



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

ANALYSIS AND COMPARISON OF DIFFERENT INDOOR LOCATION METHODS

Alumno: Juan Parras Moral.

Tutor: Prof. Dr. Ulrich Birkel y Prof. Dr. Pedro Vera Candeas.

Depto.: Ingeniería de Telecomunicación.

Julio, 2014

ANALYSIS AND COMPARISON OF DIFFERENT INDOOR LOCATION METHODS

By

Juan Parras Moral

Presented to Technische Hochschule Mittelhessen for the Degree of:

Bachelor

Supervisor at University: Prof. Dr Ulrich Birkel

Technische Hochschule Mittelhessen

Department of Information and Communication Engineering

17th June 2014

I declare that, to the best of my knowledge, the research described herein is original except where the works of others is indicated and acknowledged.

Juan Parras Moral

Gießen, Germany

17th June 2014

List of abbreviators.....	7
Figures index	8
Tables index.....	10
Chapter 1 - INTRODUCTION	11
1.1 - BACKGROUND	11
1.2 - OBJECTIVE AND APPROACH	11
1.3 - ORGANISATION OF THE THESIS.....	12
Chapter 2 - THEORETICAL BACKGROUND	13
2.1 - INDOOR LOCATION TECHNIQUES.....	13
2.1.1 - TIME MEASUREMENTS BASED	13
2.1.2 - RSSI BASED	13
2.1.3 - PHASE BASED.....	13
2.2 - WIRELESS CHANNEL	13
2.3 - WLAN STANDARD.....	16
2.4 - SIGNAL MODEL.....	18
2.4.1 - ANGULARLY MODULATED SIGNAL	19
2.4.2 - MODULATED SIGNAL IN FREQUENCY DOMAIN	20
2.4.3 - LOCATING SCENARIO 1: DIFFERENT DISTANCES, SAME FREQUENCIES	21
2.4.4 - LOCATING SCENARIO 1: DIFFERENT DISTANCES, DIFFERENT FREQUENCIES.....	21
Chapter 3 - DESCRIPTION OF THE EXPERIMENTAL SETUPS.....	23
3.1 - DESCRIPTION OF HARDWARE AND SOFTWARE	23
3.2 - CABLE SETUP DESCRIPTION.....	25
3.2.1 - INFLUENCE OF THE CABLE PROPAGATION SPEED.....	25
3.3 - LINE-OF-SIGHT SETUP 1 DESCRIPTION	26
3.4 - LINE-OF-SIGHT SETUP 2 DESCRIPTION	27
Chapter 4 - THEORETICAL DESCRIPTIONS AND RESULTS	28
4.1 - OVERVIEW	28
4.2 - METHOD 1: FREQUENCY SWEEP	29
4.2.1 - THEORETICAL EXPLANATION.....	29
4.2.2 - ALGORITHM DESCRIPTION	31
4.2.3 - CABLE-BASED SETUP	31
4.2.3.1 - RESULTS USING A SIGNAL GENERATOR	32
4.2.3.2 - RESULTS USING A WLAN ACCESS POINT	33
4.2.3.3 - RESULTS AND CONCLUSIONS	37
4.2.4 - LOS SETUP	37

4.2.4.1 - RESULTS AND CONCLUSIONS	38
4.3 - METHOD 2: FFT	42
4.3.1 - THEORETICAL EXPLANATION.....	42
4.3.2 - ALGORITHM DESCRIPTION	43
4.3.3 - CABLE BASED SETUP.....	44
4.3.3.1 - PARAMETERS STUDY	44
4.3.3.1.1 - FFT SAMPLES NEEDED	44
4.3.3.1.2 - FFT PHASE BANDWIDTH LIMITS	46
4.3.3.2 - RESULTS AND CONCLUSIONS	47
4.3.4 - LOS SETUP	47
4.4 - METHOD 3: SIGNAL POWER.....	48
4.4.1 - THEORETICAL EXPLANATION	48
4.4.2 - ALGORITHM DESCRIPTION	49
4.4.3 - CABLE-BASED SETUP	49
4.4.3.1 - SYSTEM OFFSET	49
4.4.3.2 - RESULTS AND CONCLUSIONS	50
4.4.4 - LOS SETUP	51
4.4.4.1 - PARAMETERS STUDY	51
4.4.4.1.1 - ATENUATION CONSTANT	51
4.4.4.1.2 - SYSTEM OFFSET	53
4.4.4.2 - RESULTS AND CONCLUSIONS	53
4.5 - METHOD 4: CROSS CORRELATION	56
4.5.1 - THEORETICAL EXPLANATION.....	56
4.5.2 - INFLUENCE OF THE INTERMEDIATE FREQUENCY.....	58
4.5.3 - PROPOSED METHODS	61
4.5.4 - HIGH IF APPROACH.....	61
4.5.4.1 - ALGORITHM DESCRIPTION	61
4.5.4.2 - ALTERNATIVE CALCULUS	62
4.5.4.3 - PARAMETERS STUDY	62
4.5.4.3.1 - RESOLUTION OF THE SYSTEM	62
4.5.4.3.2 - EMPIRICAL UNAMBIGUITY RANGE	63
4.5.4.3.3 - BIAS ERROR	64
4.5.4.3.4 - SAMPLES NEEDED	65
4.5.4.4 - RESULTS AND CONCLUSIONS	66
4.5.5 - NEAR-ZERO-IF APPROACH	69

4.5.5.1 - ALGORITHM DESCRIPTION	69
4.5.5.2 - GENERAL PARAMETERS STUDY	69
4.5.5.2.1 - INTERMEDIATE FREQUENCY VALUE	69
4.5.5.2.2 - RESOLUTION	70
4.5.5.3 - BARKER-IF PARAMETERS STUDY	70
4.5.5.3.1 - SAMPLING FREQUENCY ELECTION	70
4.5.5.3.2 - CROSS CORRELATION SAMPLES USED	71
4.5.5.3.3 - ABSOLUTE VALUE	73
4.5.5.3.4 - UNAMBIGUITY RANGE	73
4.5.5.4 - SINUSOID-IF PARAMETERS STUDY	73
4.5.5.4.1 - PHASE CONSTANT OFFSET	73
4.5.5.4.2 - ENVELOPE DETECTION	73
4.5.5.5 - RESULTS AND CONCLUSIONS	77
4.5.5.5.1 - BARKER-IF METHOD	77
4.5.5.5.2 - SINUSOID-IF APPROACH	79
4.6 - COMPARATIVE	82
Chapter 5 - SUMMARY AND OUTLOOK	84
5.1 - SUMMARY OF THE WORK	84
5.2 - FUTURE WORK PROPOSALS	84
BIBLIOGRAPHY	88
APPENDIX 1: ANTENNA CALIBRATION PROCESS	90
APPENDIX 2: FFT METHOD SPEED	92
APPENDIX 3: CODE STRUCTURE EXPLANATION	93
APPENDIX 4: CODE OF FREQUENCY SWEEP APPROACH IN A LOS SETUP	95
APPENDIX 5: CODE OF FFT APPROACH, LOS SETUP	102
APPENDIX 6: CODE OF SIGNAL POWER APPROACH, LOS SETUP	109
APPENDIX 7: CODE TO ESTIMATE ATTENUATION CONSTANT, POWER SIGNAL APPROACH	115
APPENDIX 8: CODE OF CROSS-CORRELATION HIGH-IF APPROACH, LOS SETUP	120
APPENDIX 9: CODE OF CROSS-CORRELATION NEAR-ZERO-IF APPROACH, LOS SETUP	126

List of abbreviators

ADC – Analog to Digital Converter

AP – Access Point

BPSK – Binary Phase Shift Keying

CDF – Cumulative Distribution Function

DBPSK – Differential Binary Phase Shift Keying

DFT – Discrete Fourier Transform

DSB – Double Sideband

DSP – Digital Signal Processor

DSSS – Direct Sequence Spread Spectrum

FFT – Fast Fourier Transform

FHSS – Frequency Hopping Spread Spectrum

GFSK – Gaussian Frequency Shift Keying

GPS – Global Positioning System

IF – Intermediate Frequency

IQ – In-phase and Quadrature

ISM – Industrial, Scientific and Medical

LOS – Line of Sight

LTI – Linear and Time Invariant

MAC – Media Access Control

OFDM – Orthogonal Frequency Division Multiplexing

PPDU – Presentation Protocol Data Unit

RF – Radiofrequency

WiFi – Wireless Fidelity

WLAN – Wireless Local Area Network

Figures index

Figure 2.1 - ISI when Delay Spread is bigger than Symbol Period T_s	16
Figure 2.2 - Barker chips, for 1 Mbps data rate.....	17
Figure 2.3 - Block diagram of the signal model	18
Figure 3.1 - IQ data downconverter block diagram	24
Figure 3.2 - Real downconverter block diagram	24
Figure 3.3 - Schematic setup, cable.....	25
Figure 3.4 - Schematic setup, LOS	26
Figure 3.5 - Schematic setup, LOS	27
Figure 4.1 - Linear slope function.....	29
Figure 4.2 - Algorithm flow chart, frequency sweep.....	31
Figure 4.3– Phase slope obtained using a signal generator, cable setup ($d_1 = d_2 = 5m, d = 10 m$)	32
Figure 4.4 – Phase slope obtained using a signal generator, cable setup ($d_1 = 5m, d_2 = 3 m, d = 8 m$).....	33
Figure 4.5 – Phase difference obtained using a WLAN access point, WiFi channel 1, cable setup ($d = 10 m$)	34
Figure 4.6 – Phase slope obtained using a WLAN access point, mean, cable setup ($d_1 = d_2 = 5 m, d = 10 m$).....	34
Figure 4.7 – Phase slope obtained using a WLAN access point, mean, cable setup ($d_1 = 5m, d_2 = 3 m, d = 8 m$)	35
Figure 4.8 – Phase slope obtained using a WLAN access point, median, cable setup ($d_1 = d_2 = 5 m, d = 10 m$)	36
Figure 4.9 – Phase slope obtained using a WLAN access point, median, cable setup ($d_1 = 5m, d_2 = 3 m, d = 8 m$)	36
Figure 4.10 – Phase slope offset obtained, 11 WiFi Channels, with phase wrapped and unwrapped, LOS setup.....	39
Figure 4.11 – CDF of the absolute error, using 11 and 3 WiFi Channels, phase unwrapped, LOS setup	41
Figure 4.12 - Flow chart, FFT	43
Figure 4.13 – Distance estimation as a function of FFT samples used, $d = 8 m$, cable setup.....	45
Figure 4.14 – Distance estimation as a function of FFT samples used, $d = 10 m$, cable setup	45
Figure 4.15 – Phase difference in WiFi Channel 1, FFT approach , cable setup.....	46
Figure 4.16 – Attenuation constant fit obtained, LOS setup.....	52
Figure 4.17 – CDF of the absolute error, WiFi Channels 1, 6 and 11, using attenuation offset, LOS setup.....	54
Figure 4.18 – CDF of the absolute error, WiFi Channels 1, 6 and 11, without using attenuation offset, LOS setup.....	54
Figure 4.19 – Signals received as a function of sampling frequency.....	59
Figure 4.20– Signals received and cross correlation detail when $f_{if} = 100 kHz$	60
Figure 4.21– Signals received and cross correlation detail when $f_{if} = 17 MHz$	61
Figure 4.22 – Theoretical resolution as a function of WiFi Channels, high IF approach.....	63
Figure 4.23 - Distance estimation as a function of cross correlation samples used, unbiased, LOS setup, high IF approach	65

Figure 4.24 – Distance estimation as a function of cross correlation samples used, biased, LOS setup, high IF approach	66
Figure 4.25 – Real and estimated distances obtained, with and without modulus, LOS setup, high IF approach.....	67
Figure 4.26 – CDF of the absolute error, with and without modulus, LOS setup, high IF approach.....	68
Figure 4.27 - Estimated distance difference as a function of cross correlation samples and sampling frequency	72
Figure 4.28 - Signal IF spectrum, before and after decimation (N=10).....	74
Figure 4.29 - Received signals and envelope detected when $f_s = 250 \text{ MHz}$	75
Figure 4.30 - Received signals and envelope detected when $f_s = 500 \text{ MHz}$	76
Figure 4.31– CDF of the absolute error, Barker-IF approach, no limit	78
Figure 4.32– CDF of the absolute error, Barker-IF approach, limit used	78
Figure 4.33 - CDF of the absolute error, Sinusoid-IF approach	80
Figure 4.34– Real and estimated distance, Sinusoid-IF approach.....	81
Figure 5.1 - Comparison of FHSS and DSSS specifications	85
Figure 5.2 – OFDM four-subcarriers spectrum.....	86
Figure 5.3 - Filter bank transfer function	86
Figure 0.1 – Phase slope offset in WiFi antennas.....	90

Tables index

Table 2.1 - WiFi Channels frequencies	18
Table 3.1 - Default oscilloscope parameters used	24
Table 3.2 - Parameters in the cable setup.....	25
Table 4.1 - Results of frequency sweep approach, cable setup	37
Table 4.2 - Results of frequency sweep approach, LOS setup, using WiFi antennas and 11 Channels	40
Table 4.3 - Results of frequency sweep approach, LOS setup, using WiFi antennas and 3 Channels ..	40
Table 4.4 - Results of FFT approach, cable setup	47
Table 4.5 – System attenuation offsets empirically obtained, cable setup	50
Table 4.6 - Results of power approach, cable setup	50
Table 4.7- Attenuation constant offsets empirically obtained, LOS setup	52
Table 4.8 - System attenuation offsets empirically obtained, LOS setup	53
Table 4.9- Theoretical resolution as a function of WiFi Channels, high-IF approach.....	62
Table 4.10- Unambiguity range as a function of WiFi Channels, high-IF approach	64
Table 4.11 - Results of high-IF approach, LOS setup	67
Table 4.12- Results of Barker IF approach, LOS setup, $f_s = 250 \text{ MHz}$	77
Table 4.13 - Results of Barker IF approach, LOS setup, $f_s = 500 \text{ MHz}$	77
Table 4.14 - Results of sinusoid-IF approach, LOS setup.....	80
Table 4.15 - Comparison of proposed location methods.....	83

Chapter 1 - INTRODUCTION

1.1 - BACKGROUND

Not so many years ago, the mere idea of having a portable apparatus capable of locating would have been considered science fiction, but today, the research and development of some location systems has made possible that this fiction comes true. Indeed, in nowadays world, location systems are playing an important role, just to mention a few of them, IP tracking on internet based services, power level triangulation on telephony towers and the best-known location system, GPS, which stands for Global Positioning System.

This last method, based on Line-of-Sight (LOS) communications between a receiver and four or more GPS satellites [1] has spread really fast over the last years. Especially significant has been the appearance of the Smartphones, which usually integrate a GPS Receiver system and allows locating and implementing navigation systems using these devices. However, even though GPS System has proved to be enough for outdoor location in many situations, it is not able to locate indoor, because it requires LOS situations, which are not satisfied if the receiver is located inside a building.

Although a lot of research effort has been directed lately to achieve a good indoor location, none of them has already become a universally accepted standard, so research and advances are still conducted in order to find a system which could combine simplicity, high accuracy, reliability and a low price. Finding a good technique on this topic would open a new field of possibilities and new opportunities. For example, it would be interesting to navigate in big buildings, such as administrative ones, train stations or airports, and so avoid getting lost. It could be helpful as well in tunnels and mines underground. Supermarkets and malls may benefit also from this idea. And it is possible to keep on adding other applications, since the possibilities are only bound by human imagination.

1.2 - OBJECTIVE AND APPROACH

The main goal of this thesis is to adapt and test some approaches to this problem of indoor location. The secondary goal of this thesis is locating using out-of-the-shelf components, so the frequency band to be chosen is critical, as well as the standard to be used. In this case, it has been decided to analyze the approach validity in the ISM Band, since a number of affordable and widely used technologies employ this band (such as Bluetooth, ZigBee and the 802.11 standards family). Namely, the 802.11b standard will be used in order to check the validity of the approaches.

Four different locating methods will be used and tested: a frequency sweep method [8], an FFT based method, a power measure method and an IF correlation one. They all will analyze the WLAN 802.11b preamble signal in order to estimate a location in two different environments: a cable propagating one, which will be an almost ideal-conditions environment, and a LOS air propagating environment, so that the accuracy of the proposed methods in real-life environments can be estimated. The LOS setup results will be compared and the strong and weak points of each method will be highlighted.

In order to process the signals, a digital oscilloscope with two antennas will be used to capture the signals emitted by a real WLAN access point and downconvert them to the IF band. Afterwards, a

computer running MatLAB, interfaced to the oscilloscope, will receive those signals and process them in order to obtain the location estimation according to the methods mentioned before.

1.3 - ORGANISATION OF THE THESIS

Chapter 2 will give theoretical background on how other approaches to indoor location work, briefly describing the principles they are based on. Also, the characteristics of radio channels affected by multipath will be described, the WLAN 802.11b standard will be summarized, paying especial attention to the physical signal which propagates, and finally, a signal model is presented. Chapter 3 gives a description of the hardware and software used, as well as the different setups where the proposed methods were put to the test. In Chapter 4, the proposed methods are presented following the same scheme: the theoretical description of the method is given first, a brief description of the algorithm used to locate with that method is shown and in the different setups used, the parameters that have an influence on each method are studied and finally, the results are presented and commented. Chapter 5 summarizes the conclusions of the thesis, as well as presents some future work proposals on the methods studied. Finally, the MatLAB code used is contained in the appendixes.

Chapter 2 - THEORICAL BACKGROUND

2.1 - INDOOR LOCATION TECHNIQUES

Different approaches have been tested in order to achieve a good indoor location system based on RF signals. Some of them are briefly described in the following paragraphs, according to the principia they are based on.

2.1.1 - TIME MEASUREMENTS BASED

Some of the methods which use a time measurement in order to obtain a location estimation are:

- RTT based methods, which measure the Round Time Trip, the time difference between the instant in which a pulse is sent to a device and that device sends back another pulse. This method requires that the time delay in the device which echoes the pulse is known in advance [2]. Those methods have been impused by the appearance of Ultra Wide Band approaches which are based on this idea [3] [4].
- ToA (Time of Arrival) methods, based on measuring the travel time of the signal from the emitter to a receiver, requiring that both emitter and receiver are synchronized. It is also called ToF (Time of Flight). A concrete implementation is studied in [5]
- TDoA (Time Difference of Arrival), which are a set of techniques based on studying the difference of arrival of a signal to two different receptors or two signals coming from two different transmitters to a receptor. It also requires synchronization between transmitters and receivers.
- AoA (Angle difference of Arrival), is a technique which uses an array of antennas to compute the angle of arrival, like in [6]

2.1.2 - RSSI BASED

Those methods are based on the Received Signal Strength Indicator (RSSI) is a scale used to determine the amount of power that a receiver is detecting from a RF transmitter. The most important location method based on this idea is RF fingerprinting, which have been deeply studied and implemented [7] [15]. They are based on the comparison of the power level of the signals arriving to a receiver from the access points located in the vicinity and a database of previously stored power level measurements.

2.1.3 - PHASE BASED

Another set of approaches are based on phase measurements on the full ISM band, as the one described in [8]. This method is similar to TDoA, since two receivers are needed in order to locate, and it is based in the fact that the received signal phase is frequency and distance dependent, and trough measuring the signal phases in different frequencies, the location can be estimated.

2.2 - WIRELESS CHANNEL

The propagating channel is an important object of study, since it modifies the signal which propagates trough it in a statistical way. The first factor that should be taken into account is the presence of thermal noise and interference. The first one is caused by the thermal motion of the electrons [9] and it is usually modeled as Additive, White and Gaussian Noise (AWGN), a Gaussian process with zero mean, and even though this is an ideal type of noise and in some cases it is an

oversimplification, it does approximate well to systems whose frequency of operation is up to hundreds of Gigaherces; subsequently, it will be used in this work.

The interference is caused by the presence of other signals near to the band of work. It is possible to make a difference between adjacent channel interference, which is caused by signals in the vicinity of the spectrum used by the signal of interest, and co-channel interference, caused by signals within the same bandwidth that the signal of interest. This problem becomes important in public frequency bands, such as ISM, where a lot of different protocols (such as Bluetooth, ZigBee, 802.11x...) and devices share the same part of the spectrum, and so, interference is going to be a major issue. Thus, any communication protocol designed to work in this band is going to have some mechanisms oriented to minimize the impact of this issue in the communication process.

Another characteristic of the wireless channels is the presence of fading [10], which can be classified into large and small scale fading. The first one represents the average signal power attenuation or the path loss due to motion over large areas, and is related to reflection, diffraction or scattering phenomena. On the other hand, small scale fading is the dramatic change in signal amplitude or phase caused by small changes in the spatial position, and it manifests through two principal mechanisms: the signal dispersion introduced by the channel on the signals that propagate through it and the time-variant behavior of the channel. The main factors that influence small scale fading are [11] multipath propagation, the speed of the mobile or the surrounding objects and the transmission bandwidth of the signal.

One of the simplest channel model used is supposing it to be Linear and Time invariant (LTI), adding a term of AWGN. This model does not take into account small scale fading, for example. If $x(t)$ is the input signal, $y(t)$ the signal that is at the end of the channel, $h(t)$ is the channel impulse response and $n(t)$ is the expression modeling the noise, then all those signals are related using the convolution operator as follows:

$$y(t) = x(t) * h(t) + n(t) \quad (2.2.1)$$

However, in a channel where multipath is present, the channel characterization is becomes harder. The signal propagates through different paths, due to reflections and other phenomena, and subsequently, each path has a different attenuation and also a different phase, which means that in the receiver, different copies of the signal will arrive at different times and will add constructively or destructively depending on the phase. Subsequently, according to [12], the multipath channel can be modeled as:

$$y(t) = \sum_n \alpha_n(t) \cdot x[t - \tau_n(t)] + n(t) \quad (2.2.2)$$

Where n represents the n^{th} propagating path, $\alpha_n(t)$ represents the time varying gain of every single path and $\tau_n(t)$ is the delay that every path introduces. Thus, the system now could be considered linear, but time varying, and depending on the frequency response of it, there are two different type of channels: flat fading channels, whose frequency response is even, but there are strong power fluctuations in the received signal, and frequency selective fading channels, whose frequency response present phase linearity and constant gain over a bandwidth smaller than the transmitted signal, which means that the signal is distorted and that intersymbol interference (ISI) appears on the channel as well [13].

In order to compare different multipath channels, some measures are often used to characterize them [14]. The delay is usually characterized by the rms delay spread, which is defined as:

$$\sigma_{\tau} = \sqrt{\overline{\tau^2} - (\bar{\tau})^2} \quad (2.2.3)$$

Where $\bar{\tau}$ is the mean excess delay:

$$\bar{\tau} = \frac{\sum_n P(\tau_n) \cdot \tau_n}{\sum_n P(\tau_n)} \quad (2.2.4)$$

And $\overline{\tau^2}$ is defined as:

$$\overline{\tau^2} = \frac{\sum_n P(\tau_n) \cdot \tau_n^2}{\sum_n P(\tau_n)} \quad (2.2.5)$$

Another important concept used to characterize multipath channels is the Coherence Bandwidth, which is a statistical measure of the range of frequencies over which a channel can be considered to have an approximately equal gain and linear phase. The concept of frequency correlation is used here to determine the coherence bandwidth: if this value is desired to be over 90%, the coherence bandwidth is:

$$B_c = \frac{1}{50\sigma_{\tau}} \quad (2.2.6)$$

That means that the frequencies within that bandwidth will change less than 90% on average.

If the frequency correlation is desired to be over 50%, the most used expression for the coherence bandwidth is obtained:

$$B_c = \frac{1}{5\sigma_{\tau}} \quad (2.2.7)$$

If a signal presents a larger bandwidth than the coherence bandwidth of a channel and its symbol period is smaller than the delay spread of the channel, then there will be frequency selective fading present in the communication (otherwise, it would be flat fading), and subsequently, multipath related ISI will appear, as it can be seen in Figure 2.1. This means that the maximum data rate achievable will be limited because the BER will increase with the ISI.

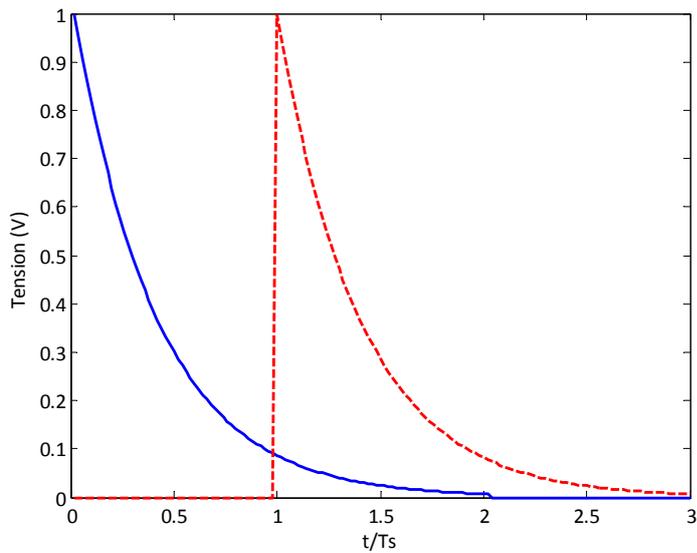


Figure 2.1 - ISI when Delay Spread is bigger than Symbol Period T_s

2.3 - WLAN STANDARD

In 1997, the IEEE released the first of their WLAN standard, and they named it 802.11. However, this original standard showed to be too slow and so, it had to be revised and updated [16]. That is the way in which the first revision was released: 802.11a. And then, more revisions and updates came: 802.11b, 802.11g, 802.11n... All of those standards are available in the IEEE website [17].

Nowadays, the IEEE 802.11x standards family is maybe the most popular set of specifications used in industrial and home applications in order to implement Wireless LAN networks. Namely, those standards only define two layers of the OSI model: the physical and the MAC ones. The fact that the technology behind those standards is today broadly extended is why this technology is ideal as a base to develop an indoor location technique, and they low cost is also a point in favor of that. Moreover, any smartphone today has a WLAN 802.11 module integrated that is compiling with the WiFi certification [16]. Even though 802.11 standards and WiFi are different terms, since the firsts are a set of specifications and the second is a certification of those standards, the use of both terms is equivalent in the practice and so they will be used in this work.

In order to locate with those standards, the signal which propagates must be studied, since the properties of this signal will be used to calculate the distance estimation. Since there are different physical layer specifications in the 802.11x standards, which use different modulation schemes (DPSK, BPSK, OFDM...), a careful election of the standard to use is very important, because depending on the chosen one the characteristics of the RF signal will vary. Since the most used standards are b, g and n, they should be used so that the highest compatibility with the market devices is achieved. And of all these standards, g and n standards have backwards compatibility with the b one, so that means that it is possible to choose the b standard and use it with transmitters compliant with the standards b, g or n.

According to the 802.11b standard, using the DSSS physical layer specification [18], there will be two possible header formats in the physical layer, which are called Long and short PPDU and whose transmission speeds are variable, and in some parts of the PDU, also, different modulation schemes could be used, like GFSK, BPSK or QPSK. However, the synchronization part of the signal is always transmitted with BPSK, using a constant data rate of 1 Mbps, and that steadiness is what makes it ideal to locate since that way it is possible to make sure about how the received signal is going to be.

This preamble is going to last between 72 and 144 μs , and it is formed by scrambled ones (long PPDU) or zeros (short PPDU), spread using an 11 chip Barker Code. This code is used because of two main reasons [19]: because of being part of the Spread-Spectrum communications techniques, it can increase the signal immunity against interferences using a higher bandwidth, and because of being a Barker Code, it will facilitate the detection of the preamble and the receiver synchronization by using a correlation block. Figure 2.2 helps explaining how the modulator block works in the case of the preamble.

