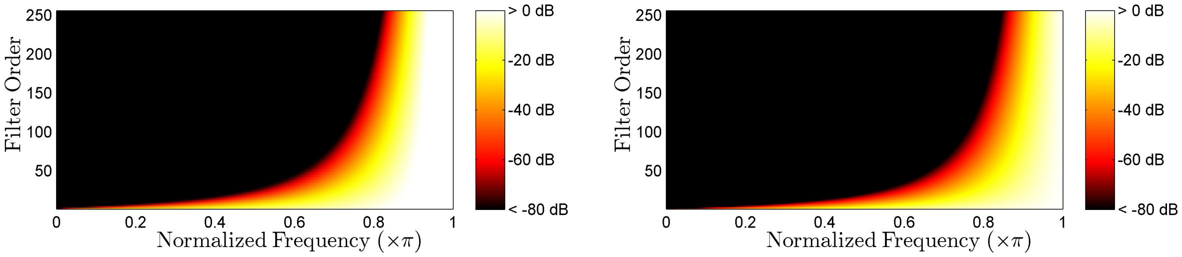


(a) Group Delay Error.

(b) Frequency Response Error.

Figure B.13: Fractional delay impact into the group delay and frequency response errors for a 32nd order filter.

Figure B.14 presents the frequency response and the group delay performance for different filter orders. As on can see in both, the frequency response and the group delay, the useful bandwidth increases with increased filter order. The filter bandwidth rapidly increases from around zero for a filter order of 1 to around 0.7π for a filter order of 70. However, for a filter order above 70 the filter bandwidth increases extremely slowly with increased filter order. As a result of this, the best trade-off between the filter performance and the filter complexity is obtained for a filter order around 70. The same results can be found in Figure B.15 where it is presented the filter bandwidth for different filter order and fractional delays considering a maximum error of -60 dB in the frequency response and group delay.



(a) Group Delay Error.

(b) Frequency Response Error.

Figure B.14: Filter order impact into the group delay and frequency response errors. The presented error is the maximum error for a fractional delay between 0 and 1.

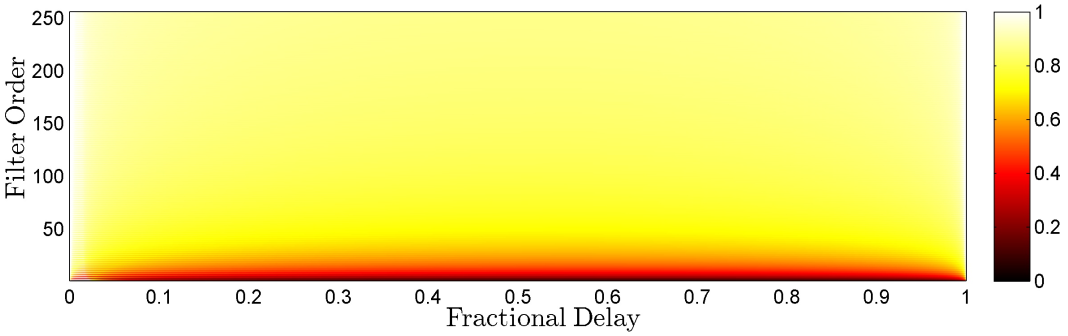


Figure B.15: Fractional delay filter bandwidth as a function of the filter order and the fractional delay.

In conclusion, the Lagrange fractional delay filters presents several advantages:

- The filter coefficients can be easily obtained by a simple formula;
- Excellent frequency response and group delay for low frequencies;
- Smooth frequency response and group delay.

Another important feature is that the frequency response is always less than the unity gain. Allowing them to be used into feedback loops.

This page intentionally contains only this sentence.