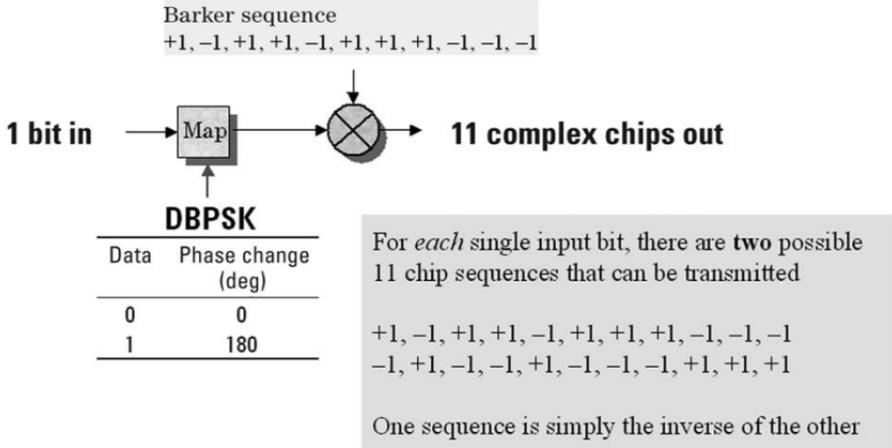


Figure 2.2 - Barker chips, for 1 Mbps data rate

The 802.11b standard uses frequencies in the ISM band (2.4 -2.5 GHz) to radiate the signals. The available spectrum is divided into channels, each one of them having different carrier frequencies. Since the distance between consecutives carrier frequencies is 5 MHz, whereas each channel bandwidth is 22 MHz, the WiFi channels are overlapped, as it can be seen in Table 2.1.

Channel number	Lower (Mhz)	frequency	Center (Mhz)	frequency	Upper (Mhz)	frequency
1	2401		2412		2423	
2	2404		2417		2428	
3	2411		2422		2433	
4	2416		2427		2438	
5	2421		2432		2443	
6	2426		2437		2448	
7	2431		2442		2453	
8	2436		2447		2458	

Channel number	Lower frequency (Mhz)	Center frequency (Mhz)	Upper frequency (Mhz)
9	2441	2452	2463
10	2451	2457	2468
11	2451	2462	2473
12	2456	2467	2478
13	2461	2472	2483
14	2473	2484	2495

Table 2.1 - WiFi Channels frequencies

Finally, not in every country are available the same channels: their number vary from the 11 channels available in North America to the 14 used in Japan, but in most of the world, 13 channels can be used. The number of channel identifies always the same carrier frequency (so, channel 1 carrier frequency is 2.412 GHz in every country). For the sake of interoperability and wide use of the results obtained, only channels 1 to 11 are going to be used in this work.

2.4 - SIGNAL MODEL

The signal to be used in this work has been already presented in the section before, as a periodical Barker Code, modulated using a DBPSK schema. This signal will be downconverted to intermediate frequency and filtered, and this will be the signal that is going to be studied and which will be used to locate. There are some different possible representations of the signal; each of them is more or less convenient depending on the signal parameter that is going to be studied. Some of those approaches will be introduced in here, and they will be used afterwards in the theoretical demonstration of each method. Finally, a description of two locating scenarios will be described.

In all cases, as it is illustrated in Figure 2.3, an RF signal is emitted by a transmitter and propagates a distance d , at a speed c , before reaching the receiver, where the signal will be multiplied by a cosine signal in order to downconvert it and low-pass filtered, in order to erase the high frequency components and obtain the IF signal.

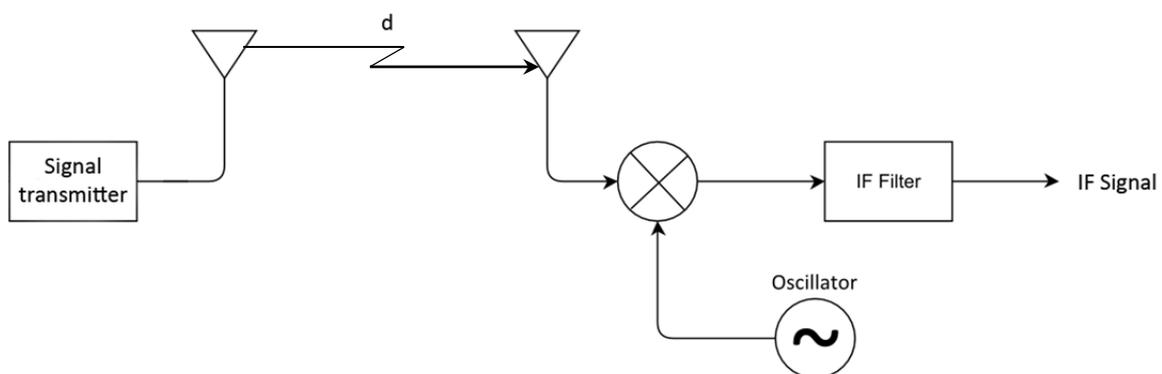


Figure 2.3 - Block diagram of the signal model

For referencing, x_{transm} will be the signal emitted by the transmitter, x_{rf} will be the signal that arrives to the receiver antenna, in other words, it is x_{transm} after propagating a distance d , x_{ol} will

be the local oscillator signal and x_{if} is going to be the signal obtained after downconverting and filtering x_{rf} .

2.4.1 - ANGULARLY MODULATED SIGNAL

The modulation used in the preamble signal studied is BPSK, which is a phase modulation. Subsequently, a signal model based on an angularly modulated signal can be used here. An angularly modulated signal on the RF band could be expressed as follows:

$$x_{transm} = A_0 \cdot \cos(2\pi f_c t + \phi(t) + \varphi) \quad (2.4.1)$$

In this equation, A_0 is the signal amplitude, f_c is the carrier frequency of the signal, $\phi(t)$ is the phase term where the modulation information goes, term that will be dependent on the phase modulation chosen and φ is a constant phase term. After propagating a distance d (at c speed), the amplitude of the signal is going to vary and a phase variation due to the propagation is going to appear, namely:

$$x_{rf} = A_1 \cdot \cos\left(2\pi f_c t + \phi(t) + \varphi + \frac{2\pi f_c}{c} d\right) \quad (2.4.2)$$

Where A_1 is the new signal amplitude, dependent on the propagated distance d , and $\frac{2\pi f_c}{c} d$ is the phase variation due to the signal propagation. This signal is downconverted to intermediate frequency band using a local oscillator signal, whose expression is:

$$x_{ol} = A_2 \cdot \cos(2\pi f_{ol} t + \varphi_2) \quad (2.4.3)$$

According to that equation, the frequency of the local oscillator is f_{ol} , its amplitude is A_2 and φ_2 is a constant phase term. If the signal received is downconverted and goes through a low pass filter, the obtained intermediate frequency signal would be:

$$\begin{aligned} x_{if} &= LPF(x_{rf} \cdot x_{ol}) = LPF\left(A_1 \cdot \cos\left(2\pi f_c t + \phi(t) + \varphi + \frac{2\pi f_c}{c} d\right) \cdot A_2 \cdot \cos(2\pi f_{ol} t + \varphi_2)\right) = \\ &= \frac{A_1 \cdot A_2}{2} \cdot \cos\left(2\pi(f_c - f_{ol})t + \phi(t) + \varphi + \frac{2\pi f_c}{c} d - \varphi_2\right) = \\ &= A \cdot \cos\left(2\pi f_{if} t + \phi(t) + \frac{2\pi f_c}{c} d + \varphi'\right) \end{aligned} \quad (2.4.4)$$

The new amplitude, which depends on the amplitude of the received signal and the local oscillator, is A , the carrier frequency is $f_{if} = f_c - f_{ol}$ and the phase term dependent on the constant phase terms of the received signal and the local oscillator would be $\varphi' = \varphi - \varphi_2$. It is possible to see, then, that in intermediate frequency band, the amplitude, phase term and carrier frequency vary, whereas the modulation scheme is maintained, since the term that contains the modulation information is the same in RF and IF.

Provided that the modulation schema used is BPSK, so $\phi(t)$ can take only two values, another possibility is representing the signal using as a pure sinusoid multiplied by a square pulse train signal $B(t)$, whose amplitude values are ± 1 , as the Barker Code used is. Thus, taking equation (2.4.1) and rearranging its terms, the radiofrequency signal obtained would be:

$$x_{rf} = B(t) \cdot \cos\left(2\pi f_c t + \frac{2\pi f_c}{c} d + \varphi_1\right) \quad (2.4.5)$$

Then this signal can be downconverted exactly as described before, so the intermediate frequency signal obtained in this case would be:

$$x_{if} = \frac{B(t) \cdot A_2}{2} \cos\left(2\pi f_{if} t + \frac{2\pi f_c}{c} d + \varphi'\right) \quad (2.4.6)$$

And if the amplitude effect of the downconverted is eliminated by normalizing with a value $A_2/2$, then the signal would be:

$$x_{if} = B(t) \cos\left(2\pi f_{if} t + \frac{2\pi f_c}{c} d + \varphi'\right) \quad (2.4.7)$$

2.4.2 - MODULATED SIGNAL IN FREQUENCY DOMAIN

A second option is studying the signal from the frequency point of view. Since the Fourier transform allows to do so, a signal can be studied either in time domain (as it has been done before) or in frequency domain. Namely, the relation between both domains is the one following:

$$X(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{j2\pi f t} dt \quad (2.4.8)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(f) \cdot e^{-j2\pi f t} df \quad (2.4.9)$$

The modulation property of the Fourier Transform says that the Fourier transform of the product of two signals is the convolution of their Fourier transform, namely:

$$F(y(t) \cdot z(t)) = Y(f) * Z(f) \quad (2.4.10)$$

Every time that a signal enters a frequency mixer, it is multiplied by a pure sinusoid; let it be a cosine, whose Fourier transform is:

$$F(\cos(2\pi f_0 t)) = \frac{1}{2} [\delta(f - f_0) + \delta(f + f_0)] \quad (2.4.11)$$

The Fourier transform of the sinusoid signals are Dirac delta functions, and when any signal is convolved with any of those, a translation is obtained. Subsequently, applying the properties of the Dirac delta convolutions, the IF signals in the frequency domain would be:

$$F(x(t) \cdot \cos(2\pi f_0 t)) = X(f) * \left(\frac{1}{2} [\delta(f - f_0) + \delta(f + f_0)]\right) = \frac{1}{2}(X(f - f_0) + X(f + f_0)) \quad (2.4.12)$$

That means, that the spectrum of the analyzed signal will always be the same, independently that it is analyzed in RF, IF or baseband, because this process just translates it, but does not make any further modifications in the signal received, at least in ideal conditions. The Fourier Transform of the signal in intermediate frequency will have values in the interval $[f_{if} - B/2, f_{if} + B/2]$, where B is the signal bandwidth.

Another possible way of representation would be using phasors. They are pure sinusoid waves, thus they only contain information about one frequency, but the frequency response of an LTI system is determined as the sum of the single frequency responses of all the phasors that constitute the input

signal. In other words, the computation of the response for a single frequency is easily generalized for all the frequencies of the input signal. Thus, the signal model in this case could be a single phasor, whose equation can be expressed as follows:

$$x_{transm} = A_0 \cdot \cos(2\pi f_c t + \beta) = Re \{A_0 \cdot e^{j(2\pi f_c t + \beta)}\} \quad (2.4.13)$$

$$x_{rf} = A_1 \cdot \cos\left(2\pi f_c t + \beta + \frac{2\pi f_c}{c} d\right) = Re \left\{A_1 \cdot e^{j(2\pi f_c t + \beta + \frac{2\pi f_c}{c} d)}\right\} \quad (2.4.14)$$

Where A_0 is the transmitted signal amplitude, A_1 is the received signal amplitude, f_c is the carrier frequency and β is the signal constant phase term.

2.4.3 - LOCATING SCENARIO 1: DIFFERENT DISTANCES, SAME FREQUENCIES

Now, at least two possible scenarios arise in order to locate using phase measurements, if a signal coming from a transmitter is processed by two receivers according to Figure 2.3, each one of them at a distance d_1 and d_2 of the signal emitter. In the first scenario, only one carrier frequency is used to obtain the location estimation, thus, according to equation (2.4.4), the two IF signals could be expressed as:

$$x_{if,1} = A \cdot \cos(2\pi f_{if} t + \phi(t) + \frac{2\pi f_c}{c} d_1 + \phi') \quad (2.4.15)$$

$$x_{if,2} = A \cdot \cos(2\pi f_{if} t + \phi(t) + \frac{2\pi f_c}{c} d_2 + \phi') \quad (2.4.16)$$

If the phase of both signals is subtracted, the distance difference can be estimated as:

$$\varphi_1(t) - \varphi_2(t) = \frac{2\pi f_c}{c} (d_1 - d_2) = \frac{2\pi f_c}{c} \Delta d \quad (2.4.17)$$

However, the maximum variation of distance is limited by the periodic phase. The measure of the distance is unique or unambiguous when the phase variation is limited to 2π , and thus, the maximum distance difference will be:

$$\Delta d_{max} = \frac{2\pi c}{2\pi f_c} = \frac{c}{f_c} \quad (2.4.18)$$

For example, in a medium whose propagation speed is $c = 2 \cdot 10^8 m/s$ and the carrier frequency is $f_c = 2.4 GHz$, the maximum difference distance to be detected without ambiguity would be $\Delta d_{max} \approx 8.3 cm$.

Thus, locating using phase measurements in this scenario is going to be strongly limited by the carrier frequency.

2.4.4 - LOCATING SCENARIO 1: DIFFERENT DISTANCES, DIFFERENT FREQUENCIES

In this scenario, two carrier frequencies will be used to locate. In the same scenario as in the section before, now two different carrier frequencies will be sent from the transmitter, $f_{c,a}$ and $f_{c,b}$, to two receivers placed at a distance d_1 and d_2 of the signal emitter. According to equation (2.4.4), the IF signals could be expressed as follows:

$$x_{if,1,a} = A \cdot \cos(2\pi(f_{c,a} - f_{ol})t + \phi(t) + \frac{2\pi f_c}{c} d_1 + \phi') \quad (2.4.19)$$

$$x_{if,1,b} = A \cdot \cos(2\pi(f_{c,b} - f_{ol})t + \phi(t) + \frac{2\pi f_c}{c} d_1 + \varphi') \quad (2.4.20)$$

$$x_{if,2,a} = A \cdot \cos(2\pi(f_{c,a} - f_{ol})t + \phi(t) + \frac{2\pi f_c}{c} d_2 + \varphi') \quad (2.4.21)$$

$$x_{if,2,b} = A \cdot \cos(2\pi(f_{c,b} - f_{ol})t + \phi(t) + \frac{2\pi f_c}{c} d_2 + \varphi') \quad (2.4.22)$$

Therefore, the phase variation for signals with the same carrier frequency would be:

$$\varphi_{1,a}(t) - \varphi_{2,a}(t) = \frac{2\pi f_{c,a}}{c} (d_1 - d_2) = \frac{2\pi f_{c,a}}{c} \Delta d \quad (2.4.23)$$

$$\varphi_{1,b}(t) - \varphi_{2,b}(t) = \frac{2\pi f_{c,b}}{c} (d_1 - d_2) = \frac{2\pi f_{c,b}}{c} \Delta d \quad (2.4.24)$$

Thus, computing the phase difference between different carrier frequencies, the distance difference is obtained as:

$$\left(\varphi_{1,a}(t) - \varphi_{2,a}(t) \right) - \left(\varphi_{1,b}(t) - \varphi_{2,b}(t) \right) = \frac{2\pi}{c} (f_{c,b} - f_{c,a}) \Delta d \quad (2.4.25)$$

Again, the maximum variation of distance is limited by the periodic phase. Limiting the phase variation to 2π , the maximum distance difference is:

$$\Delta d_{max} = \frac{2\pi c}{2\pi(f_{c,b} - f_{c,a})} = \frac{c}{(f_{c,b} - f_{c,a})} \quad (2.4.26)$$

For example, in a medium whose propagation speed is $c = 2 \cdot 10^8 m/s$ with carrier frequencies of $f_{c,a} = 2.412 GHz$ and $f_{c,b} = 2.417 GHz$, the maximum difference distance without ambiguity would be $\Delta d_{max} \approx 40m$.

As a conclusion, in order to locate using phase-related measurements, it is highly recommendable using different frequencies instead of only one, since that way the unambiguity range becomes dependent on the carrier frequencies interval instead of depending on the carrier frequency itself. That means that the unambiguity range could be adjusted in the second case, whereas in the first one that will not be possible if using a standard protocol, as WLAN 802.11.

Chapter 3 - DESCRIPTION OF THE EXPERIMENTAL SETUPS

3.1 - DESCRIPTION OF HARDWARE AND SOFTWARE

The approaches which are going to be described in the next chapters will be tested in different setups in order to check their characteristics and the location accuracy that can be obtained from each of them. The components used and their characteristics are summarized here:

1. Signal generator. It is going to be used to generate a pure sinusoid wave in different frequencies. Even though the signals that are going to be tested are the ones defined in the 802.11b standard, it is useful to use a signal generator as a first step in some approaches, so that the error detection becomes simpler and faster. In the setups, the carrier frequency used was between 2.4 and 2.45 MHz, using a 5 MHz increment, and the wave type was a pure sinusoid without any modulation.
2. WLAN 802.11b compliant access point, whose concrete model and datasheet can be accessed from [23]. Antennas or coaxial cables can be easily connected to this AP, what makes it ideal in order to test the proposed methods in two different environments as they are. Also, the configuration parameters of the AP allow us to take control over some of the emission parameters, such as WiFi Channel or transmitting power. The power is setup to 20 dBm and the Channel election will be indicated in each approach.
3. A precision coaxial cable, whose datasheet is in [21], is going to be used in the wired connections. This cable has been chosen because its characteristics allow it to have good behaviour in the chosen frequency band (ISM). The parameters that are going to be important in the setups are its propagation speed ($0.66 \cdot c_0$) and the attenuation factor at 2.4 GHz (which is 0.45 dB/m)
4. A splitter will be used in order to simulate two different length paths of propagation in the cable setup. Its bandwidth is 5-2500 MHz, which is enough to contain the ISM band
5. An attenuator will be used to guarantee that the power that arrives to the oscilloscope never exceed the maximum allowed. This is important only in the cable setup, since in Wireless propagation the losses are bigger. Its attenuation factor is regulable between 0 and 60 dB
6. A digital oscilloscope will be used in order to capture the RF data. The chosen model can be seen in [22], and its characteristics make it appropriate both to work in the ISM band directly (it has a maximum sampling rate of 10 GSamples/s) or to downconvert the data to IF and sampling in that band, which is the feature that is going to be used in the approaches in this work.

The oscilloscope provides two different downconversion schemes, depending on whether IQ data or real data need to be obtained. The block diagram of the first case is in Figure 3.1, whose blocks are a RF filter (which is a broadband one and cannot be adjusted in the oscilloscope parameters), a local oscillator and two frequency mixers, which obtain the In-Phase and Quadrature components by adding a $\pi/2$ phase shift to the local oscillator signal. Finally, two equal IF lowpass filters are used to erase high frequency components from the IQ components, and afterwards they are sampled with the sampling rate indicated.

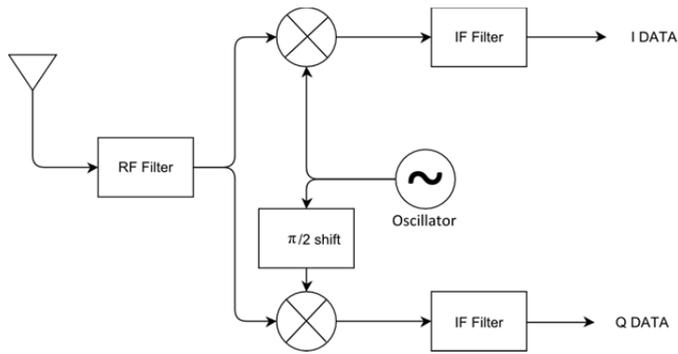


Figure 3.1 - IQ data downconverter block diagram

If real data is obtained instead of complex, the oscilloscope also has a real data downconverter whose diagram is in Figure 3.2. In this case, the blocks are the RF filter (which is the same as in the previous case), a local oscillator and a frequency mixer that makes the frequency downconversion and an IF filter, which eliminates the high frequency components. The signal obtained is sampled after the downconversion is finished.

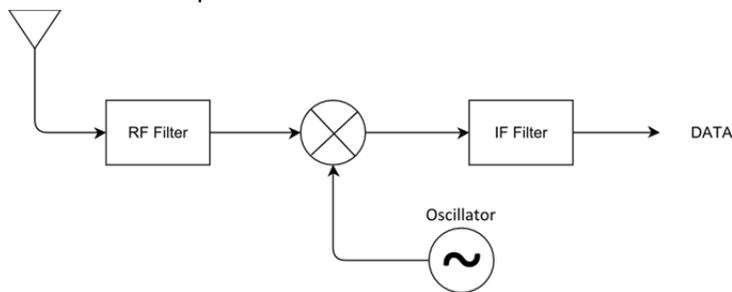


Figure 3.2 - Real downconverter block diagram

Unless indicated otherwise, the parameters used to set up the oscilloscope are in the following table:

Local oscillator frequency	2.395 GHz
IF Low pass filter cutoff frequency	100 MHz
Sampling frequency	250 MHz
Number of samples recorded	10000

Table 3.1 - Default oscilloscope parameters used

7. ISM band antennas, whose model is in [25]. Those antennas are omnidirectional and have an isotropic gain of 5 dBi.
8. A computer with MatLAB installed on it will be used to import the samples from the oscilloscope and to process them. The concrete model of oscilloscope chosen can be connected to MatLAB using a VISA protocol and it can be controlled using scripts. All the code used has been written in MatLAB and part of it is included in the appendixes.

3.2 - CABLE SETUP DESCRIPTION

This setup will be used as a first approach in order to test the methods proposed and also, to make easier the error detection in case that something fails. A schematic description of the setup is presented in the next figure, with the important configuration parameters in Table 3.2:

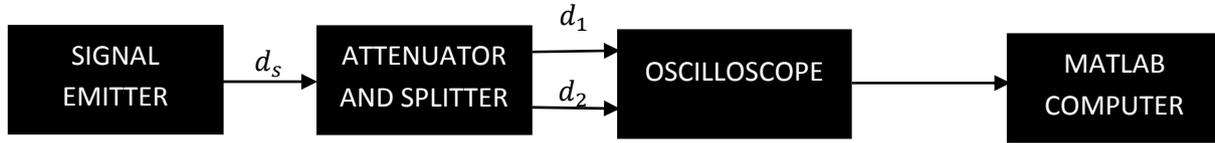


Figure 3.3 - Schematic setup, cable

Distance to the splitter (d_s)	170 cm
Distance channel 1 (d_1)	500 cm
Propagation speed channel 1 (c_1)	$0.66 \cdot c_0$
Distance channel 2 (d_2)	500 or 300 cm
Propagation speed channel 2 (c_2)	$0.66 \cdot c_0$

Table 3.2 - Parameters in the cable setup

In this setup, the signal emitter, which can be either the signal generator or the AP, is connected through a known length cable to the attenuator and the splitter. Two cables are connected between the splitter outputs and two oscilloscope channels, and these cables will simulate the different length propagation paths. After the signal is in the oscilloscope, it will downconvert it according to the algorithm used in each method and sent to MatLAB, in order to obtain a location.

3.2.1 - INFLUENCE OF THE CABLE PROPAGATION SPEED

In this setup, it is necessary to keep on mind the fact that the propagation speed in the cable could dramatically affect the results obtained. So it is crucial to validate the speed propagation that the manufacturer gives in the datasheet [21].

In order to do so, the experimental setup described above will be used with two cables whose length is known. The wavelength is related to the distance difference between propagating channels (in this case, the two cables) according to:

$$d_1 - d_2 = n \cdot \lambda = n \cdot \frac{c}{f_c} \rightarrow n \in \mathbb{N} \quad (3.2.1)$$

Measuring with the oscilloscope the phase difference of the signal received from both channels, the carrier frequency of the pure sinusoid wave is adjusted until this phase difference between channels is zero (in the lowest frequency possible). This will mean that the distance difference between both cables contains, exactly, a wavelength of signal, that is, that $n = 1$ and that allows verifying the speed propagation in the cable.

For example, the cables used in all the cable-based setup, according to their datasheet, have a propagation speed of $0.66 \cdot c_0$. Using the method described above, the frequency in which a zero phase shift should be obtained was 98.5 MHz, with a cable length difference of 2 m. So, applying the expression (3.2.1), it is possible to check whether the information in the datasheet was accurate or not:

$$2 = \frac{c}{98.5 \cdot 10^6} \rightarrow c = 1.97 \cdot 10^6 \cong 0.66 \cdot c_0$$

It is important to note that, for this setup, the same cable type must be used. It is also possible do that with different cable types, with different propagation speeds, but the expression to be used would be different.

Finally, special care must be taken with any possible phase shift in the system because some of the locating methods to be implemented are based on phase measurements. Thus, the splitter must be symmetric in terms of phase (or at least, introduce a constant phase shift between its outputs), and so with all the components in the system. Also, the constant phase shift introduced in both signals by the cable between the oscilloscope and the splitter d_s and other phase shifts introduced in both signals can be ignored since the difference of phases is being computed in the proposed methods which are based in phase measurements to locate, so any constant term added to both signals will not affect the final result.

3.3 - LINE-OF-SIGHT SETUP 1 DESCRIPTION

The setup description will be different to the one used in the cable-based setup section. It will be used to test the methods behavior in a real LOS environment, and the schematic would be as follows:

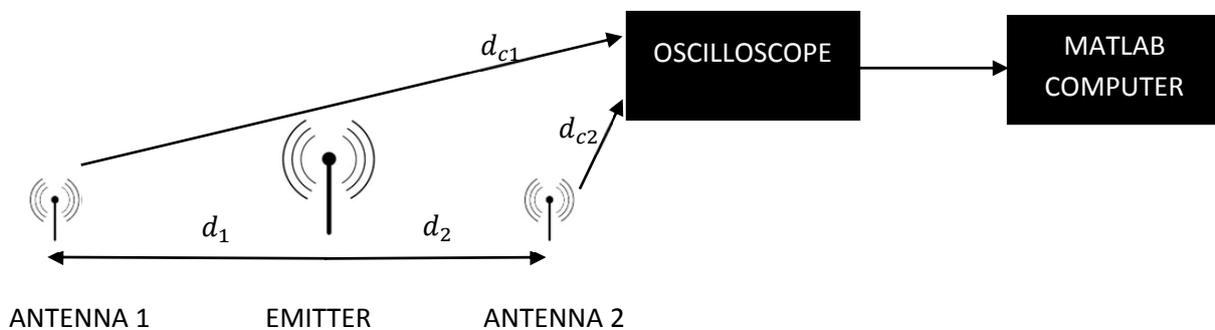


Figure 3.4 - Schematic setup, LOS

The components used are the following:

1. Signal emitter, that will be the AP (in this setup the signal generator will not be used)
2. Two receiving antennas. In order to simplify the calculations, a one-dimensional approach will be used in which the two antennas and the emitter will follow a straight line. The emitter will move following this imaginary line which goes from one antenna to the other, so that the concept behind the cable setup is the same in this one. The distances definition used in that approach is still valid as well: d_1 is the distance from the emitter to the first antenna, d_2 is the distance from the emitter to the second antenna and $d = d_1 + d_2$ is the total distance. In this setup, $d = 6$ m will be used unless it is indicated otherwise.
3. Two cables will communicate the antennas to the oscilloscope. These cables do not have to have the same distance and/or propagation speed, since any variation in those parameters will mean

only a frequency-dependent phase shift, which can be easily computed, but in the setup used they were equal cables, with a length of 5 meters.

4. The oscilloscope and the MatLAB computer are the same as in the previous setup, so their main setup parameters can be seen in that section (as sampling frequency, intermediate frequency and so). However, the power that arrives to the oscilloscope is inferior that in the cable based setup, so the triggering level will be different.
5. Finally, in a LOS environment there will be thermal noise and signal interference that will cause the oscilloscope to trigger when noise is present instead of signal. However, since under the proposed setup the signal level is significantly higher than the noise floor, it is possible to discriminate between signal and noise using a basic power threshold. So, after the signal from the oscilloscope has been imported by the computer, the first step is to compute the power of the signal in one channel, and compare it to a fix threshold. If the power of the signal is lower than this threshold, the signal will be considered to be noise and the samples will not be used in order to compute the location measurements. That way has proved to be enough in LOS conditions, and the concrete threshold can be easily estimated observing the oscilloscope data. In this concrete setup, a power threshold between $2 \cdot 10^{-3}$ and $200 \cdot 10^{-3} \text{ mW} \cdot \Omega$ has proved to work quite well. A different option would be using a level based threshold, which works similarly: instead of using the signal power, it uses the signal average value. Both options have been implemented and used in the setups, and both work fine.
6. The environment will be a LOS setup, as clear as possible. To satisfy that condition, the antennas and the signal emitter will be placed at a distance of approximately 2 meters from the ground.

3.4 - LINE-OF-SIGHT SETUP 2 DESCRIPTION

The LOS setup described in the section before will be used in some of the methods proposed, however, there will be some especial cases in which that setup will not be enough because some methods will have a centimeter precision and subsequently, a more precise setup is needed.

The components that are going to be used are the same that in the previous setup, but their placement will be different: the access point and the two antennas will be placed in a straight line, and one of the antennas will be moved along that line in order to change the distance difference between received signals. A schematic diagram is shown next to illustrate how the setup has been done.

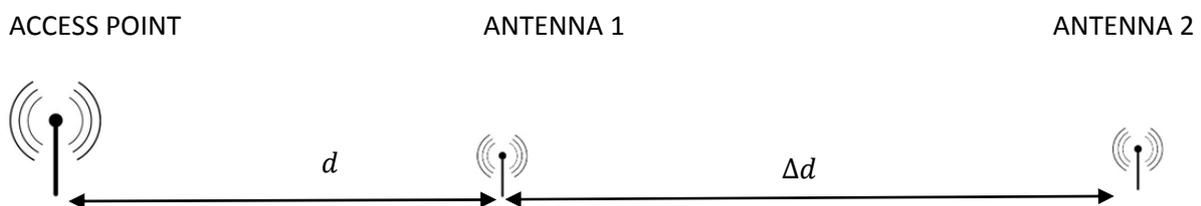


Figure 3.5 - Schematic setup, LOS

The distance between the access point and the first antenna is d and the distance difference between antennas is Δd . Both antennas are equal and are connected to the oscilloscope with two equal cables. When increasing Δd , in order to keep fixed d , the second antenna will be moved, so that a higher precision is achieved.

Chapter 4 - THEORETICAL DESCRIPTIONS AND RESULTS

4.1 - OVERVIEW

In this chapter, four different methods to locate will be tested, using 802.11b signals since one of the goals of this work is locating using out of the shelf components. The methods implemented can be classified according to the parameter of the signal which is used to locate:

- Phase-based methods: Three different methods which use phase information are implemented in this work. In the first of them, a frequency sweep is done in order to take advantage of the fact that the phases of the signals which have propagated through different paths depend on the carrier frequency and the distance difference between both paths, so having frequency information allows us estimating the distance difference and thus, locating.
Another method which is going to be put to the test is based on the FFT of a WiFi Channel. It is based on the same principia as the one before, but the calculations are different and also, the possibilities of reaching a tradeoff between system hardware requirements and accuracy substantially differ from both methods.
Finally, an IF cross correlation method will be used to estimate the phase differences due to the different paths that the waves have propagated through. This method has even more parameters than the others that can be adjusted in order to modify the location resolution, the system hardware requirements and thus, the cost.
All of those methods, however, have a common limitation: since they are based on phase measurements, there will be a limit on the maximum location range due to the phase periodicity.
- Power based method. This is a simple RSSI method implementation, which will have no fingerprinting information and will train the propagation constant before locating, so that it can adapt better to the concrete room where the location system is.

Finally, all the signal processing will be applied after doing an IF downconversion and sampling the received signals. For testing purposes, the oscilloscope and other components described in Section 3.1 will be used, but if a real life implementation wants to be done, a commercial downconverter and ADC can be used instead.

4.2 - METHOD 1: FREQUENCY SWEEP

In this section, the main idea behind the paper in [8] will be adapted and put to the test in order to clarify whether the proposed approach in that paper is feasible using WLAN 802.11b standard. This method is based on the use of different carrier frequencies, and it has already been shown in Section 2.4.4 that in order to obtain a better unambiguity range in the measurements, that should be done.

4.2.1 - THEORETICAL EXPLANATION

In this approach, the signal model in Sections 2.4.1 and 2.4.2 is going to be used. Let us suppose a pure sinusoidal wave, that is, a phasor, which could be mathematically studied according to the expression in (2.4.14) as follows:

$$x_{rf} = Re \{A_1 \cdot e^{j(\omega t + kd + \theta)}\} = Re \left\{ A_1 \cdot e^{j \left(2\pi f t + \frac{2\pi}{\lambda} d + \theta \right)} \right\} = A_1 \cdot \cos \left(2\pi f t + \frac{2\pi}{\lambda} d + \theta \right) \quad (4.2.1)$$

Where θ is a constant phase term, f is the carrier frequency, t is the time, λ is the wavelength, A_1 the received signal amplitude and d is the distance or the spatial dimension. The phase of the wave, which is time and distance dependent, is:

$$\varphi(t, d) = 2\pi f t + \frac{2\pi}{\lambda} d + \theta \quad (4.2.2)$$

Now, let us suppose that the mentioned wave is transmitted through two different lengths mediums, with different propagation speeds, that will be called d_a and c_a . At the end of each cable, the following phases will be obtained:

$$\text{Cable 1} \rightarrow \varphi_1(t, d_1) = 2\pi f t + \frac{2\pi}{\lambda_1} d_1 + \theta = 2\pi f t + \frac{2\pi f}{c_1} d_1 + \theta = 2\pi f \left(t + \frac{d_1}{c_1} \right) + \theta \quad (4.2.3)$$

$$\text{Cable 2} \rightarrow \varphi_2(t, d_2) = 2\pi f t + \frac{2\pi}{\lambda_2} d_2 + \theta = 2\pi f t + \frac{2\pi f}{c_2} d_2 + \theta = 2\pi f \left(t + \frac{d_2}{c_2} \right) + \theta \quad (4.2.4)$$

If the phases received at the end of the two cables in the same time are subtracted, the following expression will be obtained:

$$\varphi_1(t, d_1) - \varphi_2(t, d_2) = 2\pi f \left(\frac{d_1}{c_1} - \frac{d_2}{c_2} \right) \quad (4.2.5)$$

So, there is a linear relation between the frequency and the phase, if the distances and propagation speeds difference remains constant. Thus, it should be possible, provided that the speeds difference and the total distance ($d = d_1 + d_2$) are known, to perform a frequency sweep in order to determine either d_1 or d_2 , that is, changing the carrier frequency so that different phase differences can be obtained. Thus, if the slope of the linear function obtained with the different phase differences is obtained, the calculation of the distance will be possible. Let be m the slope of the function obtained, so the linear fit should follow this expression:

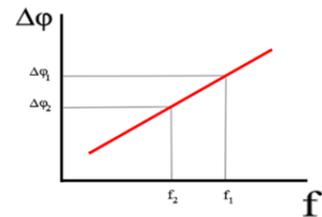


Figure 4.1 - Linear slope function

$$\Delta\varphi = \varphi_1 - \varphi_2 = m \cdot f \rightarrow m = \frac{\Delta\varphi(f_1) - \Delta\varphi(f_2)}{f_1 - f_2} = 2\pi \left(\frac{d_1}{c_1} - \frac{d_2}{c_2} \right) \quad (4.2.6)$$

Considering that $d = d_1 + d_2$, the total length of the cables, is known, it is possible to proceed as follows to compute d_1 :

$$m = 2\pi \left(\frac{d_1}{c_1} - \frac{d_2}{c_2} \right) = \frac{2\pi}{c_1 c_2} (c_2 d_1 - c_1 d_2) = \frac{2\pi}{c_1 c_2} (c_2 d_1 - c_1 (d - d_1)) = \frac{2\pi}{c_1 c_2} (d_1 (c_2 + c_1) - c_1 d)$$

And rearranging terms, the expression to be implemented would be:

$$d_1 = \left(m \frac{c_1 c_2}{2\pi} + c_1 d \right) \frac{1}{c_2 + c_1} \quad (4.2.7)$$

A problem that must be taken into account is the fact that the phase is periodical with the distance difference, so there will be a limit in the maximum distance that can be obtained without ambiguity, which is going to be 2π . To calculate this maximum distance, let us suppose that two different consecutive frequencies are had in the sweep performed, namely f_1 and f_2 , so that their difference is $\Delta f = f_1 - f_2$. Their phase difference, if $c_1 = c_2 = c$ is considered, will be according to the expression (4.2.6):

$$\Delta\varphi(f_2) - \Delta\varphi(f_1) = \frac{2\pi}{c} \cdot (f_1 - f_2) \cdot (d_1 - d_2) = \frac{2\pi}{c} \cdot \Delta f \cdot \Delta d \quad (4.2.8)$$

Now, let us suppose another phase difference between the two same frequencies f_1 and f_2 , but with another distance difference, let it be $\Delta d + d_a$, so that the phases differences difference would be a 2π multiple, the ambiguity distance d_a could be obtained:

$$(\Delta\varphi(f_2) - \Delta\varphi(f_1))|_{\Delta d} - (\Delta\varphi(f_2) - \Delta\varphi(f_1))|_{\Delta d + d_a} = \frac{-2\pi}{c} \cdot \Delta f \cdot d_a = 2k\pi$$

Thus, rearranging terms:

$$d_a = k \frac{c}{f_2 - f_1} \quad (4.2.9)$$

Where $k \in \mathbb{Z}$. Considering $k = 1$, the maximum distance that can be measured without unambiguity is obtained, which turns out to be:

$$d_a = \frac{c}{f_2 - f_1} \quad (4.2.10)$$

The maximum range that can be measured without ambiguity depends on the interval between consecutive frequencies on the sweep performed. That allows adjusting the precision, depending on the distance that need to be covered.

Another way to calculate this unambiguity range comes from the observation of the slope. Since modulus 2π will be used in the calculations, there is a maximum slope between two consecutive frequencies, f_1 and f_2 , of 2π . Mathematically expressed:

$$m_{max} = \frac{2\pi}{\Delta f} \quad (4.2.11)$$

Applying the slope definition in (4.2.6):

$$m_{max} = \frac{2\pi}{\Delta f} = \frac{2\pi}{c} (d_1 - d_2) = \frac{2\pi}{c} \Delta d \quad (4.2.12)$$

Since the conditions in (4.2.12) and (4.2.8) are the same, from this point on the calculations would be the same.

4.2.2 - ALGORITHM DESCRIPTION

The proposed algorithm used to estimate the location using the method described above would be as follows:

1. An emitter sends a wave, which propagates in two different length (and maybe speed) channels, so when it arrives to the oscilloscope, there will be a phase shift between the two waves
2. The oscilloscope mixes the two waves, taking them to complex intermediate frequency using an IQ downconverter. That way it is possible to reduce the computational efforts, since according to Nyquist Theorem, the sample rate of the wave could be reduced (and also, the calculus complexity and time required).
3. Once the IQ data of the downconverted waves is obtained, MatLAB will read them and process them. Each measure process will be repeated N times and the final result will be an average measure of the distance estimations obtained.
4. Since a frequency sweep is needed in order to achieve the phase slope, the whole process will be repeated using a different carrier frequency, that is, a different WiFi Channel will be used in each measurement in order to obtain this slope.
5. Once the phase differences have been computed, since the frequencies of the sweep are known, it is possible to obtain the slope of the function. The method used to fit the data to a linear function will be the least-squares one.
6. After the slope has been obtained, the distance is calculated according to the expression deduced before in (4.2.7).

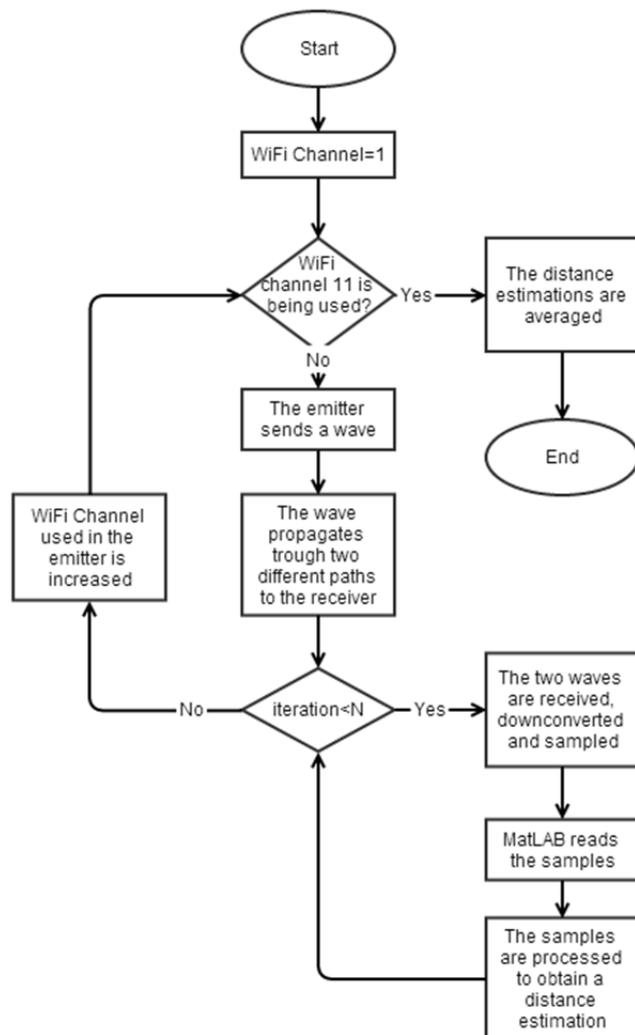


Figure 4.2 - Algorithm flow chart, frequency sweep

4.2.3 - CABLE-BASED SETUP

In order to test this approach, the setup described in Chapter 3.2 will be used here.

4.2.3.1 - RESULTS USING A SIGNAL GENERATOR

Using the signal generator as signal emitter, ten repetitions and averaging the ten measures obtained using the mean, the results obtained are:

- a) If $d_1 = d_2 = 5m$ is taken, the theoretical slope should be:

$$m = 2\pi \left(\frac{d_1}{c_1} - \frac{d_2}{c_2} \right) = 0$$

Running the algorithm, the results obtained are: $m = 6.0477 \cdot 10^{-10} \rightarrow d_1 = 5.0095 m$

In the next two figures, it is possible to see the representation of the phase difference obtained and the linear fit done by MatLAB, as well as the information obtained in the 2.42 GHz frequency.

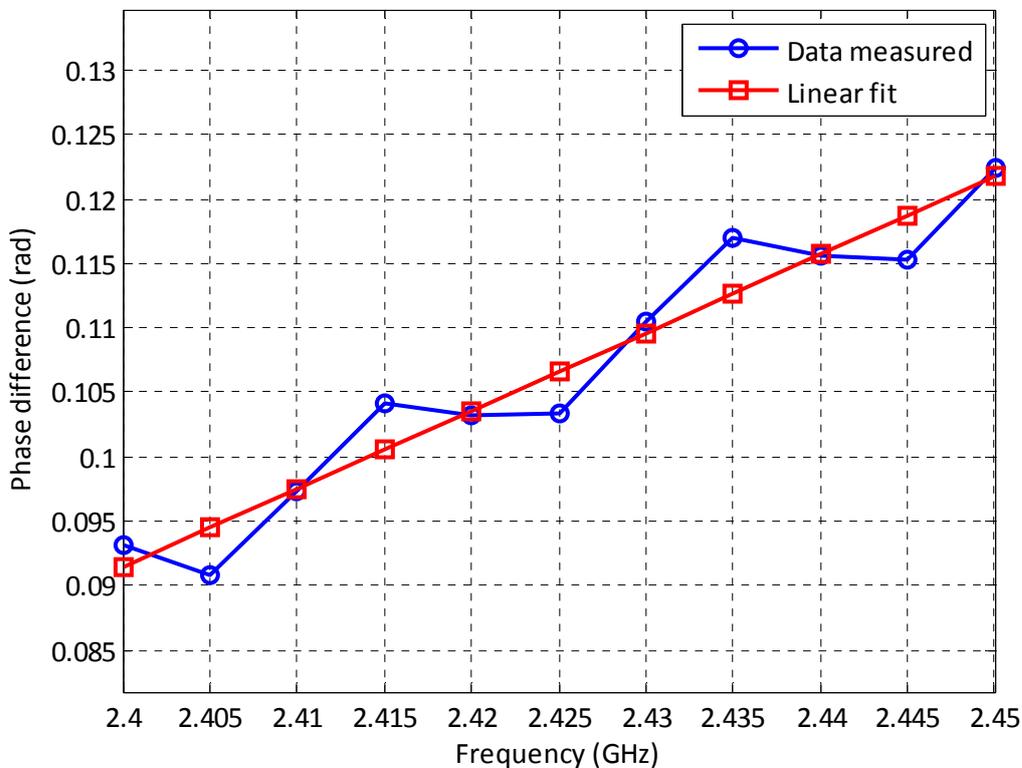


Figure 4.3– Phase slope obtained using a signal generator, cable setup ($d_1 = d_2 = 5m, d = 10 m$)

- b) If take $d_1 = 5 m, d_2 = 3 m$ is taken, the theoretical slope should be:

$$m = 2\pi \left(\frac{d_1}{c_1} - \frac{d_2}{c_2} \right) = 6.3466 \cdot 10^{-8} \text{ rad/Hz}$$

Running the algorithm, the results obtained are:

$$m = 6.3362 \cdot 10^{-8} \text{ rad/Hz} \rightarrow d_1 = 4.9984 m$$

In the next figure, it is possible to see the representation of the phase difference obtained and the linear fit done by MatLAB.

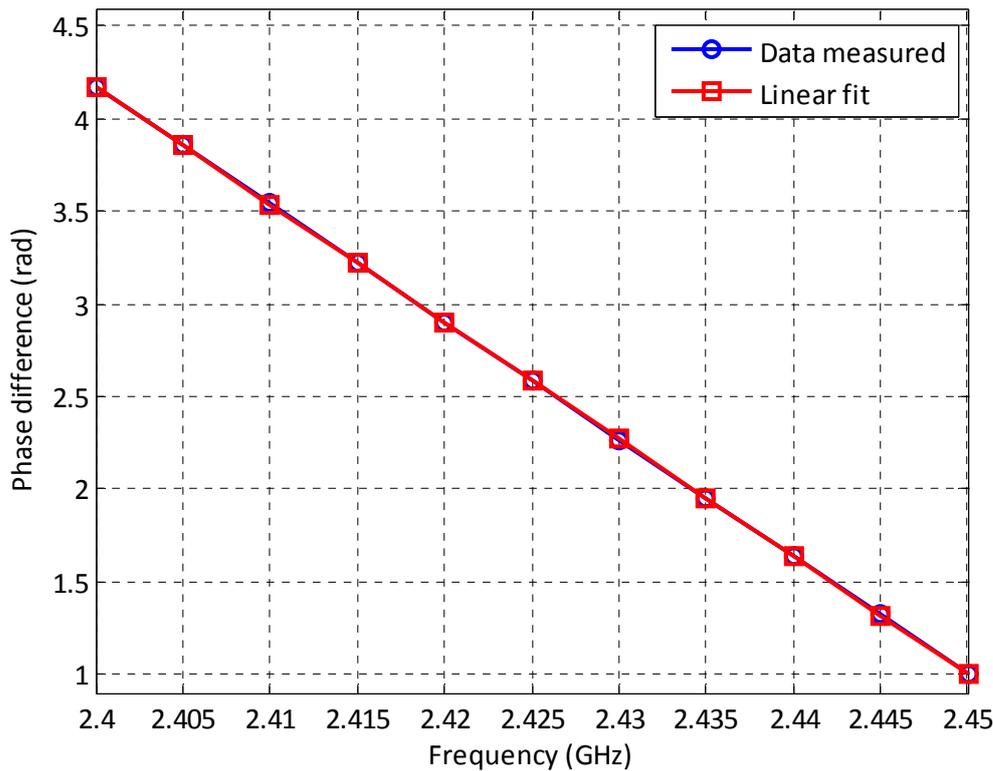


Figure 4.4 – Phase slope obtained using a signal generator, cable setup ($d_1 = 5m, d_2 = 3m, d = 8m$)

4.2.3.2 - RESULTS USING A WLAN ACCESS POINT

In this step, the signal emitter will be changed and the access point will be used. There is a new problem that arises in this concrete setup and it is the noise increase, as it can be seen in Figure 4.5. First, there is more thermal noise present than when using the signal generator, because the oscilloscope triggers a bit before the signal is captured and, so the first samples are pure noise. The second noise present is the phase noise, inherent to the internal oscillator of the access point. This is the noise which can be observed in the phase difference after the thermal one, and consists of deviations from the constant value that the phase should have.

In the signal generator setup, it was observed that using the mean to average the results obtained in each experiment was enough to get accurate results. However, with the access point the increase in the amount of noise present must be kept in mind, and so two different ways to average the phase difference are tested: first, the mean will be used, as in the previous setup, and then, the median. Theoretically, the median will yield superior results, since it is a statistical indicator which works better with impulsive noise (as the phase one), which is the one that dominates in the signal obtained.

The following results are obtained using the access point and the mean-based approach:

- a) If $d_1 = d_2 = 5m$ is taken, the theoretical slope should be, as it was demonstrated before, 0 rad/Hz

Running the algorithm, the results obtained are: $m = 2.6038 \cdot 10^{-9} \rightarrow d_1 = 5.041 m$

In the Figure 4.6, it is possible to see the representation of the phase difference obtained and the linear fit done by MatLAB.

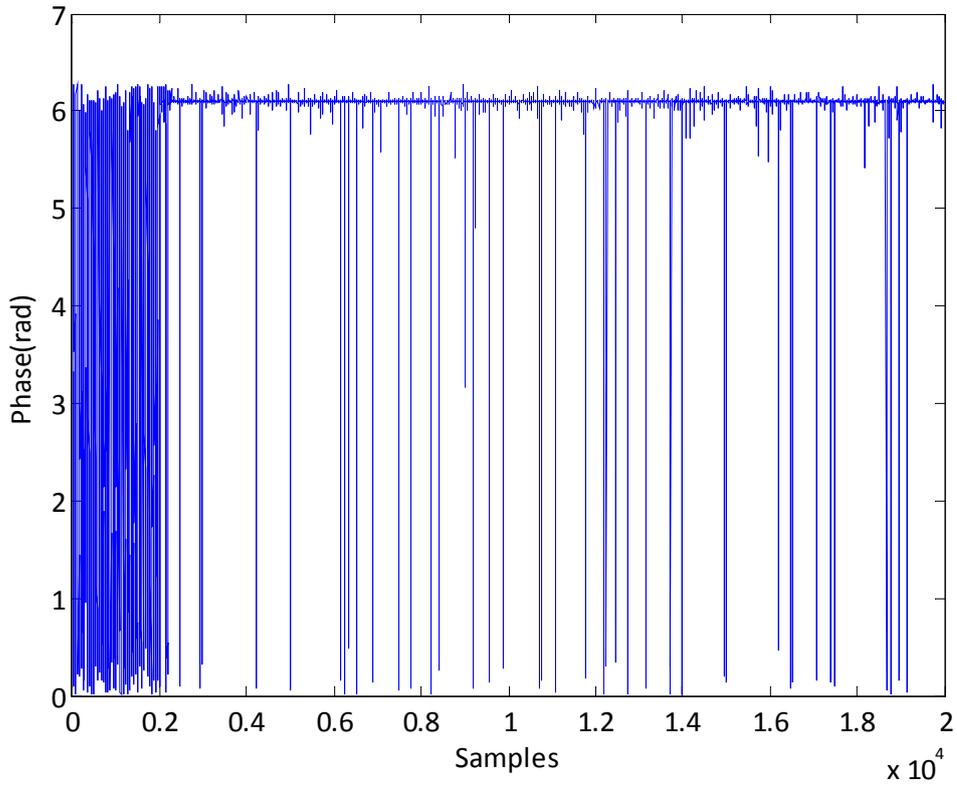


Figure 4.5 – Phase difference obtained using a WLAN access point, WiFi channel 1, cable setup ($d = 10\text{ m}$)

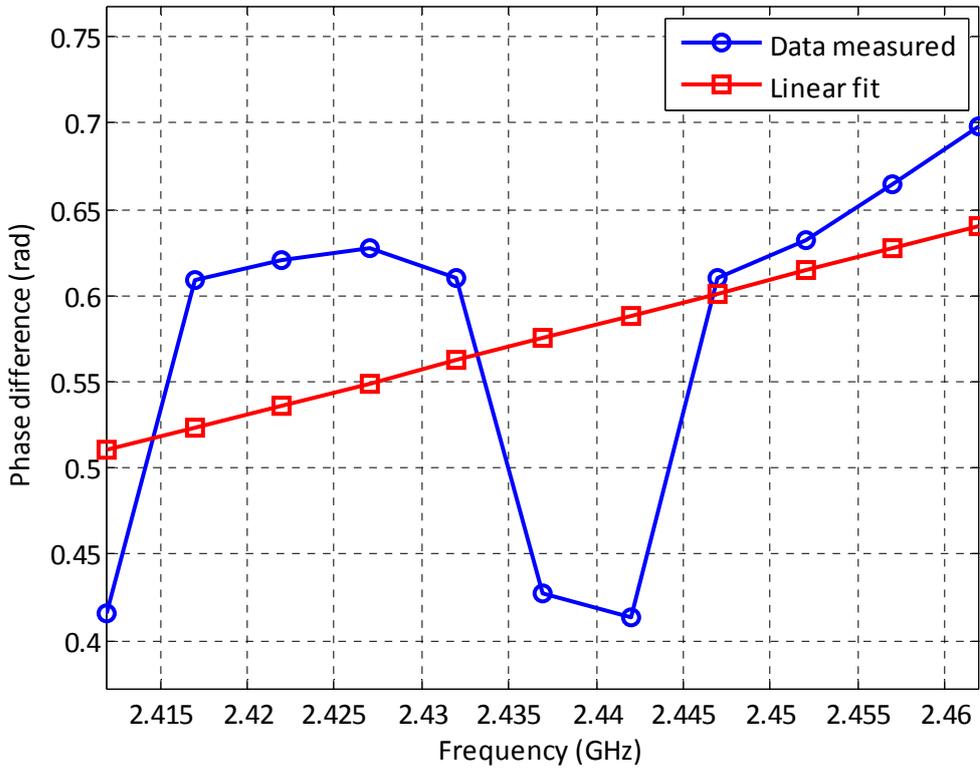


Figure 4.6 – Phase slope obtained using a WLAN access point, mean, cable setup ($d_1 = d_2 = 5\text{ m}$, $d = 10\text{ m}$)

b) If $d_1 = 5\text{ m}, d_2 = 3\text{ m}$ is taken, the theoretical slope should be, as it was demonstrated before, $6.3466 \cdot 10^{-8}\text{ rad/Hz}$

Running the algorithm, the results obtained are: $m = 5.2281 \cdot 10^{-8} \rightarrow d_1 = 4.8228\text{ m}$

In Figure 4.7, it is possible to see the representation of the phase difference obtained and the linear fit done by MatLAB.

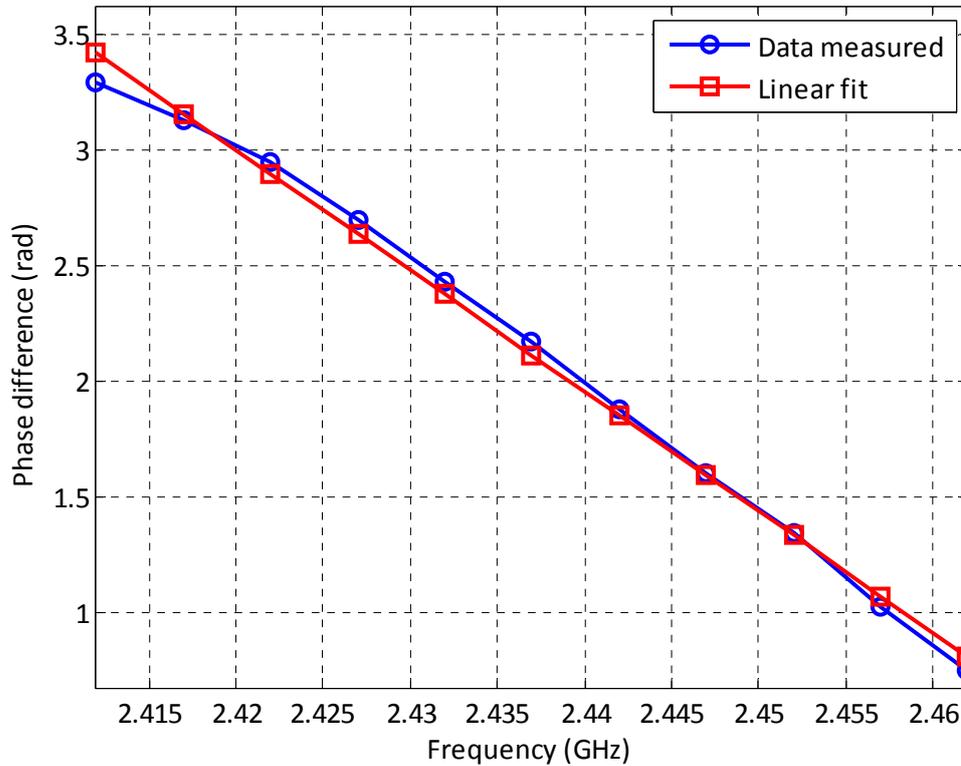


Figure 4.7 – Phase slope obtained using a WLAN access point, mean, cable setup ($d_1 = 5\text{ m}, d_2 = 3\text{ m}, d = 8\text{ m}$)

The following results are obtained using the access point and the median-based approach:

a) If $d_1 = d_2 = 5\text{ m}$ is taken, the theoretical slope should be, as it was demonstrated before, 0 rad/Hz

Running the algorithm, the results obtained are: $m = 4.3288 \cdot 10^{-10} \rightarrow d_1 = 5.0068\text{ m}$

In Figure 4.8, it is possible to see the representation of the phase difference obtained and the linear fit done by MatLAB.

b) If $d_1 = 5\text{ m}, d_2 = 3\text{ m}$ is taken, the theoretical slope should be, as it was demonstrated before, $6.3466 \cdot 10^{-8}\text{ rad/Hz}$

Running the algorithm, the results obtained are: $m = 5.9326 \cdot 10^{-8} \rightarrow d_1 = 4.9348\text{ m}$

In Figure 4.9, it is possible to see the representation of the phase difference obtained and the linear fit done by MatLAB.

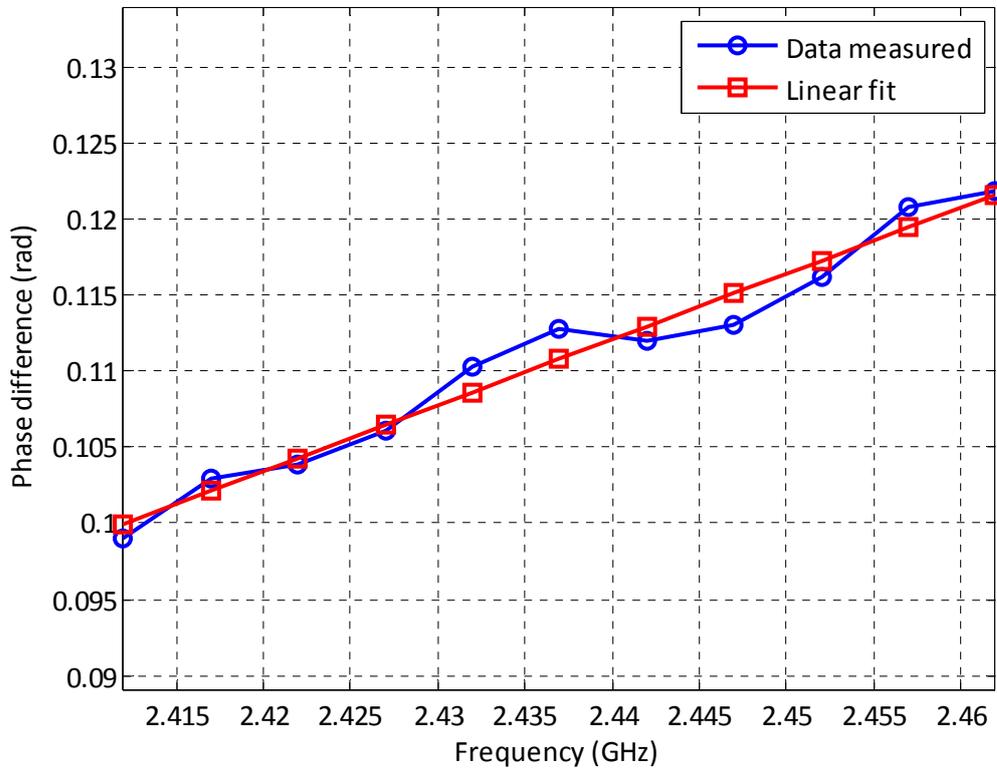


Figure 4.8 – Phase slope obtained using a WLAN access point, median, cable setup ($d_1 = d_2 = 5\text{ m}$, $d = 10\text{ m}$)

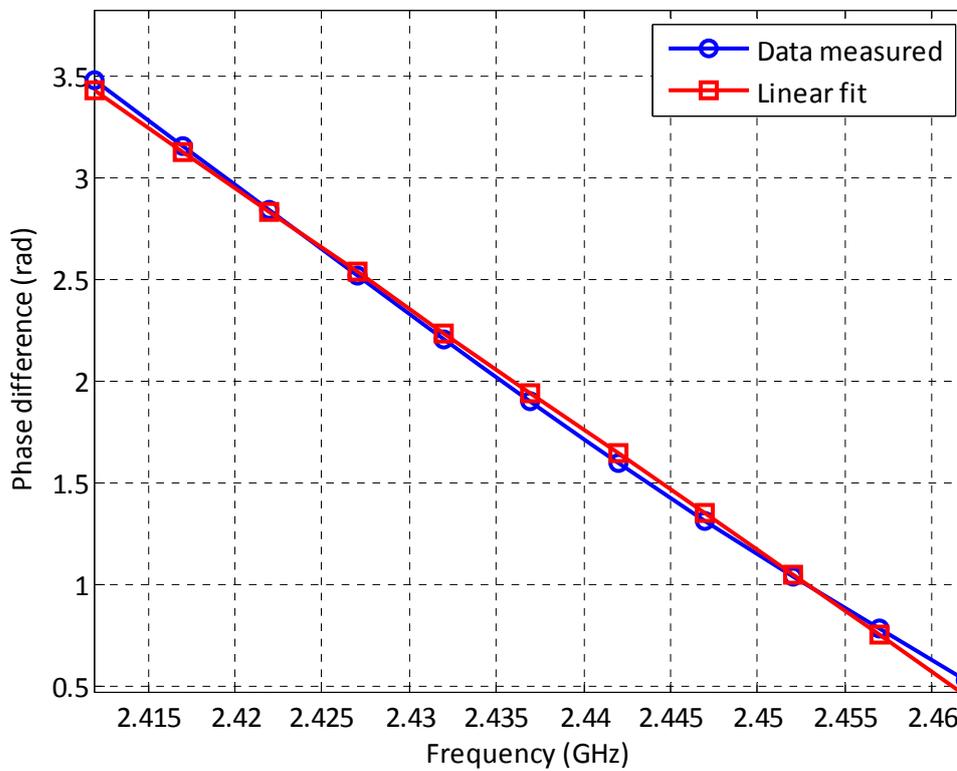


Figure 4.9 – Phase slope obtained using a WLAN access point, median, cable setup ($d_1 = 5\text{ m}$, $d_2 = 3\text{ m}$, $d = 8\text{ m}$)

4.2.3.3 - RESULTS AND CONCLUSIONS

SETUP	$d_1(m) / c_1$	$d_2(m) / c_2$	$d (m)$	Slope obtained (rad/Hz)	d_1 obtained (m)	Absolute error (cm)	Relative error (%)
Function generator	$5/0.66 \cdot c_0$	$5/0.66 \cdot c_0$	10	6.0477 e-10	5.0095	0.95	0.19
	$5/0.66 \cdot c_0$	$3/0.66 \cdot c_0$	8	6.3362 e-8	4.9984	0.16	0.032
WLAN cable, mean	$5/0.66 \cdot c_0$	$5/0.66 \cdot c_0$	10	2.6038 e-9	5.041	4.1	0.82
	$5/0.66 \cdot c_0$	$3/0.66 \cdot c_0$	8	5.2218 e-8	4.8228	17.72	3.544
WLAN cable, median	$5/0.66 \cdot c_0$	$5/0.66 \cdot c_0$	10	4.3288 e-10	5.0068	0.68	0.136
	$5/0.66 \cdot c_0$	$3/0.66 \cdot c_0$	8	5.9326 e-8	4.9348	6.52	1.304

Table 4.1 - Results of frequency sweep approach, cable setup

Some of the conclusions until this point are:

1. The detection method based on phase measurements seems to work really fine in controlled conditions. So far, errors below the 5% have been always obtained, with a standard accuracy close to the 1%
2. When noise is present (WLAN cable setups), changing the statistical processing based on mean to a median based one improves the results significantly. So, it seems to be preferable employing the median-based approach, since noisy environments will be the real ones. Also, according to the plots, the adjustment is better in median calculated phases than in mean ones, due to the fact that median is very robust to impulsive noise, whereas the mean is easily distorted (that can specially be observed in the 10m setup, where the linear adjustment does not fit as good to the data obtained as the median approach)

4.2.4 - LOS SETUP

The setup described in Chapter 3.3 is going to be used in order to test the proposed approach in a wireless environment. This is a more challenging setup than the one before: the problem of noise, which appeared on the WLAN Access Point cable-based setup, will get worse since the thermal noise floor increases. Interference is going to be another problem, because there will be signals coming from any close WLAN 802.11x device (such as another access points, laptops, smartphones...). There might be multipath propagation related problems as well and finally, another problem that will arise is related to the signal attenuation: in short distances, there will be a higher attenuation in radio propagation than in cable one.

In order to deal with those problems, the following modifications will be made to the algorithm used:

1. The phase difference calculations will always use the median-based approach, since it has already been shown that it is expected to work better against phase noise, which is the dominant noise component.
2. The noise and the signal attenuation will bear problems to the triggering in the oscilloscope. Thus, the trigger must be very carefully set in order to capture the desired signal instead of

noise. Also, since invalid triggers are bound to happen, a noise amplitude threshold will be implemented in the code, so that if a signal is captured and its mean amplitude is smaller than the threshold, it will be considered noise and will not be used to calculate the phase differences.

3. In case that a constant phase shift exists in the system, it must be taken into account in order to get an accurate position. If this constant shift is ignored, then a bias error will appear in the results (so, if that happened, it would be important to check that the system does not introduce this constant phase shift by any reason)

Finally, the antennas election on this setup is going to be really important, since any problem with them will have consequences on the results obtained. So, it will be critical choosing the correct antennas, paying special attention to their phase characteristics. The ideal situation is that they are exactly the same, so the phase shift they might introduced would be the same in both and so, according to equations (4.2.3), (4.2.4) and (4.2.5), this constant phase term would be eliminated when processing the signal. Also, another important point is that the antennas are designed to work in the frequency band that it is being used, in this case, the ISM band (so, a WiFi antenna could be used), to make sure that the results are coherent and there is no error introduced by a wrong antenna, and also, to facilitate the triggering process in the oscilloscope. The requirements in terms of frequency response, then, are critical in order to test the proposed setup.

Because of that, a calibrating method is described in Appendix 1 in order to test whether the system will behave as it is expected or not. Those calibration steps may help to check whether the antenna election and general setup is correct or not.

4.2.4.1 - RESULTS AND CONCLUSIONS

In this setup, the algorithm has been programmed to take 10 measures of each frequency phase difference at each single position, and to compute ten different distance estimations in each position. The total distance between antennas is $d = 6 \text{ m}$ and the noise amplitude threshold used in the calculations is $2 \cdot 10^{-4} \text{ mV}$, so the signal mean tension must be greater than that value in order to proceed to estimate the distance.

Also, there are problems due to the modulus operation and the 2π jumps which appear in the phase measurements. The solution used in Appendix 1 (shifting π radians and calculating again the modulus) can correct only one jump in the whole sweep, but as it was implied in equations (4.2.11) and (4.2.12), it is possible to have more jumps, and so the phase needs to be unwrapped in order to get correct measurements. Thus, the calculations will be done using the phase with jumps (which is bound to be not really accurate) and the phase unwrapped, so that the 2π jumps disappear and so, the correct slope is measured. An example of this problem when $d_1 = 0$ and its solution can be seen in Figure 4.10.

Finally, all the measures are taken using phase measurements in the 11 Channels, or in the equidistant channels 1, 6 and 11, whose carrier frequencies difference is 25 MHz. That means that, according to equation (4.2.10), the unambiguity range will not be the same in both cases:

1. Unambiguity distance for 11 Channel measurements:

$$d_a = \frac{c}{f_2 - f_1} = 60 \text{ m}$$

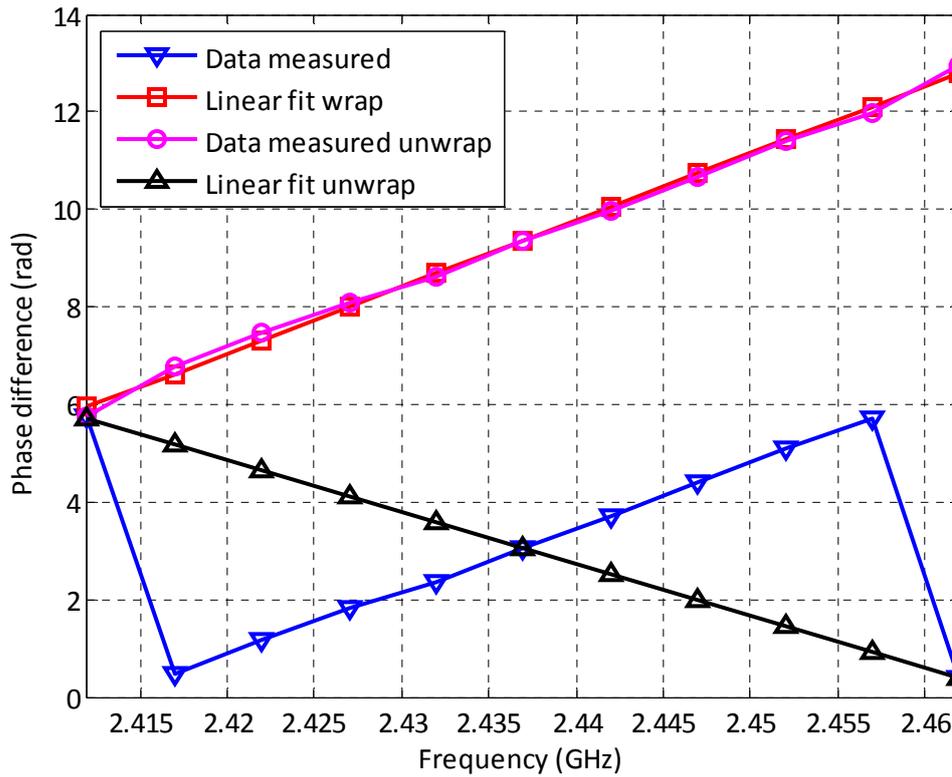


Figure 4.10 – Phase slope offset obtained, 11 WiFi Channels, with phase wrapped and unwrapped, LOS setup

2. Unambiguity distance for 3 Channel measurements:

$$d_a = \frac{c}{f_2 - f_1} = 12 \text{ m}$$

Since the distance separation between antennas $d = 6 \text{ m}$ is smaller than the unambiguity distances, there will be no problem due to this unambiguity in any of the cases analyzed.

The results obtained are in Table 4.2 and Table 4.3, depending on the number of channels used in the measurements and on the use of the wrapped phase or the unwrapped one. Also, the CDF of the absolute error of the results obtained are in Figure 4.11 (for the CDF, 110 different measures were used)

A) Using 11 channels

d_1 (cm)	Mean, wrapped (cm)	Absolute error (cm)	Relative error (%)	Mean, unwrapped (cm)	Absolute error (cm)	Relative error (%)
0	252.13	252.13	42.03	3.96	3.96	0.66
50	181.28	131.28	21.88	44.92	5.08	0.85
100	370.67	270.67	45.11	-49.33	149.33	24.89

150	451.63	301.63	50.27	165.26	15.26	2.54
200	161.89	38.11	6.35	161.89	38.1	6.35
200	268.5	18.5	3.08	268.5	18.5	3.08
300	335.4	35.4	5.9	335.4	35.4	5.9
350	284.38	65.62	10.94	480.74	130.74	21.79
400	418.4	18.4	3.07	418.4	18.4	3.07
450	365.74	84.26	14.04	442.1	7.9	1.31
500	449.69	50.31	8.38	449.69	50.31	8.38
550	488.98	61.01	10.17	625.35	75.35	12.56
600	207.7	392.3	65.38	616.79	16.79	2.79

Table 4.2 - Results of frequency sweep approach, LOS setup, using WiFi antennas and 11 Channels

B) Using 3 channels

d_1 (cm)	Mean, wrapped (cm)	Absolute error (cm)	Relative error (%)	Mean, unwrapped (cm)	Absolute error (cm)	Relative error (%)
0	547.72	547.72	91.29	547.72	547.72	91.3
50	329.6	279.6	46.6	209.6	159.6	26.6
100	415.98	315.98	52.66	115.99	15.99	2.66
150	490.33	340.33	56.72	220.33	70.33	11.72
200	176.28	23.71	3.95	176.29	23.71	3.95
200	259.95	9.95	1.66	259.95	9.95	1.66
300	318.72	18.72	3.12	318.72	18.72	3.12
350	278.06	71.93	11.99	278.06	71.93	11.99
400	374.53	25.47	4.24	374.53	25.47	4.24
450	367.93	82.07	13.68	367.93	82.07	13.68
500	457.89	42.11	7.02	457.89	42.11	7.02
550	334.45	215.55	35.93	334.45	215.55	35.93
600	320.9	279.09	46.51	20.91	579.09	96.52

Table 4.3 - Results of frequency sweep approach, LOS setup, using WiFi antennas and 3 Channels

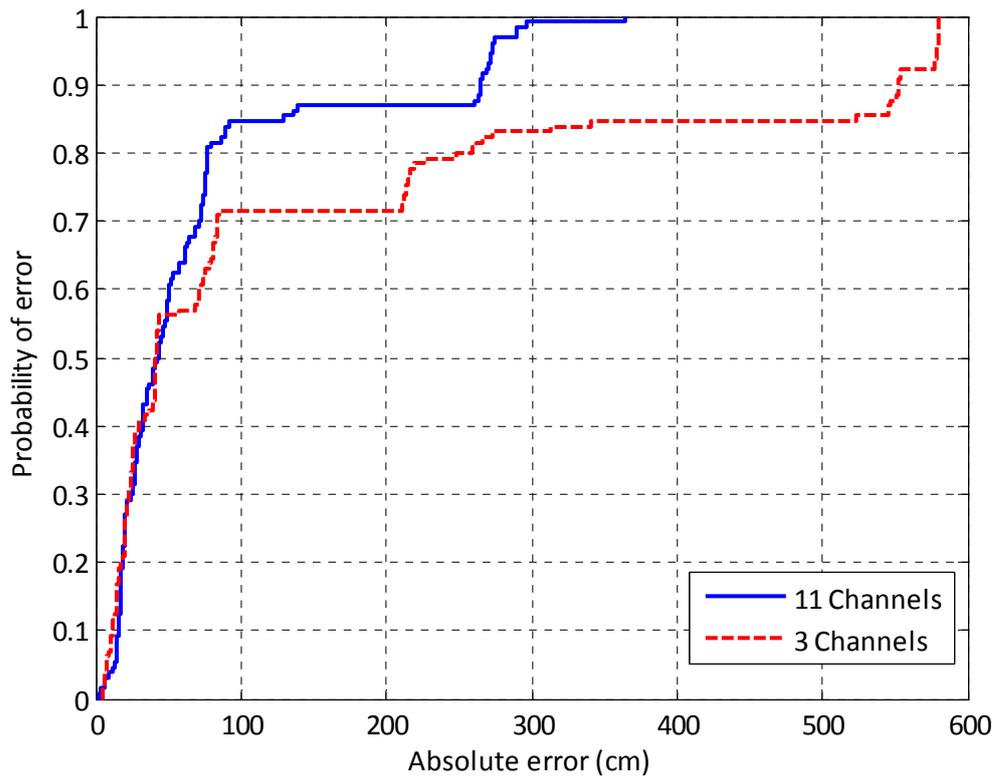


Figure 4.11 – CDF of the absolute error, using 11 and 3 WiFi Channels, phase unwrapped, LOS setup

Conclusions:

1. It is possible to see that the unwrapped phase commits a smaller error than the measurements which contain phase jumps, as it can be seen in Table 4.2 and Table 4.3. Thus, this method is preferable to be used.
2. Also, Figure 4.11 shows that, once that a phase measurement method has been fixed, it is better measuring more channels. With a probability of 80%, if the phase is unwrapped, this accuracy improvement is in the range of 150 cm
3. In the tables, it is possible to see that, independently of the number of channels used and the phase measurement method, the bigger errors tend to be near the antennas (so, $d_1 = 0$ and $d_1 = 6 m$). This error increase can be caused by the fact that an antenna will have different behavior depending on whether a far-field approach can be applied or not. Consequently, the measurements taken near the antenna will have a greater error, due to this problem.
4. Finally, the best results correspond to the 11 Channel measurements, with the phase unwrapped. The accuracy obtained is below 1 m absolute error with an 80% probability.

4.3 - METHOD 2: FFT

In this section, a method based on the use of the Fourier Transform will be used in order to estimate the location. Following the conclusions in Section 2.4.4, different frequencies are used so that the unambiguity range does not become a problem in order to locate. This is a different way to implement the same idea behind the frequency sweep method, but it does not need changing between adjacent WiFi Channels, since the full signal bandwidth can be used in order to obtain the frequency information needed.

4.3.1 - THEORETICAL EXPLANATION

In Section 2.4.2, it was already shown that the 802.11b signal could be studied using the Fourier Transform in IF, especially in equation (2.4.12). In order to provide a wide theoretical foundation for this method, a random signal will be used in the description, so that not only a WiFi signal but any could be used in order to locate using this method. Thus, let us suppose a random signal, which propagates through two different lengths channels, so that at the end of those channels it is possible to obtain the propagated signal and its delayed copy, let them be called $x_1(t)$ and $x_2(t)$. If both channels are LTI (Linear and Time invariant) and if the delay difference between the arrivals of both signals is t_0 , then it is possible to state that $x_2(t) = A \cdot x_1(t - t_0)$, where A_2 is the amplitude difference due to the different path propagations. Applying the Fourier Transform to both signals and using the time-shift property:

$$X_1(f) = FT\{x_1(t)\} = |X_1(f)| \cdot e^{j\theta} \quad (4.3.1)$$

$$X_2(f) = F\{x_2(t)\} = F\{A_2 \cdot x_1(t - t_0)\} = |A \cdot X_1(f)| \cdot e^{j\theta} \cdot e^{-j2\pi f t_0} \quad (4.3.2)$$

Both Fourier Transforms will have the same modulus, just scaled by a constant factor A , and a frequency shift that will be dependent on the time shift t_0 , which is the time difference due to the distance differences between both paths of propagation. Computing the phase difference:

$$\Delta\varphi = \arg(X_1(f)) - \arg(X_2(f)) = 2\pi f t_0 \quad (4.3.3)$$

So, a similar expression to the case in which a frequency sweep was performed (equation (4.2.5)) is obtained. There is a dependency between frequency and phase shift, so the locating scenario 2 is being used in here (Section 2.4.4). A big difference with frequency sweep method is that in this case a sweep is unnecessary since the Fourier Transform already returns information frequency dependent because the bandwidth of the signal is being used. So once the phase difference is obtained, it should be a linear function, whose slope will be:

$$m = \frac{\Delta\varphi(f_2) - \Delta\varphi(f_1)}{f_2 - f_1} = 2\pi t_0 \quad (4.3.4)$$

If both channels had the same propagation speed, let it be c , the distance difference between both channels will be $\Delta d = d_1 - d_2$ and it could be calculated as follows:

$$t_0 = \frac{\Delta d}{c} \rightarrow \Delta d = t_0 \cdot c = \frac{m}{2\pi} c \quad (4.3.5)$$

And since $d = d_1 + d_2$, the distance d_1 will be:

$$d_1 = \frac{d + \Delta d}{2} = \frac{d + \frac{m}{2\pi} c}{2} \quad (4.3.6)$$

A limitation in the previous approach must be taken into account, and it is the fact that the phase in equation (4.3.3) is going to be periodical, so only phases within the first period, namely the interval $[0, 2\pi)$, can be obtained without ambiguity. That is another way to state the phase limitation that arose in Section 2.4.4. That means that there will be a maximum unambiguity location range, which could be computed according to equation (4.3.5) as the distance that yields when the maximum unambiguous phase difference is used, namely, 2π :

$$m_{max} = \frac{2\pi}{f_2 - f_1} = 2\pi t_{0,max} \rightarrow t_{0,max} = \frac{1}{\Delta f} \quad (4.3.7)$$

Since the distance between frequencies in the FFT is limited by the sampling frequency and the number of samples used to compute the FFT, those parameters limit this unambiguity range. If all the neighbor bins are used to estimate the distance, then the frequency resolution will be:

$$\Delta f = \frac{f_s}{N_s} \quad (4.3.8)$$

And so, using the results in equations (4.3.5), (4.3.7) and (4.3.8):

$$t_{0,max} = \frac{1}{\Delta f} = \frac{N_s}{f_s} \rightarrow \Delta d_{max} = t_{0,max} \cdot c = \frac{N_s}{f_s} \cdot c \quad (4.3.9)$$

So, a higher number of samples in the FFT calculus will allow having a bigger unambiguity range, but that means that a higher computational cost would also be needed.

4.3.2 - ALGORITHM DESCRIPTION

The proposed algorithm, then, will be very similar to the one proposed in Section 4.2.2. The flow chart is in Figure 4.12 and the main changes are:

1. Since complex data are not needed in this method, it is possible either to use an IQ downconverter and then obtain the real data, or use a real downconverter.
2. Since there is no need to change the WiFi channels in order to do the frequency sweep, that part of the algorithm is gone.
3. The samples processing is going to be done according to the equations in this method, so the FFT of both oscilloscope channels is going to be obtained and then, the distance will be estimated using the phase information.

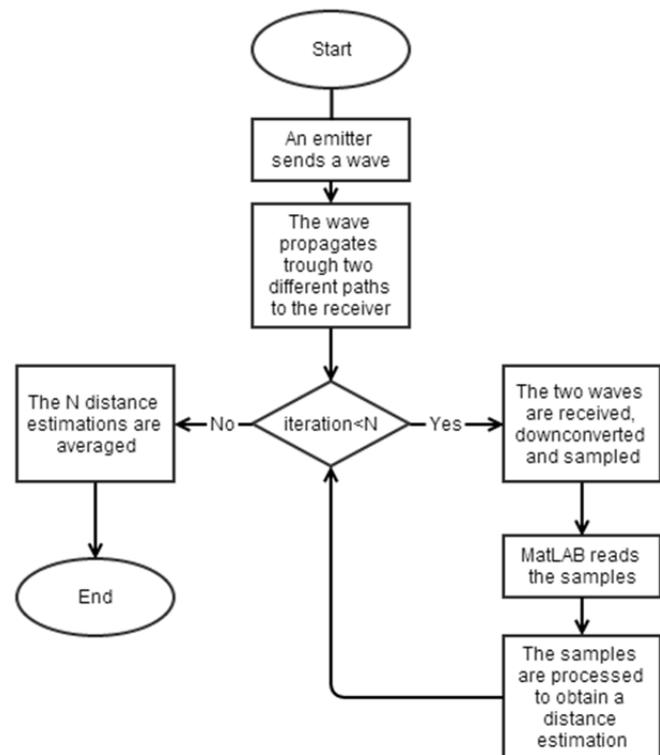


Figure 4.12 - Flow chart, FFT

4.3.3 - CABLE BASED SETUP

In order to test this approach, the setup described in Chapter 3.2 will be used here.

4.3.3.1 - PARAMETERS STUDY

In this section, some of the parameters that are going to be used in the code are studied and their influence on the location estimation is explained, so that a better understanding of them is obtained.

4.3.3.1.1 - FFT SAMPLES NEEDED

An important subject to be taken into account is the fact that the Discrete Fourier Transform is going to be used. So, the number of samples will be important, as well as the window election, since that will affect dramatically the accuracy of the obtained data. A higher number of samples will give a better frequency resolution and as it was proved in equation (4.3.9), a greater unambiguity range, but that implies higher computational costs as well. Because of that, windows are used [26] [27], since they help to reduce the spectral leakages (secondary lobes). In the algorithm implemented, the Hamming window is used, since that window gives a good tradeoff between system accuracy and leakages reduction.

In order to check how many samples are needed, a simple test will be run in order to determine empirically how it affects to the accuracy of the algorithm:

1. The sample number will be setup from 100 to 4000 samples, using increase steps of 100 samples
2. The WiFi access point will be configured to emit only in Channel 1
3. For each number of samples, the calculation algorithm will be computed 50 times, so that using the Central Limit Theorem, it can be supposed that the values will follow a normal distribution, and so, the mean and the standard deviation of the distance d_1 are computed

The results can be seen in Figure 4.13 and Figure 4.14, both for $d = 10 m$ and $d = 8 m$ (in both cases, $d_1 = 5 m$). In all plots, the estimated mean distance d_1 obtained is plot, as well as the actual d_1 distance and the 99.8% confident interval (mean plus/less three standard deviations) in the estimation obtained.

In both plots it can be seen that an accurate value is obtained from 1500 samples on (with a standard deviation of approximately 1 cm). Here, it must be observed that a bigger amount of samples does not necessarily imply a higher accuracy, since in the plots it is possible to see that there is a minimum error (bias error) which approximately constant with the number of samples, so there will be an increase in the computational cost whereas there is no noticeable effect on the algorithm location accuracy. Because of that, 1500 samples will be used in the calculations.

Also, according to equation (4.3.9), with 1500 samples and a sampling frequency $f_s = 250 MHz$, the unambiguity range in the cable ($c = 0.66 \cdot c_0$) is going to be:

$$\Delta d_{max} = \frac{N_s}{f_s} \cdot c = \frac{1500}{250 \cdot 10^6} \cdot 0.66 \cdot 3 \cdot 10^8 = 1188 m$$

So, the unambiguity range is really big and will not give any problem in this approach.

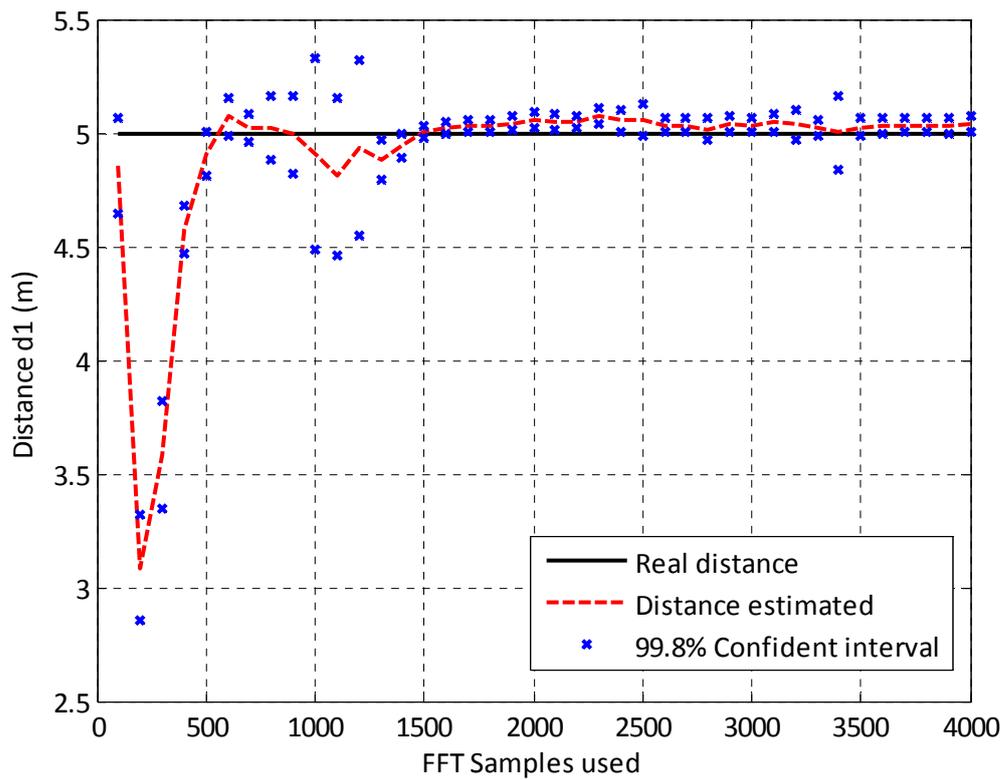


Figure 4.13 – Distance estimation as a function of FFT samples used, $d = 8\text{ m}$, cable setup

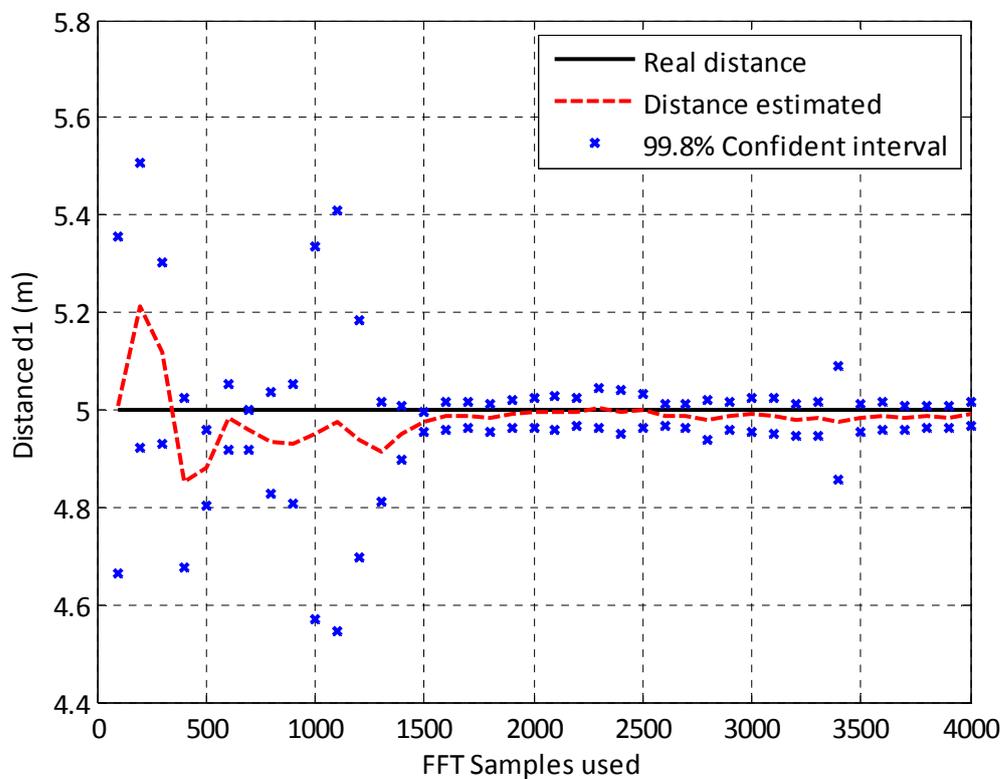


Figure 4.14 – Distance estimation as a function of FFT samples used, $d = 10\text{ m}$, cable setup

4.3.3.1.2 - FFT PHASE BANDWIDTH LIMITS

Another point to be taken into account is the fact that a WiFi Channel has a bandwidth of 20 MHz. So, once the central frequency is introduced in the algorithm, and the FFT is calculated, only some points of the FFT will be used to compute the phase difference. The question is, should all of the bandwidth be used or not?

The main advantage of using a higher bandwidth is that a lesser error will be committed since more data will be used in the fit. However, WiFi channels are overlapped, so in an environment with a high amount of WiFi emitters, this overlapping will cause that a lot of noise is present in the signals, and this noise will have a negative effect on the phase accuracy of the algorithm.

Thus, a balance is needed between the requirements to protect against fit error and other signals interferences. If the FFT of the full 20 MHz channel is computed, in the cable based setup described before (so, there will be no important interference from other WiFi channels), using the WiFi Channel 1 to emit, the result obtained can be seen in Figure 4.15 (with $d = 10\text{ m}$, so a slope $m = 0$ is to be expected)

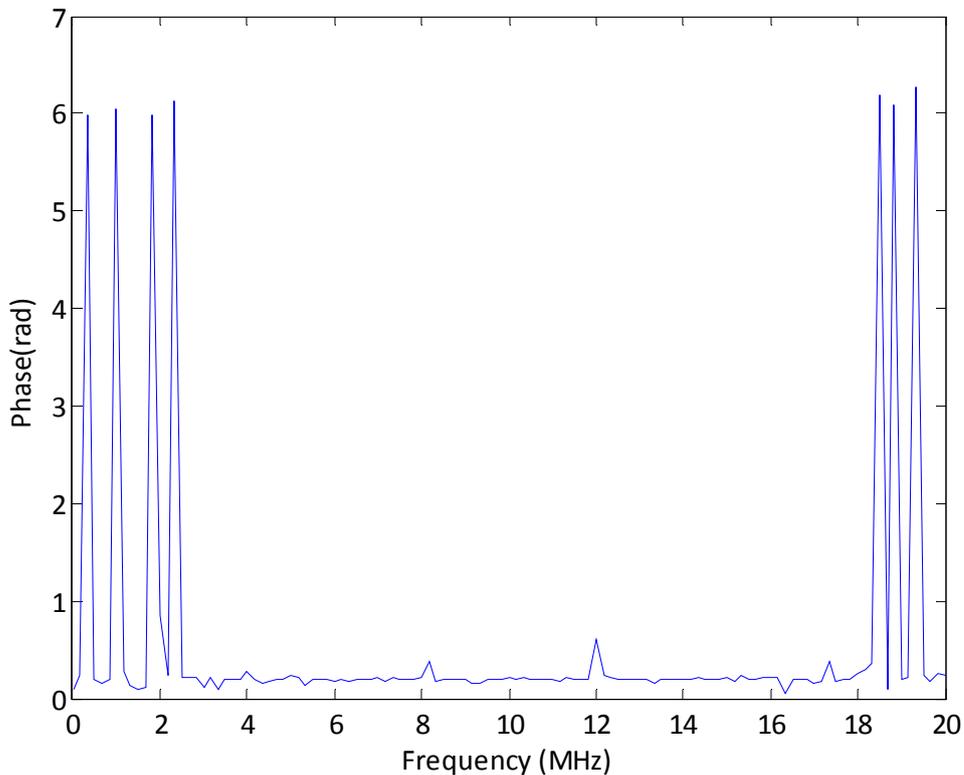


Figure 4.15 – Phase difference in WiFi Channel 1, FFT approach , cable setup

It is possible to see that even in clean environments, there is a significant error in the extremes of the channel bandwidth. This error will affect strongly the calculations of the phase difference slope and so, it is going to mean a high inaccuracy in the distance measurements. Because of that, the full 20 MHz of the signal will not be used in order to measure the phase difference, but a bandwidth of 15 MHz will be used instead, in order to reduce this error and reach a tradeoff between noise immunity and a large amount of samples to fit.

4.3.3.2 - RESULTS AND CONCLUSIONS

In the cable setup described above, the proposed algorithm will be run 10 times and then, the results will be averaged using both mean and median. Also, different WiFi channels will be used, in order to test whether the central frequency election affects or not to the measurements. In every measurement, a bandwidth of 15 MHz is going to be used, with 1500 FFT samples and a Hamming window.

As described in Appendix 1, two different versions of the slope calculations will be used, the unshifted one and the shifted one. The final measurement will be the one with a smaller standard deviation.

The algorithm will be run with two different set of cables, the first one with $d = 10\text{ m}$ and the second one with $d = 8\text{ m}$. In all cases, $d_1 = 5\text{ m}$. The results obtained are in Table 4.4.

$d\text{ (m)}$	WiFi Channel	Mean $d_1\text{ (m)}$	Absolute error (cm)	Relative error (%)	Median $d_1\text{ (m)}$	Absolute error (cm)	Relative error (%)
10	1	4.9949	0.51	0.102	4.9928	0.72	0.144
10	6	4.9841	1.59	0.318	4.9812	1.88	0.376
10	11	4.9622	3.78	0.756	4.9629	3.71	0.742
8	1	4.9862	1.38	0.276	4.9863	1.37	0.274
8	6	4.94	6	1.2	4.9414	5.86	1.172
8	11	4.7207	27.93	5.586	4.7368	26.32	5.264

Table 4.4 - Results of FFT approach, cable setup

The following conclusions are deduced from the results:

- The FFT method shows a high accuracy in the cable environment described above. With the parameters set above, it usually offers an error around 1%.
- In both distances, it is possible to observe that the accuracy gets worse if the frequency increases.
- This method is promising, since there is no need to perform a frequency sweep, with all that that means in time cost. The computations are really fast (refer to Appendix 2 for more details), so it is both cheap and cost and time effective.
- However, the fact that even in a clean environment, without multipath or signal interference, the effective bandwidth had to be reduced in order to have a clearer phase difference measure, suggests that the performance of this method is going to be affected, at least, in a LOS scenario.

4.3.4 - LOS SETUP

In this case, when the setup described in Chapter 3.3 was used to obtain a location estimation, the results were severely incoherent. Really big errors were obtained and the exact cause of them needs further research to be clarified.

4.4 - METHOD 3: SIGNAL POWER

4.4.1 - THEORETICAL EXPLANATION

In Chapter 2.4.1, it was already shown that the 802.11b signal could be studied as a modularly angulated signal, which means that the modulation schema does not introduce any changes in the signal amplitude. Thus, this amplitude will only be modified due to propagation loss and so, a method to locate based on signal attenuation can be used, not only with a WiFi signal, but with any angularly modulated signals, as it can be deduced from equations (2.4.1) and (2.4.2).

Let us suppose a random signal modularly angulated, propagating through two different lengths channels, so that at the end of those channels it is possible to obtain the propagated signal and its delayed copy, let them be called $x_1(t)$ and $x_2(t)$. If both channels are LTI (Linear and Time invariant) and if the delay between both signals is t_0 , then it is possible to state that $x_2(t) = A \cdot x_1(t - t_0)$, where A is the amplitude difference due to the different path propagations (supposing that the path covered by the signal received in channel 1 is shorter than the signal in channel 2, then $|A| < 1$)

It is possible to measure the power of each signal in the receiver, between the time instants t_1 and t_2 , and if it is supposed that the amplitude difference A is time independent (condition satisfied by angularly modulated signals in ideal conditions), the measured power will be:

$$P_1 = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} |x_1(t)|^2 dt \quad (4.4.1)$$

$$P_2 = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} |x_2(t)|^2 dt = \frac{|A|^2}{t_2 - t_1} \int_{t_1}^{t_2} |x_1(t - t_0)|^2 dt \quad (4.4.2)$$

And if the delay between signals is negligible compared to the observation period ($t_2 - t_1 \gg t_0$), then:

$$\int_{t_1}^{t_2} |x_1(t)|^2 dt \approx \int_{t_1}^{t_2} |x_1(t - t_0)|^2 dt \quad (4.4.3)$$

Computing the power relation between signals, it will be obtained that:

$$\frac{P_1}{P_2} = \frac{\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} |x_1(t)|^2 dt}{\frac{|A|^2}{t_2 - t_1} \int_{t_1}^{t_2} |x_1(t - t_0)|^2 dt} \approx \frac{1}{|A|^2} \quad (4.4.4)$$

So, the power relation between channels depends only on the square of the tension relation between the signals amplitudes. And this relation will depend, in the end, of the transmission medium, so it is possible to state that, if both signals propagate trough exactly the same medium, but through different distances, then A will depend on the attenuation constant of the medium α , which in a cable propagating medium is:

$$|A| = 10^{-\alpha \frac{\Delta d}{10}} \quad (4.4.5)$$

Where α units are dB/m and Δd is the distance difference between transmission paths, so $\Delta d = d_1 - d_2$. Operating:

$$\frac{P_1}{P_2} = \frac{1}{|A|^2} = 10^{2 \cdot \alpha \frac{\Delta d}{10}} \rightarrow \Delta d = \frac{5}{\alpha} \cdot \log\left(\frac{P_1}{P_2}\right) \quad (4.4.6)$$

And considering that $d = d_1 + d_2$, d_1 can be obtained:

$$d_1 = \frac{d + \Delta d}{2} = \frac{d + \frac{5}{\alpha} \cdot \log\left(\frac{P_1}{P_2}\right)}{2} \quad (4.4.7)$$

This algorithm can be applied to any modulation schema that is angular, since there will be no additional amplitude variations due to the modulation used. On an amplitude based schema, it could be applied as well, but the observation period in the integral must be much greater than the amplitude variation due to the modulation schema, $t_2 - t_1 \gg T_{mod}$, so that its influence can be considered negligible.

4.4.2 - ALGORITHM DESCRIPTION

The proposed algorithm, then, will be very similar to the one proposed in Section 4.3.2. The main changes are:

1. In this case, the total power of the signals is needed. Thus, the only calculations computed are the power relation between channels.
2. However, especial attention must be put to the fact that the equation (4.4.7) has been deducted using the attenuation expression in (4.4.5). If the attenuation in the channel follows a different model (what happens, for example, in a wireless channel, as can be seen in Chapter 4.4.4.1.1), then the calculations to be done in order to obtain the distance estimation will be different.

4.4.3 - CABLE-BASED SETUP

In order to test this approach, the setup described in Chapter 3.2 will be used here.

4.4.3.1 - SYSTEM OFFSET

A main supposition in the theoretical explanation was that the amplitude difference between the two signals would be caused exclusively by the additional path which one of them goes through. However, that supposition does not fit to the real world: the system will not be completely symmetrical, so there will be an offset in the amplitude difference. Thus, there will be an error due to imperfections in the system.

A priori, it seems not to be a bad idea to suppose that this offset is going to be frequency dependent, since the attenuation of the transmission lines, connectors and other passive components usually depends on the working frequency. But in a determinate frequency, the offset should be constant.

Assuming that, the offset value can be determined and corrected in the calculation process. A possible way to determine this offset is to use two equal length cable and compute the amplitude difference (since in this case, according to the previous explanation, the amplitude difference should be 0, the value obtained will correspond with the offset in the concrete frequency of work). Computing 50 times the attenuation following the described schema, the result obtained in WiFi Channels 1, 6 and 11 are in Table 4.5, as well as the offset value chosen.

However, this procedure does have some weak points. First of all, the supposition that attenuation offset only depends on frequency might not to be really accurate in real life environments, since other variables such as time or temperature could have an influence on that value. Secondly, the supposition that all cables will have the same attenuation is an approximation, since two cables will not be exactly identical and so, their attenuations will vary. And finally, the supposition that all cable

connections will attenuate the same might not be very accurate: each cable has connectors and the attenuation difference introduced by them is included in the offset value calculated, but if the connectors vary, so will do their attenuation and thus, the offset will vary as well.

WiFi Channel	Attenuation mean (dB)	Attenuation median (dB)	Attenuation standard deviation (dB)	Offset taken for calculus
1	2.6569	2.5957	0.1016	2.65
6	2.681	2.6977	0.1086	2.7
11	2.8164	2.8754	0.075	2.8

Table 4.5 – System attenuation offsets empirically obtained, cable setup

4.4.3.2 - RESULTS AND CONCLUSIONS

In the cable setup described above, the proposed algorithm will be run 10 times and then the results will be averaged using both mean and median. Also, different WiFi channels will be used (1, 6 and 11), in order to test the effects that the central frequency election has on the measurements, and in these channels, the attenuation offset obtained in the section before will be used.

The algorithm will be run with two different set of cables, the first one with $d = 10\text{ m}$ and the second one with $d = 8\text{ m}$. In both cases, $d_1 = 5\text{ m}$. The results obtained are in Table 4.6.

$d\text{ (m)}$	WiFi Channel	Offset	Mean $d_1\text{ (m)}$	Absolute error (cm)	Relative error (%)	Median $d_1\text{ (m)}$	Absolute error (cm)	Relative error (%)
10	1	2.65	5.0964	9.64	1.928	5.1151	11.51	2.302
10	6	2.7	5.0182	1.82	0.364	5.0193	1.93	0.386
10	11	2.8	5.0389	3.89	0.778	5.0327	3.27	0.654
8	1	2.65	5.3347	33.47	6.694	5.2215	22.15	4.43
8	6	2.7	5.5046	50.46	10.092	5.5067	50.67	10.134
8	11	2.8	5.7312	73.12	14.624	5.8234	82.34	16.468

Table 4.6 - Results of power approach, cable setup

The following conclusions are deduced from the results:

- The error in this method is larger than in the methods proposed before, namely, in the two methods based on phase measurements
- Also, the offset may be a problem, since the error is significantly bigger in the cables which were not used to compute it. Although the results show that this is not a so accurate way to measure,

if the differences were due to the fact that the cable and/or connectors are not the same, then in air propagation conditions it should behave better, since all those factors will lose importance because the main medium will be air, which can be supposed to be Linear, Homogeny and Isotropous with little error.

- However, moving to air propagation conditions will also mean that multipath propagation and fast fading might appear, so the accuracy could be significantly affected by these new conditions

4.4.4 - LOS SETUP

The setup described in Chapter 3.3 is going to be used in order to test the proposed approach in a wireless environment

4.4.4.1 - PARAMETERS STUDY

In this section, some of the parameters that are going to be used in the code are studied and their influence on the location estimation is explained, so that a better understanding of them is obtained.

4.4.4.1.1 - ATENUATION CONSTANT

In the LOS scenario, the power attenuation is not easy to calculate. In [28] some models are described and compared for a scenario based on WLAN 802.11b radio propagation, and a conclusion is that the attenuation constant will depend on quite a lot of factors and the model used. In this setup, the simplest approach will be used, even knowing that it does not have a high accuracy, as the paper referred before shows, but for testing purposes might be enough. That model is called linear OS (One Slope) and it is used in some of the best-known propagating models, like free space, plane Earth, Okumura-Hata and COST-231. The formula is the next:

$$L (dB) = L_0 + 10 \cdot n \cdot \log(d) \quad (4.4.8)$$

Where L is the total attenuation, dependent on the distance (d), the attenuation constant (n) and the attenuation at 1 meter (L_0). Those parameters can be obtained empirically or a model to estimate them can be used. In this case, an empirical estimation will be used.

To get that estimation, an antenna will be placed at different distances from the emitter, always within LOS conditions, and 50 measurement of the received signal power will be computed at each point, since at the receiver, the power is:

$$P_{rx}(dB) = P_{tx} + G_{tx} + G_{rx} - L = P_{tx} + G_{tx} + G_{rx} - L_0 - 10 \cdot n \cdot \log(d)$$

$$P_{rx}(dB) = B - 10 \cdot n \cdot \log(d) \quad (4.4.9)$$

Where B is constant in terms of distance. Those 50 measurements will be averaged using the mean and those values will be used to fit the model proposed, using a fit in the least-square sense and considering that $x = 10 \cdot \log d$ to compute a linear fit whose expression is:

$$P_{rx}(dB) = B - n \cdot x \quad (4.4.10)$$

Once this is done, the attenuation constant is equal to the slope obtained in the fit. To measure this, the LOS setup described before is used: the access point is placed in the position $d = 0$ and an antenna is placed at different distances from it, and the power received in each position is used for calculus purposes. The results obtained are plot in Figure 4.16 and the attenuation constant α obtained using the empirical fit in Table 4.7.

WiFi Channel	Attenuation constant empirical (dB)	Attenuation constant for calculus (dB)
1	0.9394	0.94
6	0.6975	0.7
11	0.7073	0.7

Table 4.7- Attenuation constant offsets empirically obtained, LOS setup

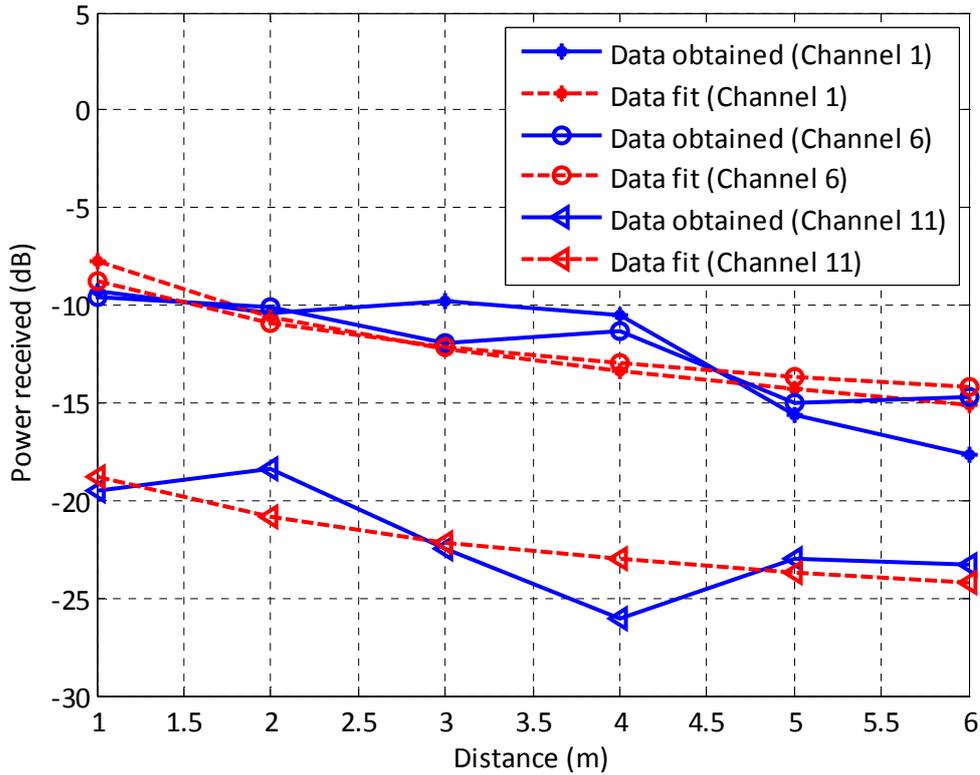


Figure 4.16 – Attenuation constant fit obtained, LOS setup

Finally, since the attenuation expression changes between the expression in equation (4.4.5) and the one in equation (4.4.8), then the distance calculation formula will be different, since in one the attenuation is proportional to the distance and in other, to the distance logarithm. According to equation (4.4.9), the power that each signal carries is, in dB:

$$P_{rx,1}(dB) = B - 10 \cdot n \cdot \log(d_1)$$

$$P_{rx,2}(dB) = B - 10 \cdot n \cdot \log(d_2)$$

If signal powers, in dB, are subtracted, the attenuation will be obtained, but this attenuation now does not follow the expression in equation (4.4.5), but in (4.4.8) and (4.4.9), so:

$$P_{rx,2}(dB) - P_{rx,1}(dB) = A(dB) = 10 \cdot n \cdot \log\left(\frac{d_1}{d_2}\right) \rightarrow \frac{d_1}{d_2} = 10^{\frac{A(dB)}{10 \cdot n}} \quad (4.4.11)$$

And if $d = d_1 + d_2$, then operating:

$$d_1 = d \cdot \frac{10^{\frac{A(dB)}{10 \cdot n}}}{1 + 10^{\frac{A(dB)}{10 \cdot n}}} \quad (4.4.12)$$

4.4.4.1.2 - SYSTEM OFFSET

The offset measurement procedure described in Section 4.4.3.1 is used, just that in this case, since antennas are being used, they are placed together at any distance from the signal generator (however, since multipath propagation, near-field antenna problems and short-term fading, it is preferable placing the antennas in a LOS scenario, close to the access point but not so much that the near-field effects could affect the measurements)

The antennas were placed at a distance $d_1 = d_2 = 1 \text{ m}$ from the access point. The empirical offset obtained for 50 iterations in the WiFi channels 1, 6 and 11 using the procedure described in that section are in Table 4.8. The standard deviation of the data in the different WiFi Channels show that while in channels 1 and 6, the attenuation difference remains constant, in channel 11 that does not happen, thus, the measures in channel 11 will have a greater error because the offset measure has a greater standard deviation.

WiFi Channel	Attenuation mean (dB)	Attenuation median (dB)	Attenuation standard deviation (dB)	Offset taken for calculus (dB)
1	0.7431	0.7434	0.0262	0.75
6	0.5093	0.5079	0.0143	0.5
11	0.6559	0.7181	0.2419	0.65

Table 4.8 - System attenuation offsets empirically obtained, LOS setup

4.4.4.2 - RESULTS AND CONCLUSIONS

With the offset and attenuation data obtained in the sections before, the algorithm to estimate the distance difference using signal power difference will be run in the LOS setup described, where $d = 6 \text{ m}$ and in each position, 50 measurements will be taken. Then, the distances obtained in each measurement will be obtained and they will be averaged using the mean to plot the distance estimated by the algorithm in each point.

The CDF obtained in each channel, using 350 measurements in each of them, are in Figure 4.17. If the offset is ignored (so, if the offset adjustment is not taken into account), the results obtained are found in Figure 4.18.

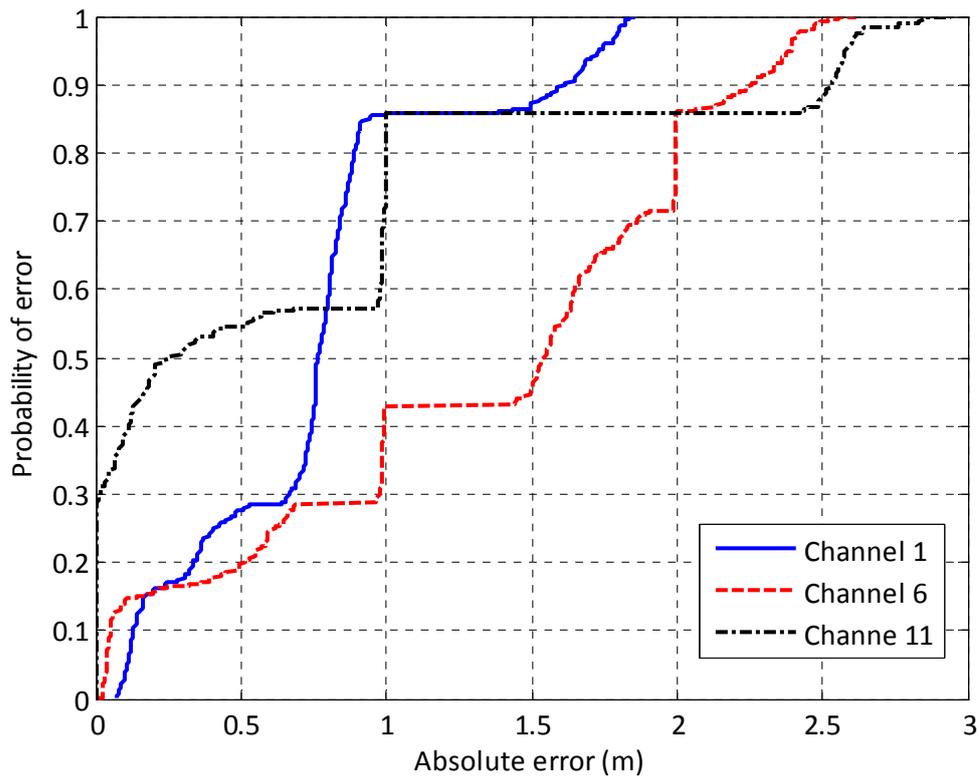


Figure 4.17 – CDF of the absolute error, WiFi Channels 1, 6 and 11, using attenuation offset, LOS setup

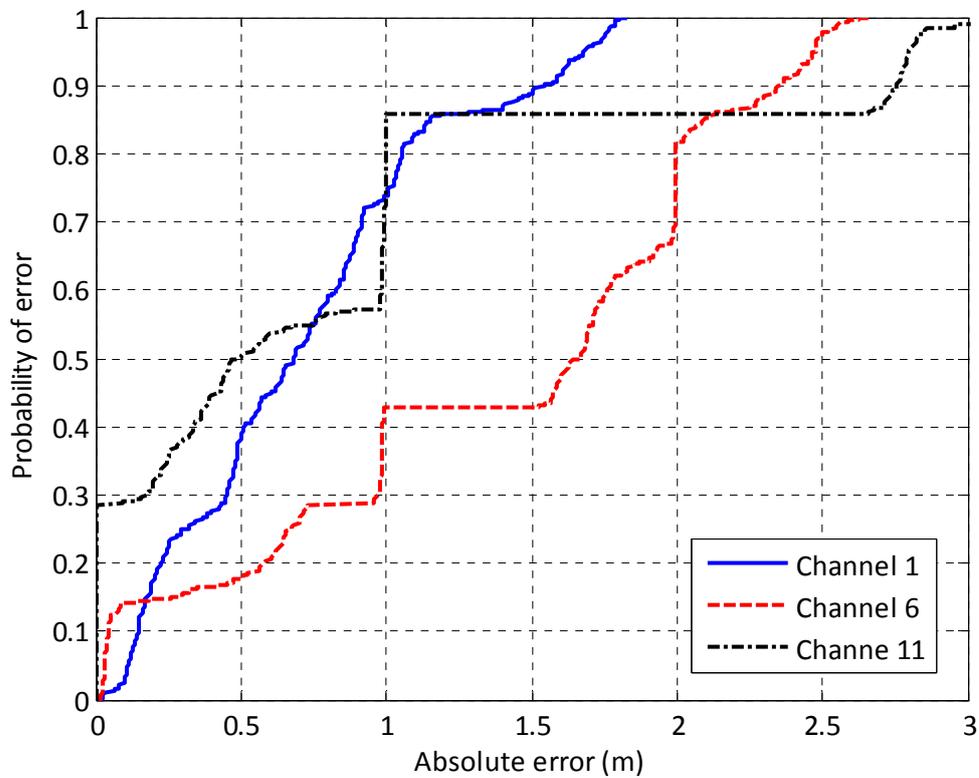


Figure 4.18 – CDF of the absolute error, WiFi Channels 1, 6 and 11, without using attenuation offset, LOS setup

With those results, the following conclusions are reached:

- The results obtained with and without offset are virtually identical. That means, that the offset measurement process could be avoided, so the method becomes even simpler at a non-significant cost in terms of location accuracy.
- This method, with the attenuation empirically obtained as it was described before and without offset, shows to be accurate. Namely, an 80% of the samples are around 1 m absolute error in Channels 1 and 11, and 2 m in Channel 6. However, there are some strong jumps present, thus, these accuracies could vary strongly, since jumps are present near the 80% accuracy in channels 6 and 11, which is the criterion chosen to determine accuracy.
- The best results correspond to channel 1. It could be due to a better adjustment of the parameters, since seems like there is no frequency dependent behavior on the accuracy (the worst channel is the number 6, which is the central one).
- Independently of the channel used, this method can be used to achieve a rough estimation of the position of the emitter. Then, another step could be using another method, so that a finer location is achieved.
- Finally, even though the attenuation model used was quite basic, the total absolute error of the system is not too high. That means that this simple approach is enough to get a non-complex and quite fast to compute location estimation, with a more than acceptable accuracy.

4.5 - METHOD 4: CROSS CORRELATION

In this approach, the properties of the Barker Code will be used in order to estimate a location using cross correlation in the IF band. Several methods will be proposed, depending on the setup parameters used.

4.5.1 - THEORETICAL EXPLANATION

Let us suppose that the Barker Code signal described in equation (2.4.5), has propagated through two different lengths paths, so that at the end of these paths it is possible to obtain the propagated signal and its delayed copy, let them be called $x_1(t)$ and $x_2(t)$, so that if the time delay between channels is t_0 . This time delay depends on the propagating speed of the wave in the medium and the distance difference between the two propagating channels, namely:

$$t_0 = \frac{\Delta d}{c} \quad (4.5.1)$$

In the receiver, in RF frequency, the signals will be, according to (2.4.5):

$$x_{1,rf} = B(t) \cdot \cos\left(2\pi f_c t + \frac{2\pi f_c}{c} d_1 + \varphi_1\right) \quad (4.5.2)$$

$$\begin{aligned} x_{2,rf} &= x_{1,rf}(t - t_0) = B(t - t_0) \cdot \cos\left(2\pi f_c (t - t_0) + \frac{2\pi f_c}{c} d_1 + \varphi_1\right) = \\ &= B(t - t_0) \cdot \cos\left(2\pi f_c t - \theta + \frac{2\pi f_c}{c} d_1 + \varphi_1\right) = \end{aligned} \quad (4.5.3)$$

Where f_{rf} is the RF frequency, φ_1 is the phase term of the signal, $\theta = 2\pi f_c t_0$ is the phase term due to the path difference and $B(t)$ is the Barker Code, which according to [18] and [19] is a known square pulse train signal. If those signals are downconverted, filtered using a Low Pass filter and normalized, the Intermediate Signals obtained will be, according to (2.4.7):

$$x_{1,if} = B(t) \cdot \cos(2\pi(f_c - f_{ol})t + \varphi_2) \quad (4.5.4)$$

$$x_{2,if} = x_{1,if}(t - t_0) = B(t - t_0) \cdot \cos(2\pi(f_c - f_{ol}) \cdot t - \theta + \varphi_2) \quad (4.5.5)$$

In those equations, f_{ol} is the local oscillator frequency, φ_2 is the phase term after the downconverter block (which is going to be the same for both signals, if the two downconverters are exactly the same, and it includes the phase term that depends on d_1) and $f_c - f_{ol} = f_{if}$ is the intermediate frequency chosen.

If the signals are sampled, there are various possibilities to process them depending on the intermediate frequency chosen, so that the delay t_0 is obtained:

- a) The first case occurs when $f_c = f_{ol} \rightarrow f_{if} = 0$, operating, the signals received are:

$$x_{1,if} = B(t) \cdot \cos(\varphi_2) \quad (4.5.6)$$

$$x_{2,if} = B(t - t_0) \cdot \cos(\varphi_2 - \theta) \quad (4.5.7)$$

In this concrete case, in which both signals are in the baseband, the delay between signals could be computed using the cross correlation. If it were computed, this function would have an absolute maximum in $\tau = t_0$.

However, in a discrete system, the time differences τ will be discrete and dependent on the sampling frequency, so:

$$\tau = \frac{n_s}{f_s} \quad (4.5.8)$$

Where n_s is the number of samples. Once the difference in samples is obtained from the cross correlation maximum, the distance difference will be:

$$\Delta d = t_0 \cdot c = \frac{n_s}{f_s} \cdot c \quad (4.5.9)$$

According to expressions (4.5.6) and (4.5.7), the amplitude of the signals depends on the phase terms. That means that there will be some values for which they will be zero and so, there will be no signal in the receptor. Thus, especial care must be taken in order to ensure that the conditions leading to this situation are not satisfied.

b) The second case occurs when $f_c \neq f_{ol} \rightarrow f_{if} \neq 0$ and so, the received signals will be:

$$x_{1,if} = B(t) \cdot \cos(2\pi f_{if} t + \varphi_2) \quad (4.5.10)$$

$$x_{2,if} = B(t - t_0) \cdot \cos(2\pi f_{if} t - \theta + \varphi_2) \quad (4.5.11)$$

In this case, the information of the delay appears both in the Barker Code and the modulated signal. Theoretically, it should be possible just proceeding as in the previous case and compute the cross correlation, look for the maxima in the signal and extract the distance difference, as it was done in the previous case. However, in this case the cross correlation will be different, since now the intermediate frequencies and θ will have an impact on the signal.

A problem that appears is that the phase term θ will be ambiguous depending on the distance Δd . Since it appears in the argument of a sinusoid, it will be possible to extract its value without unambiguity in the range $[0 \ 2\pi)$. And so, the maximum value permitted for θ will be:

$$\theta = 2\pi$$

Using the equation (4.5.1) and the definition of θ , it is possible to calculate the maximum unambiguity range as a function of distance difference:

$$\theta = 2\pi = 2\pi f_c t_0 = 2\pi f_c \frac{\Delta d}{c} \rightarrow \Delta d_{max} = \frac{c}{f_c} = \lambda_c \quad (4.5.12)$$

So, if the phase term is used to locate, there will be an unambiguity range that will depend only on the wavelength on the propagating medium. Subsequently, in the 802.11 b signals case, the unambiguity range will be of around 12 cm.

Since two terms appear on the signals, that contain location information, the distance estimation in the intermediate frequency will depend on which one of the terms above is chosen to locate:

- I. If the term in the Barker Code is used to locate, then the proceeding and equations would be the ones described in the case where $f_{if} = 0$. This method will be called Barker – IF, for referencing purposes.

- II. If the phase term is chosen to locate, there will be a limitation in the unambiguity range as it has been shown in equation (4.5.12). However, there is a strong advantage and is that the resolution of this method could be adjusted.

Let us suppose that this angle is used to detect. In intermediate frequency, it can be supposed that it is a constant phase term, so it could be computed by measuring the phase difference between the two waves which enter the receptor, namely:

$$\theta = 2\pi f_{if} t_0 \quad (4.5.13)$$

Where t_0 is the time difference between both waves. Since the system is going to be digital, using the equations (4.5.8) and (4.5.9), as well as the definition of θ , it turns out that:

$$\theta = 2\pi f_{if} t_0 = 2\pi f_{if} \frac{n_s}{f_s} = 2\pi f_c \frac{\Delta d}{c} \rightarrow \Delta d = \frac{c}{f_c} \cdot f_{if} \cdot \frac{n_s}{f_s} \quad (4.5.14)$$

In this equation, both f_{if} and f_s can be chosen, so the resolution of the system can be changed by using different values for these parameters.

For referencing purposes, this method will be called sinusoid-IF method.

- III. A third solution would be using all the information, the one contained in the sinusoid and in the Barker Code. In this case, which is the one in which the maximum amount of information is used from the signals, more information will be obtained and that can help improving the location accuracy. For referencing purposes, this method will be called dual-IF method.

Finally, it must be noted that the angle θ could be nearly constant in the band of interest, due to the fact that the only parameter that varies in the bandwidth of the signal is the frequency; but since the carrier frequency is much higher than the signal bandwidth, this angle caused by propagation could, under some conditions, be considered constant. So, in the concrete case of WiFi Channel 1, the angles in the lowest, central and highest frequencies in the bandwidth would be:

$$\theta_{low} = \frac{2\pi f_c}{c} \Delta d = \frac{2\pi}{c} \Delta d \cdot 2.402 \cdot 10^9 = 50.307 \text{ rad/m}$$

$$\theta_{f_c} = \frac{2\pi f_c}{c} \Delta d = \frac{2\pi}{c} \Delta d \cdot 2.412 \cdot 10^9 = 50.517 \text{ rad/m}$$

$$\theta_{high} = \frac{2\pi f_c}{c} \Delta d = \frac{2\pi}{c} \Delta d \cdot 2.422 \cdot 10^9 = 50.726 \text{ rad/m}$$

The absolute error committed would be, at maximum, 0.21 rad/m . That means that depending on the distance separation Δd , this approximation will be a good one or a poor one. For example, in the sinusoid-IF method, it was supposed that the angle θ was constant. Since the unambiguity range, and so the maximum distance in that approach is of 12 cm , the maximum error committed would be inferior to 0.03 rad , so it is a good approximation in this concrete case and thus, the results are valid.

4.5.2 - INFLUENCE OF THE INTERMEDIATE FREQUENCY

A really important parameter, that is going to determine the performance characteristics of the system, is the intermediate frequency chosen. In fact, if a dual-IF method wants to be implemented, this parameter plays a major role in the way this method could be implemented.

There are two possible ways to study the influence of this parameter: either the cross correlation function is obtained from equations (4.5.4) and (4.5.5) or the influence of this parameter is experimentally obtained. The second option will be used in this case.

To study its influence, a setup like the ones described in Chapter 3.3 is used. The intermediate frequency in the oscilloscope is changed, and according to that change, the signals received in both channels of the oscilloscope are plot (in all cases, $f_s = 250 \text{ MHz}$ and the WiFi Channel 1 are used). The results obtained are in Figure 4.19, and it is possible to see that the Intermediate frequency will modulate the power envelope of the signal, in a similar way that the analogical double side band modulation works.

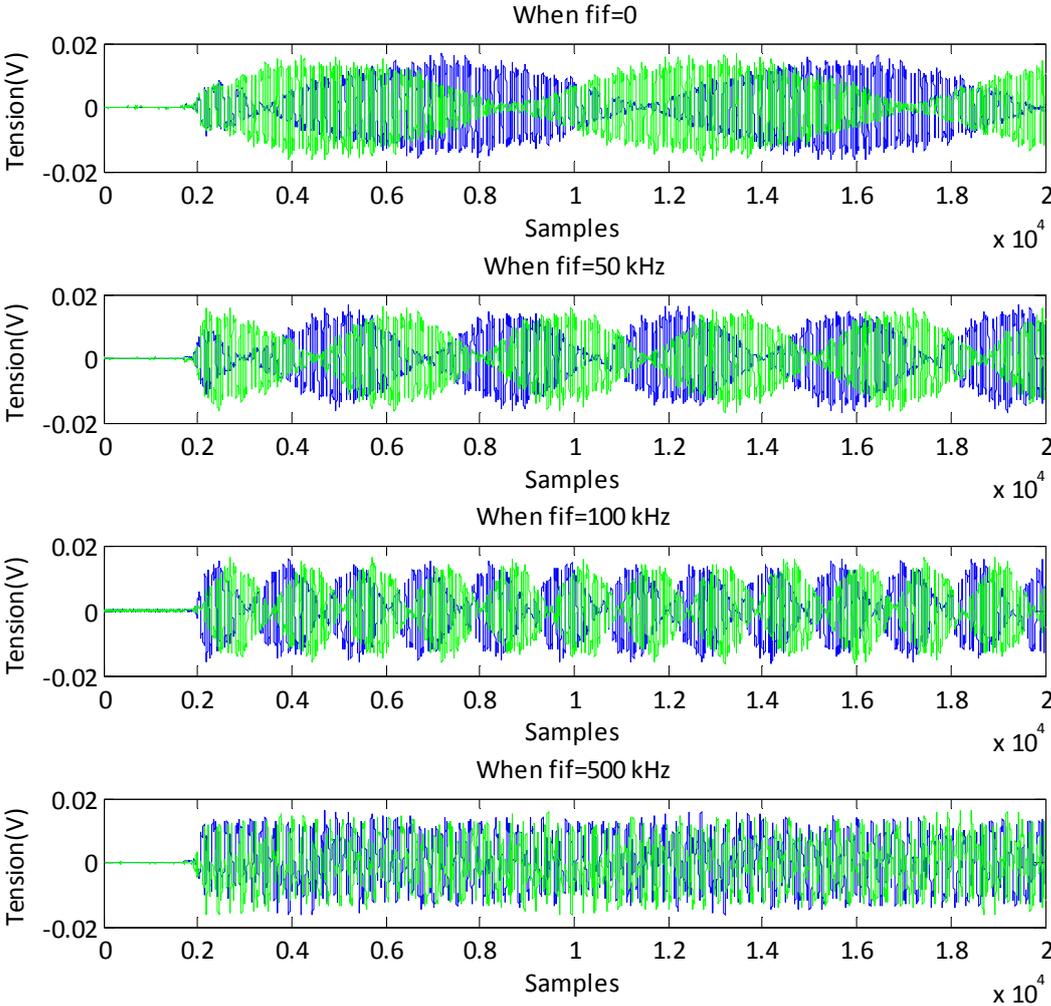


Figure 4.19 – Signals received as a function of sampling frequency

And so, it is possible to observe that:

- The frequency jitter between the oscilloscope oscillator and the signal access point oscillator is having an influence in the signal. In the case that $f_{if} = 0$, the signal envelope should be constant but it is not, and from its period it is possible to obtain the jitter in this concrete case, which is of approximately $1.8 \cdot 10^4$ samples. Since $f_s = 250 \text{ MHz}$, that means that the frequency shift between oscillators is:

$$f_{jitter} = \frac{f_s}{1.8 \cdot 10^4} \approx 14 \text{ kHz}$$

- Since the intermediate frequency appears on the signal envelope, the phase delay θ will appear there as well. Subsequently, in the case in which the envelope is clear enough to be detected using an envelope detector, the phase delay can be extracted from the time signal, and so an accurate measure according to the sinusoid-IF method will be possible to obtain. To fulfill that condition, a low intermediate frequency must be used.

Also, if a low intermediate frequency is used, the influence of the envelope on the signal will be inappreciable for time intervals much smaller than the period of the Intermediate Frequency, and that means that the equations (4.5.8) and (4.5.9) could be applied to a piece of the signal short enough.

This approach described above, using a low intermediate frequency and locating using both the information contained in the time signals and the cross correlation will be implemented and tested. For referencing purposes, it will be called near-Zero-IF method, and it is just the dual-IF method using a low IF.

In the case that a high Intermediate Frequency is used, the term dominant in the cross correlation will be the phase term one. That means, that the cross correlation will vary periodically with λ_c and its envelope will vary with t_0 , as it can be seen in Figure 4.20 and Figure 4.21, where both cases are plot, and it can be seen that the cross correlation envelope is always a sinc function, but it is dominant only in the case of low intermediate frequency.

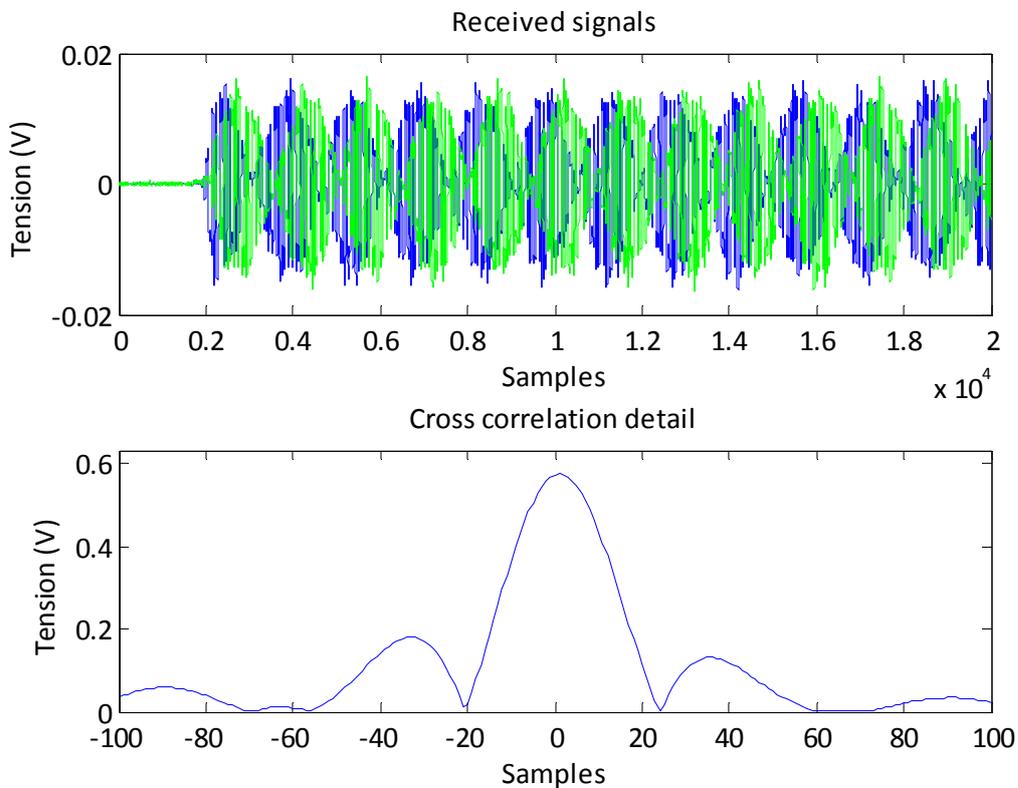


Figure 4.20– Signals received and cross correlation detail when $f_{if} = 100 \text{ kHz}$

4.5.3 - PROPOSED METHODS

This method gives a lot of options and possible combinations in order to locate. Two of them will be tested:

- 1) Firstly, a high intermediate frequency approach will be tested in order to check whether this method gives the high resolution and accuracy that, theoretically, it should give, or not. In this case, it should be possible to use an envelope detector in the cross correlation function in order to detect the Barker-IF location, but that will not be implemented.
- 2) Secondly, a near – Zero – IF system will be tested. Its advantages have already been described in the section before, and it will be put to the test to check whether the theoretical expectations are or not fulfilled.

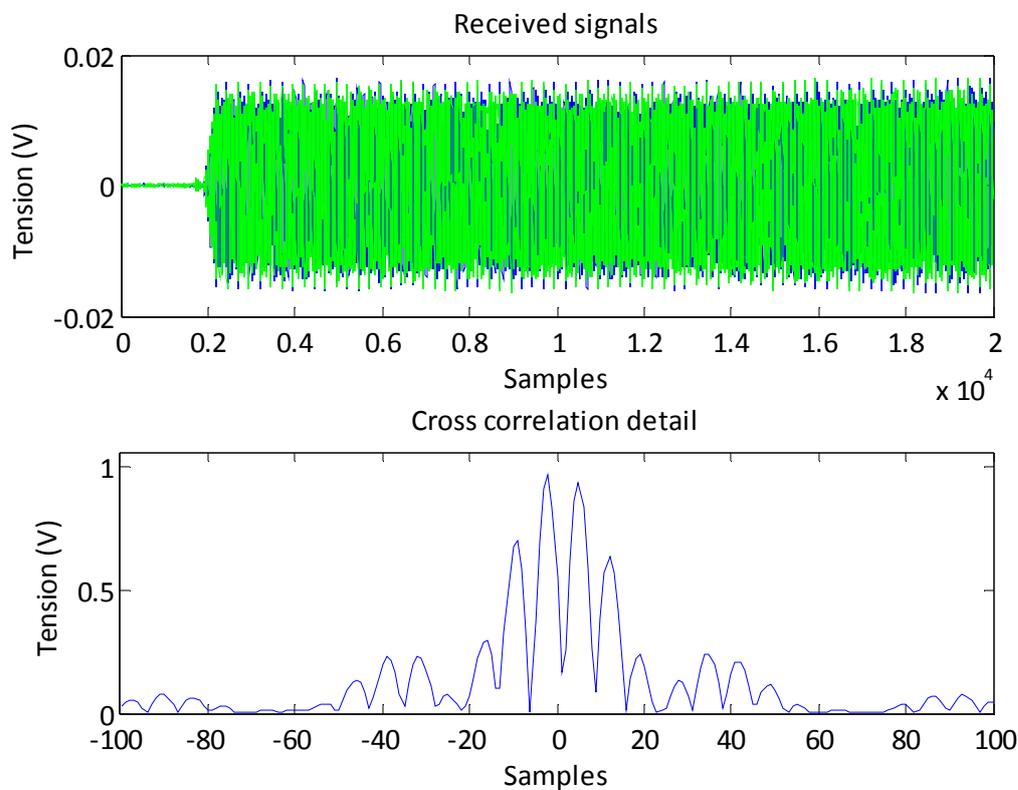


Figure 4.21– Signals received and cross correlation detail when $f_{if} = 17 \text{ MHz}$

4.5.4 - HIGH IF APPROACH

The setup described in Chapter 3.4 is going to be used in order to test the proposed approach in a wireless environment, instead of the one in Chapter 3.3, because the resolution and unambiguity range of this method makes this setup more convenient.

4.5.4.1 - ALGORITHM DESCRIPTION

The proposed algorithm, then, will be very similar to the one proposed in Section 4.3.2. The only changes will be in the samples processing part of the code, whose calculations will be based on computing the cross correlation of the IF signals, detecting its maximum and estimating the distance according to the formula (4.5.14).

4.5.4.2 - ALTERNATIVE CALCULUS

Since this method is based on the measure of a shift between two signals, different ways to process the samples could be used to compute this shift. An especially interesting set of methods to be used are those based on the estimation of the phase difference on a signal, like those based on zero-crossing, filtering or statistical analysis, a brief description of them is found in [29].

There are plenty of other methods to compute this shift, so any of them could be implemented in order to obtain a better tradeoff between simplicity, resolution and speed. In this case, the only implementation done has been the one based on cross correlation, for testing purposes; but if this method is chosen to locate it could be interesting exploring those other alternative ways to do the calculations. However, it is necessary to point out the ambiguity range of the WLAN signal, due to its periodicity, will be a problem independently of the method used to measure the phase shift.

4.5.4.3 - PARAMETERS STUDY

In this section, some of the parameters that are going to be used in the code are studied and their influence on the location estimation is explained, so that a better understanding of them is obtained.

4.5.4.3.1 - RESOLUTION OF THE SYSTEM

As it was demonstrated in equation (4.5.14), the election of the intermediate frequency and the sampling frequency will have an immediate effect on the resolution of the system. In the described setup, with $f_{ol} = 2.395 \text{ GHz}$ and $f_{s,if} = 250 \text{ MHz}$, the theoretical resolution of the system will be determined by the minimal distance difference that can be calculated; in other words, the resolution of 1 sample:

$$Resolution = \frac{1}{f_{s,if}} \cdot c \cdot \frac{f_{if}}{f_c} \quad (4.5.15)$$

WiFi Channel	f_c (GHz)	Resolution (mm)
1	2.412	8.5
6	2.437	20.7
11	2.462	32.6

Table 4.9- Theoretical resolution as a function of WiFi Channels, high-IF approach

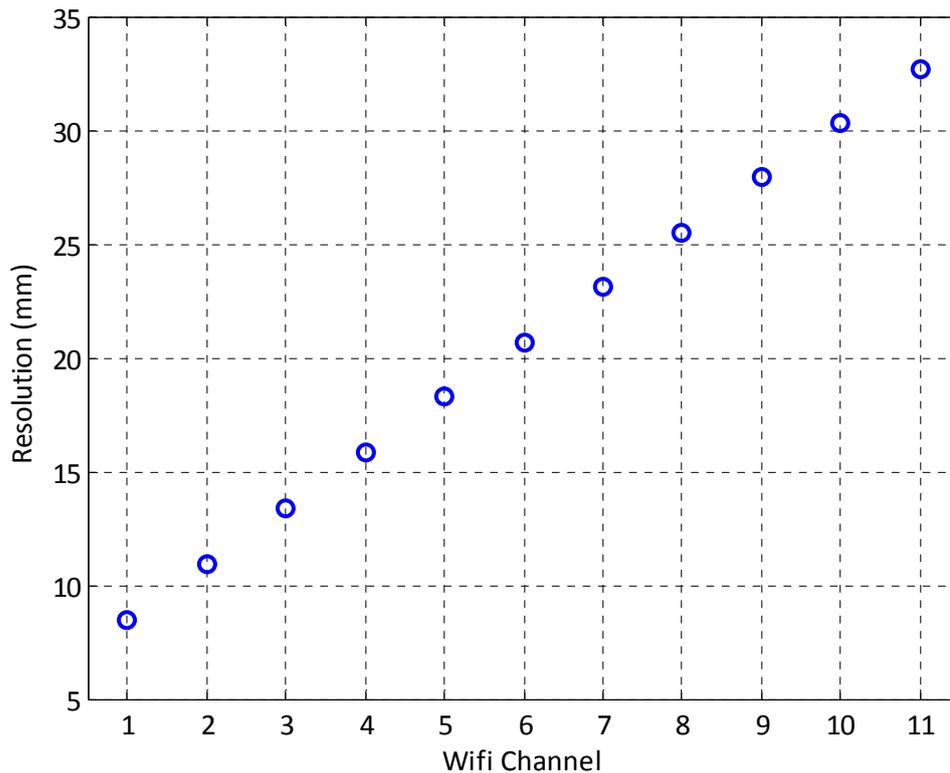


Figure 4.22 – Theoretical resolution as a function of WiFi Channels, high IF approach

From Figure 4.22 and the formula (4.5.15), it is possible to see that the resolution will be strongly affected by the channel election and the intermediate frequency used. Since a low carrier frequency yields better resolution, the algorithm will only be tested in WiFi Channel 1.

Also, in order to improve the robustness of the system against noise and interference, it could be possible to filter the signal before the calculation process, so that only the bandwidth of the WiFi channel of interest is used to compute the distance. However, in the algorithm used this filtering process will not be used, since the noise present in the described environment does not significantly affect the results and thus, it is not really necessary.

4.5.4.3.2 - EMPIRICAL UNAMBIGUITY RANGE

As it was proved in (4.5.12), the unambiguity range expected is the wavelength of the signal in the transmitting media used, namely, air, which is:

$$\Delta d_{max} = \lambda_c = \frac{c}{f_c}$$

It is possible to see that the unambiguity range is going to depend on the carrier frequency, or in other words, in the WiFi Channel chosen. It can be checked that this range is the one expected, and an empirical method to determine this range is described in this section.

In order to obtain this range, two antennas will be placed in a straight line with the access point and one of them will be moved along that line. At the same time, the real time implementation of the algorithm will be running, in order to know in each moment the amount of difference samples between the received signals. The minimum distance difference between two points with the same

difference of samples is the range looked for. Due to the finite resolution of the system, a small error is expected to be present.

WiFi Channel	Distance ambiguity (cm)	min	Distance ambiguity (cm)	max	Carrier Frequency (GHz)	Theoretical ambiguity λ_c (cm)
1	12.3		12.5		2.412	12.44
6	11.9		12.2		2.437	12.31
11	2		2.6		2.462	12.19

Table 4.10- Unambiguity range as a function of WiFi Channels, high-IF approach

Proceeding with the parameters mentioned above, the results in Table 4.10 are obtained. The WiFi Channel 11 measure is invalid, since the ambiguity range obtained is smaller than the theoretical resolution for the frequency calculated according to Table 4.9 and so, the data obtained cannot be used to get a good measure. Thus, it is possible to conclude from the other two measurements that the ambiguity range empirically measured does make sense with the theoretical results.

4.5.4.3.3 - BIAS ERROR

Another important problem that must be dealt with is the fact that a constant error may appear, due to imperfections in the components of the system. Since the unambiguity range and the resolution are small, this error should not be ignored. At this point, it could be possible to proceed in, at least, two different ways:

1. It is possible to suppose, in a first approach, that this bias error is constant. So it could be possible to be measured, manually moving the antennas until a zero distance difference (which means that the cross correlation maxima is found in zero) between them is achieved. This distance difference, then, is the bias error and it can be added as a constant to the distance calculus, so that its effect can be, if not eliminated, at least eased. The main problem that arises here is the fact that, once again, it must be supposed that this bias error is constant. As it was shown before, in the power level approach, this supposition might not be quite realistic.
2. Another possibility is simply, to ignore this error. If it is not big, it will not affect significantly the calculus, but if it is, that will be a problem. Besides, a few centimeters error is going to be quite significant, since the unambiguity range is in the range of 12 cm.

In spite of the problems that the first way has, that will be the method used. The bias error will be measured and afterwards, this distance will be considered to be constant and subtracted from the calculated distance, so that the measures got are unbiased.

In order to calculate this bias distance it, the distance $\Delta d = 0$ will be established manually, moving the antenna until a zero sample difference is achieved (so, the error in the measure of this bias distance will remain within the range of the resolution at the chosen frequency, in this case, WiFi Channel 1 is used). The bias distance obtained, when $d = 1 m$ in the setup is 16 cm, and this is going to be used in the next steps.

4.5.4.3.4 - SAMPLES NEEDED

A final consideration to be taken into account is the number of samples of the captured signal that are going to be used to compute the cross correlation. Since the signal is periodical, theoretically more samples than a few periods are required to assure a good correlation, but if too many samples are taken, the computational cost increases unnecessarily. As it was done in the FFT approach, the optimum number of samples to be used will be obtained empirically.

The proposed setup to measure the samples needed is based in the one described in Section 3.4. Taking $d = 1\text{ m}$ and $\Delta d = 10\text{ cm}$, a number of samples between 100 and 4000 is chosen, using always samples from the periodical part of the signal (to ensure that, the samples will be taken from the second half of the samples vector). Also, all the samples in the vector will be used to do the computations, just to check whether the difference is significant (20000 samples in total). Each measure will be repeated ten times and averaged with the mean, its standard deviation will be computed as well and the results can be seen in Figure 4.23 and Figure 4.24, both the unbiased measure and the ones with a corrected bias of 16 cm with a $\lambda_c = 12.5\text{ cm}$ modulus operation applied.

Two conclusions are reached:

- With as less as 200 samples, the measures obtained are already good. As the amount of data used increases, there is not a real increase in the accuracy of the system.
- The biased measures have lesser error than the unbiased ones, so it will be used in order to achieve a higher accuracy. However, the bias must be double checked, because there is still some bias error in the unbiased measure, and it must be eliminated. Since the bias origin can be diverse, it is recommendable to look for the most correct value before locating.

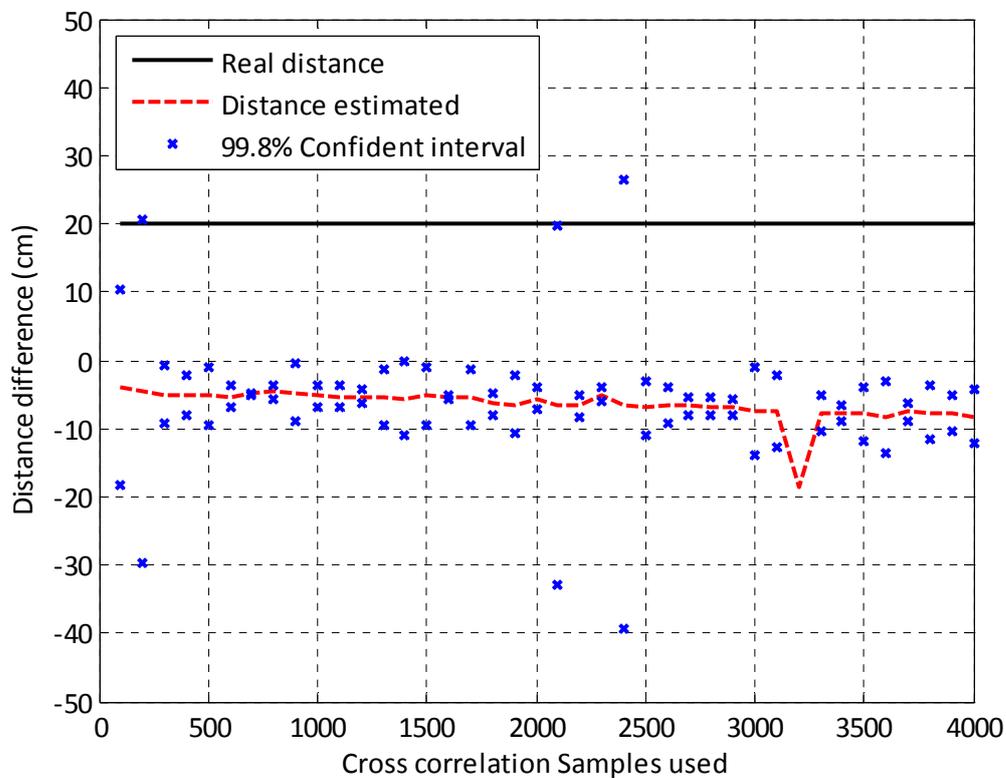


Figure 4.23 - Distance estimation as a function of cross correlation samples used, unbiased, LOS setup, high IF approach

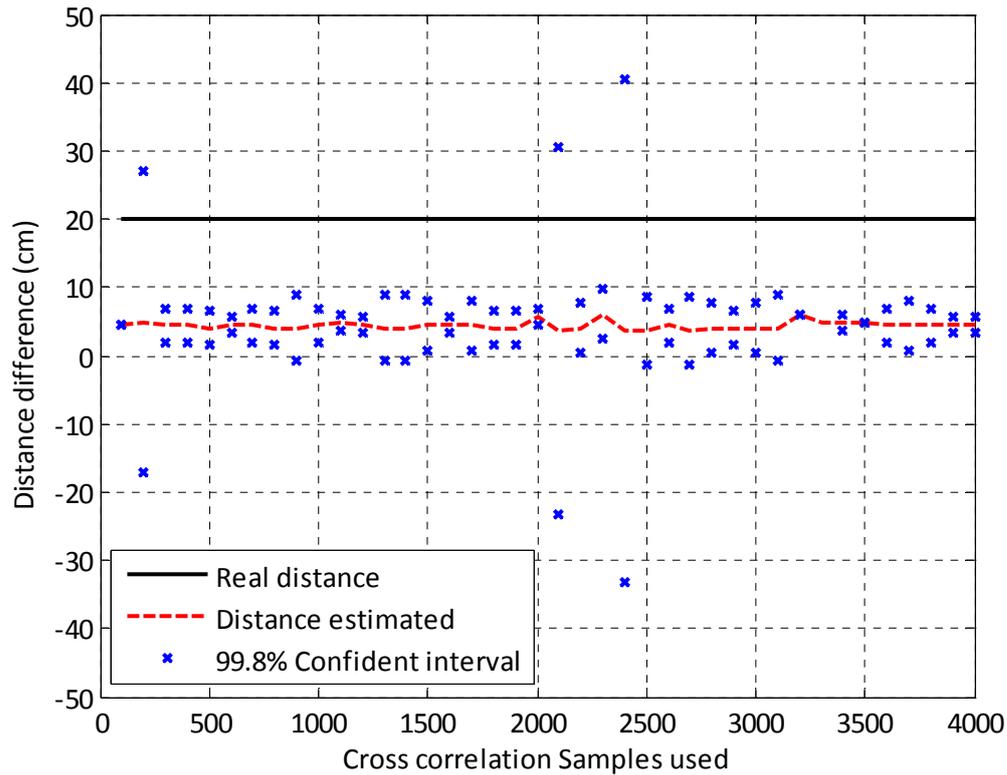


Figure 4.24 – Distance estimation as a function of cross correlation samples used, biased, LOS setup, high IF approach

4.5.4.4 - RESULTS AND CONCLUSIONS

In the LOS setup described above, the proposed algorithm will be run 50 times and then, the results will be averaged using the mean. The WiFi Channel used will be the channel 1, since it provides the best resolution (theoretically, 8.5 mm) and unambiguity range (since $\lambda_c = 12.4$ cm, both in empiric measures and analytical calculus), with $f_{ol} = 2.395$ GHz and $f_{s,if} = 250$ MHz. All the measures will have a corrected bias of 16 cm and 500 samples will be used to compute the cross correlation, using a noise power threshold of $2 \cdot 10^{-3}$ mW · Ω to discriminate between signal and noise. The results obtained are in Table 4.11.

Δd (cm)	Mean Δd (cm)	Absolute error (cm)	Relative error (%)	Mean Δd with modulus (cm)	Absolute error (cm)	Relative error (%)	STD deviation (cm)
0	-0.04	0.04	0.32	12.46	12.46	99.68	0.15
1	0.56	0.44	3.52	0.56	0.44	3.52	0
2	1.09	0.91	7.28	1.09	0.91	7.28	0.11
3	1.7	1.3	10.4	1.7	1.3	10.4	0.11
4	2.79	1.21	9.68	2.79	1.21	9.68	0
5	3.4	1.6	12.8	3.4	1.6	12.8	1.69
6	3.22	2.78	22.24	3.22	2.78	22.24	2.21

7	-0.47	7.47	59.76	12.03	5.03	40.24	3.58
8	-2.79	10.79	86.32	9.71	1.71	13.68	0
9	-1.81	10.81	86.48	10.69	1.69	13.52	0.24
10	-1.11	11.11	88.88	11.39	1.39	11.12	0
11	-0.56	11.56	92.48	11.94	0.94	7.52	0
12	0	12	96	0	12	96	0

Table 4.11 - Results of high-IF approach, LOS setup

In Figure 4.25 it is possible to see the results graphically, both with a $\lambda_c = 12.5 \text{ cm}$ modulus operation applied so that all the measures are in the range $[0 \lambda_c)$, and without it. It is possible to see that the data with the modulus operation applied have a better fit, but a problem in the extremes appear, due to the small error in the measurement and the periodicity.

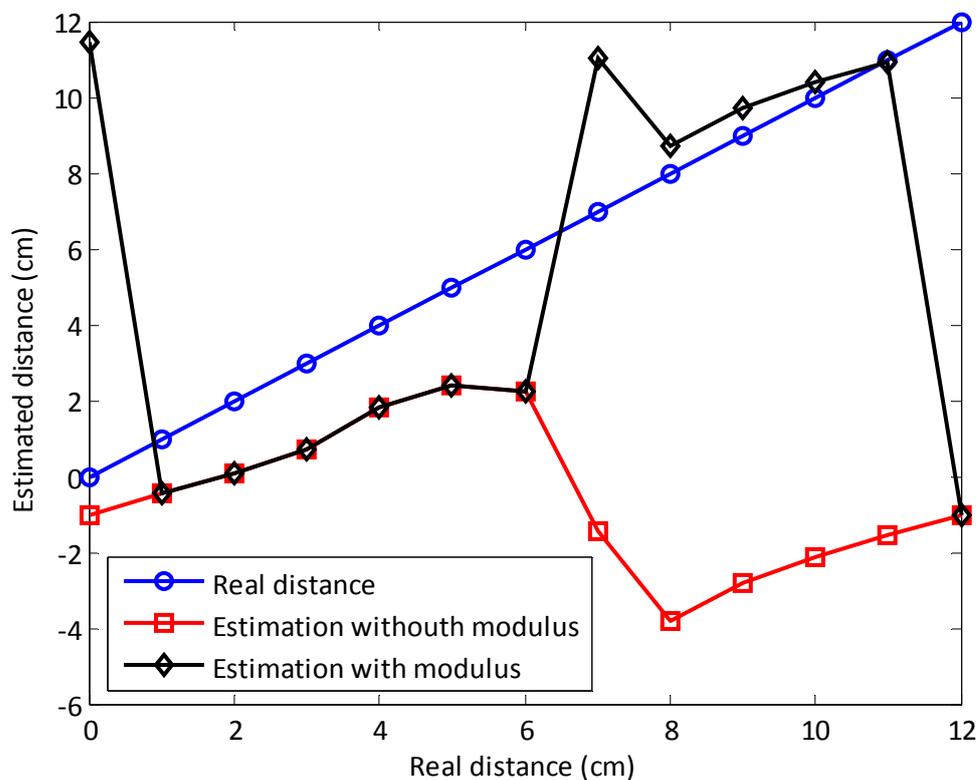


Figure 4.25 – Real and estimated distances obtained, with and without modulus, LOS setup, high IF approach

In total, there are 650 distances measured, and with them it is possible to compute the cumulative distribution functions of the absolute value of the error in the measure, which are in Figure 4.26

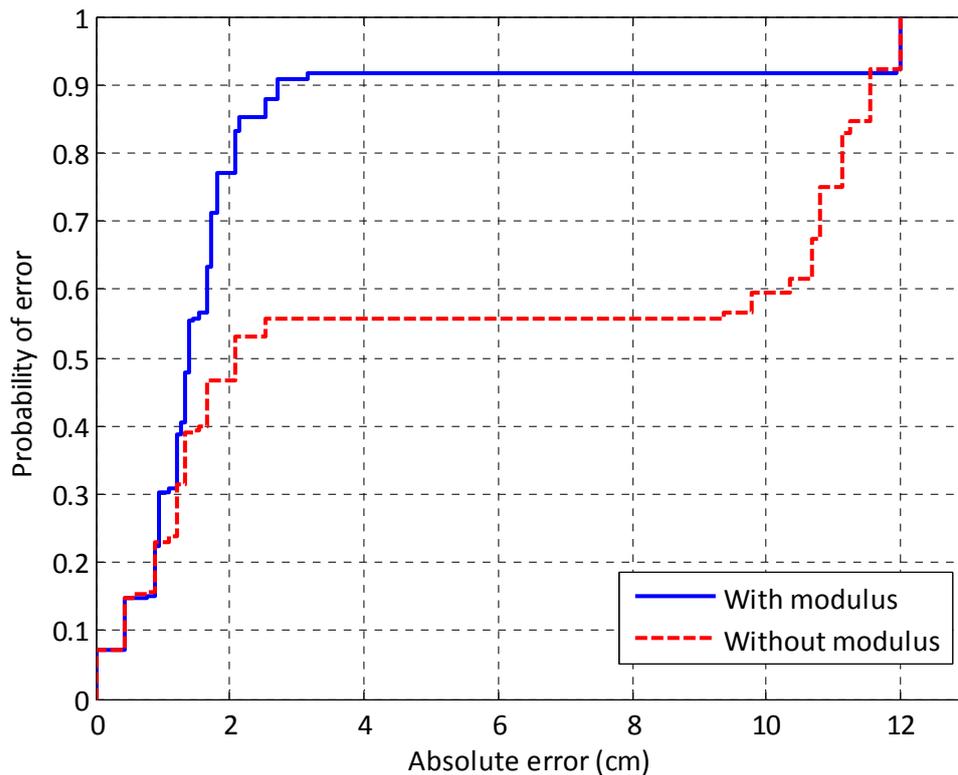


Figure 4.26 – CDF of the absolute error, with and without modulus, LOS setup, high IF approach

Some conclusions arise here:

- The absolute error of this method is really low. When using the modulus operation, almost 80% of the measures are below a 2 cm absolute error, and 90% below 3 cm. Indeed, the error that appears is due to the fact that the modulus operation fails at both extremes. So, a better algorithm that could unwrap the periodicity in the measured distance would even increase that accuracy
- However, the unambiguous range is too low, since it is only of 12 cm. That is a big hindrance, since this method can only be used to locate in very short distances or in combination with another method whose error is below this unambiguous range.
- The calculation process is quite fast, since only 500 samples has been used to compute the correlation and so, it does not take long for the computer to calculate that. The accuracy achieved with such a small number of samples and a code which is not really optimized shows that it is possible to locate using cross correlation with a low computational cost.

4.5.5 - NEAR-ZERO-IF APPROACH

Since this approach has two separate methods to estimate the location, each one of them having very different characteristics in unambiguity range and resolution, both setups described in Chapter 3.4 and 3.3 are going to be used in order to test the proposed approach in a wireless environment. The setup in Chapter 3.3 is going to be used with the Barker-IF method, whereas the one in Chapter 3.4 is going to be used with the sinusoid-IF method.

4.5.5.1 - ALGORITHM DESCRIPTION

The proposed algorithm will be very similar to the one proposed in Section 4.3.2, since the only variations appear in the processing signal part of the code and are due to the fact that in this method, two different algorithms are run in parallel.

Once the data of the downconverted waves have been obtained, MatLAB will read them and compute the two methods above described, sinusoid-IF and Barker-IF, in order to obtain two distance estimations as follows:

1. First, an envelope detector will be used to get the fine position that sinusoid-IF method yields. With the envelope of both signals, the delay between envelopes will be computed using the cross correlation and its maximum will be used in order to get the location estimation.
2. Secondly, the Barker-IF location will be obtained, so the cross correlation of the two IF signals will be computed, using a few samples only so that the envelope information does not significantly affect the result.

However, since each method has to be tested in a different setup, they will be run separately, and so, even though the code provided in the appendix computes both estimations at once, for testing purposes the estimations will be done separately.

4.5.5.2 - GENERAL PARAMETERS STUDY

In this section, some of the parameters that are going to be used in the code and that affect both to Barker-IF and sinusoid-IF methods are studied and their influence on the location estimation is explained, so that a better understanding of them is obtained.

4.5.5.2.1 - INTERMEDIATE FREQUENCY VALUE

As it has been already proved in Section 4.5.2, the intermediate value frequency is key in order to make possible that the near-Zero-IF approach works, since a high intermediate frequency will make envelope detection harder and a low one will cause that the envelope term is controlled basically for the jitter between oscillators, so any variation on this jitter will have a strong impact on the sinusoid-IF location.

Thus, the IF value must be chosen low enough that the envelope is easy to obtain, but high enough that a jitter variation does not have a strong impact on the system performance. In this concrete system, the jitter value is, according to Section 4.5.2 calculations, 14 kHz. So, if $f_{if} = 100 \text{ kHz}$ is used, both conditions before are satisfied: a variation in the jitter will have little influence in the system performance, since f_{if} is high enough (almost a magnitude order), and it is also low enough, that the envelope might be easy to detect (refer to Section 4.5.5.4.2 for more details on envelope detection).

In the code used, the jitter is corrected when setting the intermediate frequency. Thus, the oscilloscope oscillator will be set up to achieve the desired intermediate frequency, correcting the estimated jitter that is present in the system.

4.5.5.2.2 - RESOLUTION

In the near-Zero-IF approach, there are two different ways to estimate the distance difference, Barker-IF and sinusoid-IF. It has already been proved in equation (4.5.12) that the sinusoid-IF approach will have a small unambiguity range, and that is a disadvantage when compared to the Barker-IF method. However, when it comes to resolution, the sinusoid-IF method has a much finer behavior.

As it is deduced from (4.5.9), the Barker-IF resolution, in terms of distance difference, is:

$$\Delta d_{min,bar} = t_0 \cdot c = \frac{n_s}{f_s} \cdot c = \frac{c}{f_s}$$

That means, that the only parameter of the system that can be adjusted in order to get a finer location is the sampling frequency, but really high sampling frequencies are needed in order to get a good resolution and so, the cost of the system increases because a more expensive ADC is needed, as well as more memory to store samples and a more powerful DSP to do the calculations.

On the other hand, as it was proved in (4.5.14), the resolution in distance difference terms of the sinusoid-IF method is:

$$\Delta d_{min,sin} = \frac{c}{f_c} \cdot f_{if} \cdot \frac{n_s}{f_s} = \frac{c}{f_c} \cdot \frac{f_{if}}{f_s}$$

And so, the resolution relation is:

$$\Delta d_{min,sin} = \Delta d_{min,bar} \cdot \frac{f_{if}}{f_c} \tag{4.5.16}$$

If the IF chosen is equal to the carrier frequency, the resolutions would be the same. However, in order to get a high resolution, in this setup the intermediate frequency will be lower than the carrier frequency. Also, in terms of resolution, a high sampling frequency and a low intermediate frequency are preferred if a high resolution is needed.

In the concrete setup used, $f_{if} = 100 \text{ kHz}$ and $f_c = 2.412 \text{ GHz}$ (WiFi Channel 1), so that means that the resolution relation will be, according to (4.5.16):

$$\Delta d_{min,sin} = \Delta d_{min,bar} \cdot 4.15 \cdot 10^{-5}$$

4.5.5.3 - BARKER-IF PARAMETERS STUDY

In this section, some of the Barker-IF parameters that are going to be used in the code are studied and their influence on the location estimation is explained, so that a better understanding of them is obtained.

4.5.5.3.1 - SAMPLING FREQUENCY ELECTION

As it was already seen in Section 4.5.5.2.2, in this method the only parameter that controls the resolution is the sampling frequency, so the election of this parameter will have a strong impact in the system features and in the cost too.

If $f_s = 250 \text{ MHz}$ is chosen, then according to equation (4.5.9), the theoretical resolution of the system would be:

$$\Delta d_{min,bar} = \frac{c}{f_s} = 1.2 \text{ m}$$

It is not a big resolution, so a big error is going to be obtained with this approach. If the sampling frequency is increased and $f_s = 500 \text{ MHz}$ is used, then the resolution becomes:

$$\Delta d_{min,bar} = \frac{c}{f_s} = 0.6 \text{ m}$$

In order to check the system real performance, both sampling frequencies will be used in order to determine the system performance under both conditions. A higher sampling frequency should mean a higher resolution and a smaller error, but the cost is increased as well.

4.5.5.3.2 - CROSS CORRELATION SAMPLES USED

A key parameter is the number of samples used in the cross correlation. An advantage of the cross correlation is that a high number of samples is not required in order to get good results. In the case of a periodical signal, as the Barker Code that is used in this case, only a few periods are needed. Assuming that at least a period of the signal is needed in order to obtain acceptable results when computing the cross correlation, and since the Barker Code received has a frequency of 11 MHz , then the number of samples that are needed for a period are:

$$n_{s,per} = T_{barker} \cdot f_s = \frac{f_s}{f_{barker}} \quad (4.5.17)$$

And so, the number of samples in a period for each sampling frequency used is:

$$f_s = 250 \text{ MHz} \rightarrow n_{s,per} = \frac{f_s}{f_{barker}} = 23$$

$$f_s = 500 \text{ MHz} \rightarrow n_{s,per} = \frac{f_s}{f_{barker}} = 46$$

That means that a higher value than these must be taken in order to ensure obtaining correct cross correlation values. A smaller value could be taken as well, but it is preferable using those values, not only because some periods are used and that improves accuracy, but as well because if not enough samples are taken, there could be no samples enough to obtain the correct time delay.

Also, there is an upper limit in the approach, since if too many samples are taken, the envelope will have an impact on the cross correlation, to the point that it could become dominant and so, the equations (4.5.8) and (4.5.9) would not be valid anymore.

In order to set a limit, it could be considered that the cross correlation is not significantly affected by the envelope if the number of samples used to compute it is smaller than the tenth part of the envelope period, namely:

$$t_{corr} \leq \frac{T_{env}}{10} \quad (4.5.18)$$

For calculus purposes it can be said that $f_{if} = f_{env}$, assuming that the oscillator jitter has little effect on the envelope frequency (the intermediate frequency was chosen to fulfill that condition, Section

4.5.5.2.1). Thus, it is possible to theoretically determine the maximum number of samples to be used, which is:

$$t_{corr} = \frac{n_s}{f_s} \leq \frac{T_{env}}{10} \leq \frac{1}{10f_{env}} \leq \frac{1}{10f_{if}} \rightarrow n_{s,max} \leq \frac{f_s}{10f_{if}} \quad (4.5.19)$$

Since the intermediate frequency and the sampling frequencies are already known, it is possible to determine the limits in each case:

$$f_s = 250 \text{ MHz} \rightarrow n_{s,max} \leq \frac{f_s}{10f_{if}} \leq 250$$

$$f_s = 500 \text{ MHz} \rightarrow n_{s,max} \leq \frac{f_s}{10f_{if}} \leq 500$$

These expressions will be checked empirically. To do so, a setup like the one described in Section 3.4 is used, with a distance difference between antennas $\Delta d = 1.5 \text{ m}$ and a variable number of samples between 25 and 500, with an increase of 25 samples. The results for each number of samples are repeated 50 times and they are averaged using the median, and the results obtained are plot in Figure 4.27

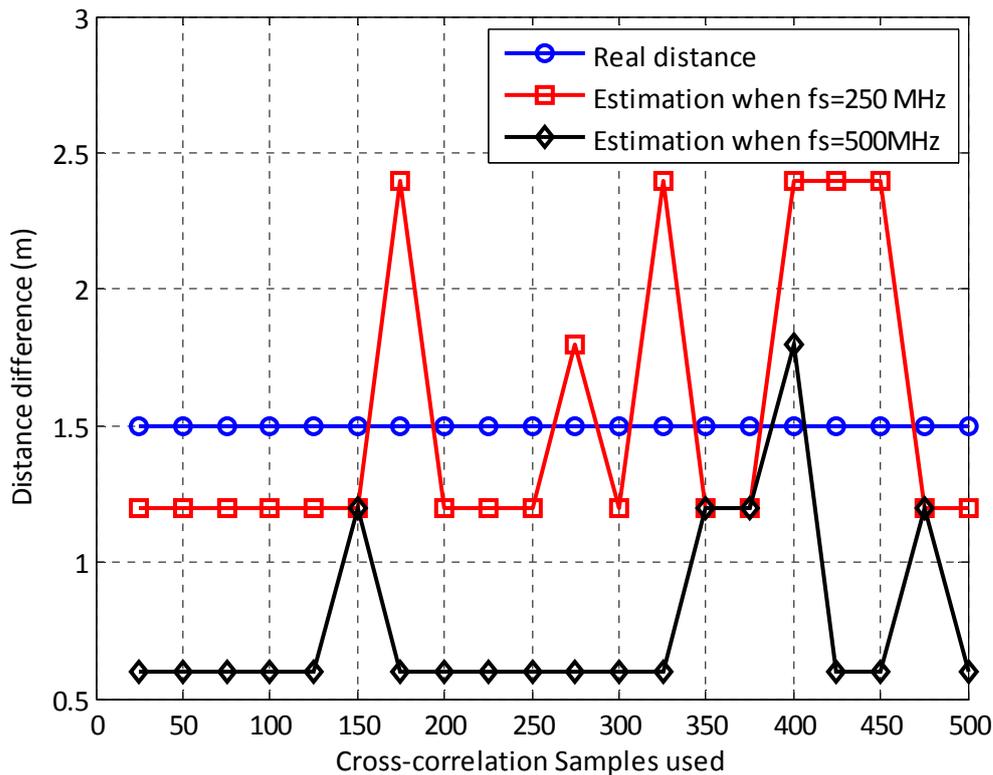


Figure 4.27 - Estimated distance difference as a function of cross correlation samples and sampling frequency

It is possible to see that in the first case, when $f_s = 250 \text{ MHz}$, the results obtained are the ones expected: from approximately 250 samples on, the error starts to increase. However, in the second case, the result is not that clear. Because of that, the tests will be run both with 250 and 500 samples, in each sampling frequency, so that better conclusions can be extracted from there. Theoretically, when $f_s = 250 \text{ MHz}$, with 250 samples the error should be equal or smaller than with

500 samples, whereas when $f_s = 500 \text{ MHz}$, a similar result should be obtained with both numbers of samples.

4.5.5.3.3 - ABSOLUTE VALUE

According to equations (4.5.6) and (4.5.7), another problem arises when computing the cross correlation. If the cross correlation is applied to those signals, the result would be:

$$y = \int_{-\infty}^{+\infty} x_{1,if}^* \cdot x_{2,if}(t + \tau)dt = \cos(\varphi_2) \cdot \cos(\varphi_2 - \theta) \cdot \int_{-\infty}^{+\infty} B(t) \cdot B(t - t_0 + \tau)dt \quad (4.5.20)$$

It is possible to see that the cross correlation function will be multiplied by a term phase dependent, which can be positive or negative: thus, it will not be clear whether a maximum or a minimum is to be expected. In order to solve this, the absolute value of the cross correlation will be used in order to detect its maxima and so, the problem with the phase terms disappears.

However, this brings another problem, due to the fact that a minimum could be detected as a maximum, for example, if the signals have a π phase shift. But since the Barker Codes are specifically designed to minimize this effect [28], it should not be a problem.

4.5.5.3.4 - UNAMBIGUITY RANGE

Since the Barker Code is a repeated in the WiFi preamble, there will be an unambiguity range related to its duration. That means that if the time of propagation is greater than the Barker Code duration, the location will become ambiguous. Since the signal is repeated every $T = 1 \mu s$, according to equations (4.5.8) and (4.5.9), the unambiguity range will be:

$$\Delta d_{max} = T \cdot c = 300 \text{ m}$$

4.5.5.4 - SINUSOID-IF PARAMETERS STUDY

In this section, some of the sinusoid-IF parameters that are going to be used in the code are studied and their influence on the location estimation is explained, so that a better understanding of them is obtained.

4.5.5.4.1 - PHASE CONSTANT OFFSET

As it was described in Section 4.5.4.3.3, a phase offset may appear in this approach, and as it is described there, it can be corrected by measuring the zero phases offset manually and then use that empirical offset for location.

In this concrete setup, the values obtained, using the method described in that section, are:

$$f_s = 250 \text{ MHz} \rightarrow d_{offset} = 14 \text{ cm}$$

$$f_s = 500 \text{ MHz} \rightarrow d_{offset} \approx 14 \text{ cm}$$

For testing purposes, it will be supposed that the distance is the same in both cases.

4.5.5.4.2 - ENVELOPE DETECTION

In this approach, the key element is the envelope detector, since it allows getting the information needed to locate. In order to obtain this envelope, at least four different methods could be used:

- Since the shape of the envelope is known (in this case, it is a sinusoid), it could be possible to use a fit in the time signals, so that the envelope is obtained. That means that the envelope would be obtained as the output of a fitting block. This method has a high computational cost, but if the fit obtained is good, it allows obtaining a good interpolation and so, a very precise detection.
- A low pass filter can be used to obtain the envelope, because that information is located in the low frequencies received. A filter or a set of them can be used in order to obtain the signal envelope and so, the location information would be obtained.
- A variation of the method mentioned before is computing the FFT of the signals and then compute the inverse FFT of the low frequency samples. It is another way to filter, which is really fast because it is FFT-based, but has a limitation due to FFT resolution.
- Using a DSB demodulator, since as it was shown in Section 4.5.2, the intermediate frequency signals follow this modulation schema.

In this case, the method chosen to be used will be the third one (but all of them could be used). The FFT of the IF signals will be obtained, then all the FFT samples will be set to zero except the low frequencies samples and then the inverse FFT will be computed.

First, the signals will be decimated before computing the FFT, since the signal bandwidth is 20 MHz and the sampling frequencies used are higher than the Nyquist frequencies needed (250 and 500 MHz are the sampling frequencies used, while 40 MHz is the Nyquist frequency). This allows conserving the low frequency information, as well as reducing the amount of samples to be processed. A decimation factor of $N = 10$ will be used, and the effects that it has on the signal spectrum when the original sampling rate is 250 MHz can be seen in Figure 4.28.

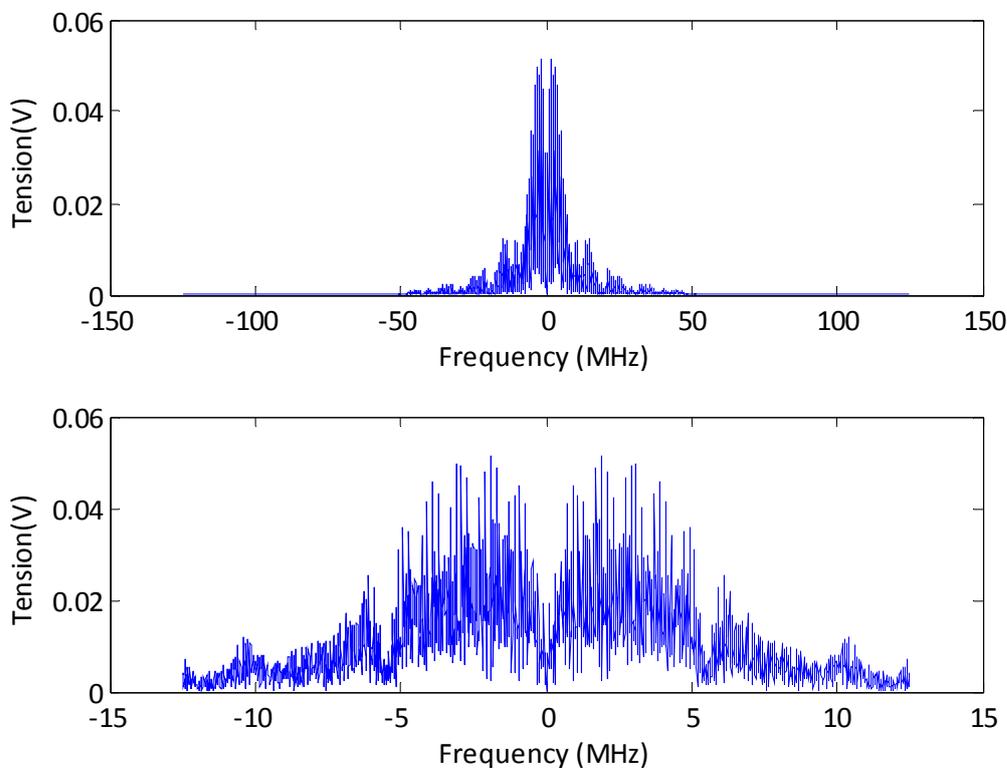


Figure 4.28 - Signal IF spectrum, before and after decimation (N=10)

Secondly, the FFT spectral resolution sets a limit on the system performance. To check its effect, this resolution must be obtained, and since the number of samples used to compute the FFT is $N = 2 \cdot 10^4$, then the FFT spectral resolution will be:

$$f_s = 250 \text{ MHz} \rightarrow \text{resolution} = \frac{f_s}{N} = 12.5 \text{ kHz/sample}$$

$$f_s = 500 \text{ MHz} \rightarrow \text{resolution} = \frac{f_s}{N} = 25 \text{ kHz/sample}$$

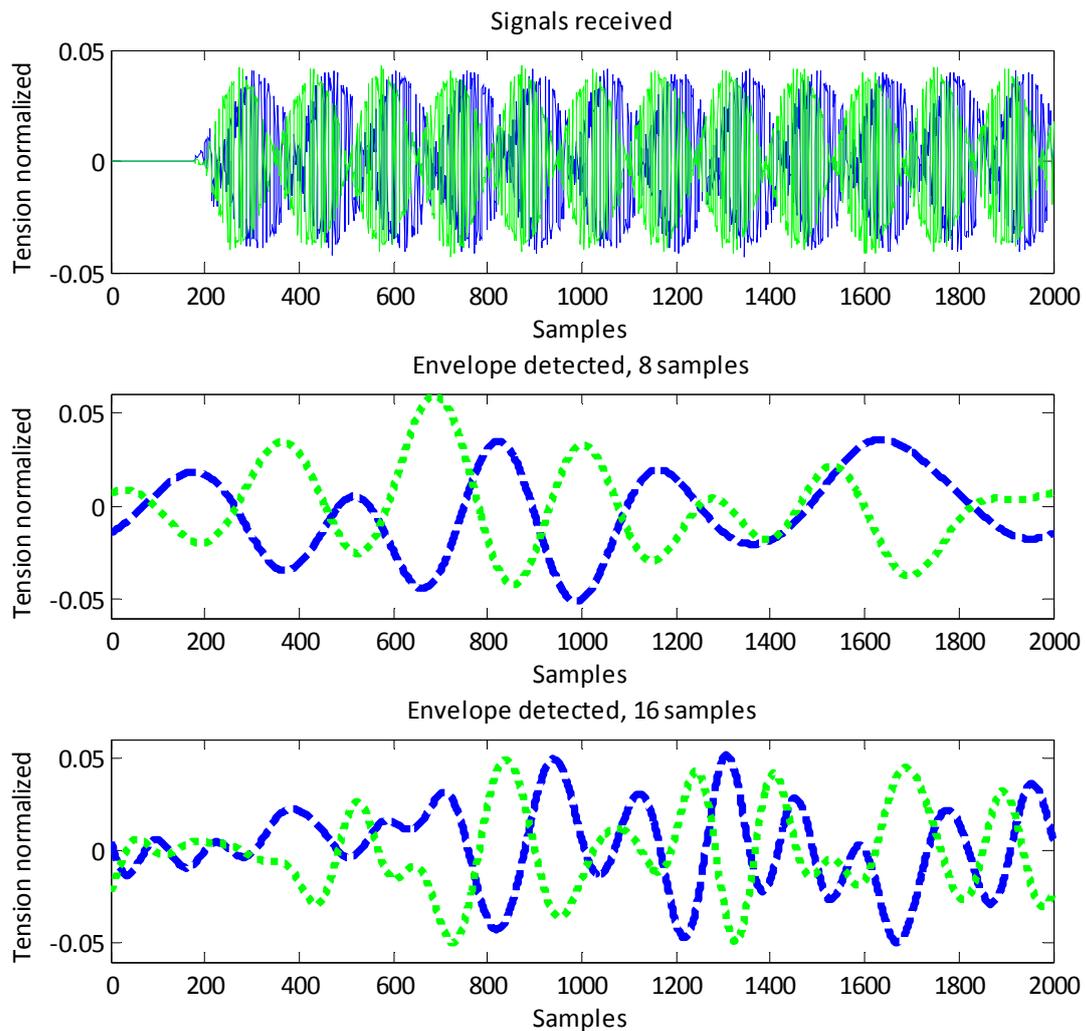


Figure 4.29 - Received signals and envelope detected when $f_s = 250 \text{ MHz}$

Since the envelope signal has a frequency $f_{env} \approx f_{if} = 100 \text{ kHz}$, the number of samples of the FFT which are not set to zero must be big enough so that this frequency remains. Theoretically, the minimum number of samples would be:

$$f_s = 250 \text{ MHz} \rightarrow n_{s,min} = \frac{f_{env}}{\text{resolution}} = \frac{f_{env}}{f_s} N = 8 \text{ samples}$$

$$f_s = 500 \text{ MHz} \rightarrow n_{s,min} = \frac{f_{env}}{\text{resolution}} = \frac{f_{env}}{f_s} N = 4 \text{ samples}$$

In order to estimate the effect of the number of samples in the envelope detection, the mentioned algorithm was run using 8 and 16 samples in the FFT not set to zero, and the resulting envelope, as well as the original signal in each case, can be seen in Figure 4.29 and Figure 4.30.

As it was expected, the envelope detection using this method is not very accurate. However, since the theoretical resolution on this method is really big, a tradeoff could be achieved. For testing purposes, 8 samples will be used in both cases and the results will contain error due to the method used.

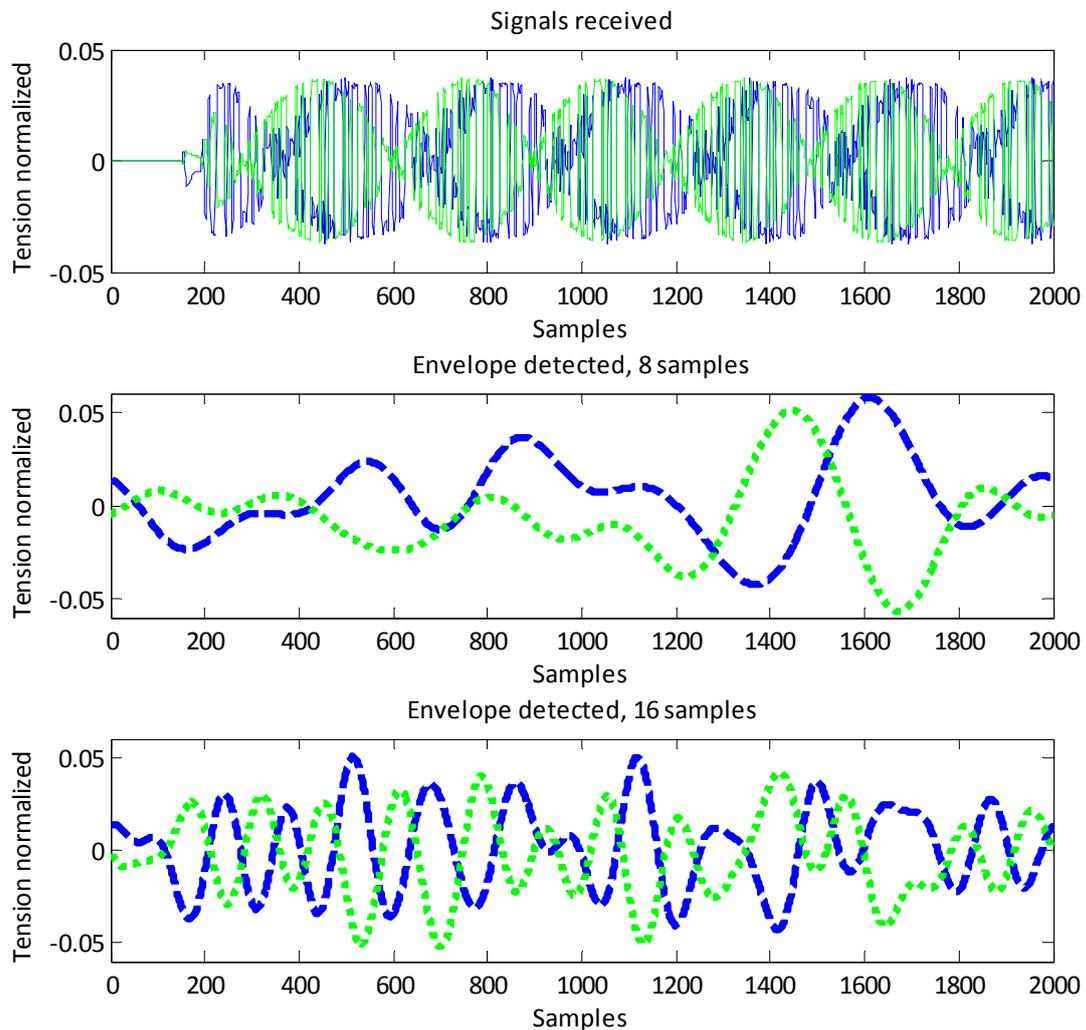


Figure 4.30 - Received signals and envelope detected when $f_s = 500 \text{ MHz}$

4.5.5.5 - RESULTS AND CONCLUSIONS

In the LOS setups described in Section 4.5.5, the proposed algorithm will be run 50 times and then, the results will be averaged using both mean and median. The WiFi Channel used will be the channel 1, since it provides the best resolution (theoretically, 8.5 mm) and unambiguity range (since $\lambda_c = 12.4$ cm, both in empiric measures and analytical calculus), with $f_{if} = 100$ kHz and $f_s = 250$ MHz and $f_s = 500$ MHz. A noise power threshold of $2 \cdot 10^{-3}$ mW · Ω has been used to discriminate between signal and noise. Since both algorithms will be tested separately, the results of Barker-IF will be in Section 4.5.5.5.1 and the results of Sinusoid IF, in Section 4.5.5.5.2

4.5.5.5.1 - BARKER-IF METHOD

The obtained results, with the parameters described before when the total distance between antennas is $d = 6$ m are in Table 4.12 ($f_s = 250$ MHz) and Table 4.13 ($f_s = 500$ MHz). The obtained results are in Table 4.12 and Table 4.13, and the empirical CDF, computed with the 350 data measures obtained in each case, are plot in Figure 4.31.

$d_1(m)$	250 samples			500 samples		
	d_1 estimated(m)	Absolute error (m)	Relative error (%)	d_1 estimated(m)	Absolute error (m)	Relative error (%)
0	-0.6	0.6	10	-0.9	0.9	15
1	0	1	16.67	-0.9	1.9	31.67
2	2.1	0.1	1.67	1.8	0.2	3.33
3	4.8	1.8	30	0	3	50
4	11.4	7.4	123.33	10.8	6.8	113.33
5	6.6	1.6	26.67	7.8	2.8	46.67
6	9	3	50	9	3	50

Table 4.12- Results of Barker IF approach, LOS setup, $f_s = 250$ MHz

$d_1(m)$	250 samples			500 samples		
	d_1 estimated(m)	Absolute error (m)	Relative error (%)	d_1 estimated(m)	Absolute error (m)	Relative error (%)
0	0	0	0	-1.5	1.5	25
1	-0.75	1.75	29.17	-3.15	4.15	69.17
2	2.1	0.1	1.67	1.8	0.2	3.33
3	3.3	0.3	5	-1.2	4.2	70
4	11.1	7.1	118.33	12.6	8.6	143.33
5	4.2	0.8	13.33	7.2	2.2	36.67
6	5.5	0.5	8.33	8.1	2.1	35

Table 4.13 - Results of Barker IF approach, LOS setup, $f_s = 500$ MHz

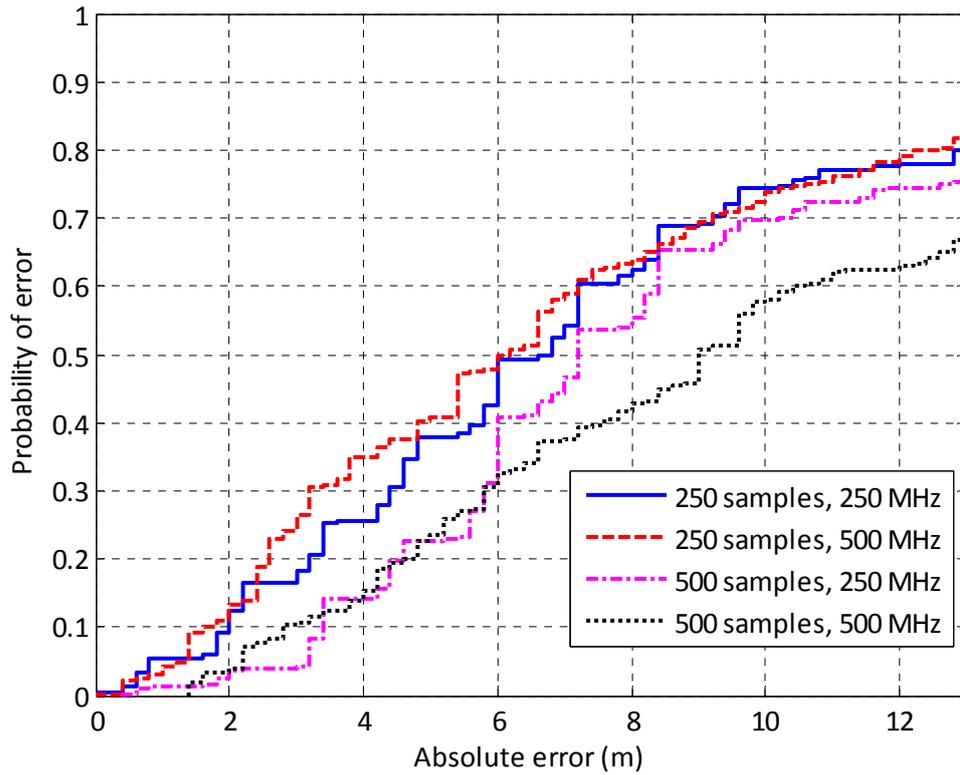


Figure 4.31– CDF of the absolute error, Barker-IF approach, no limit

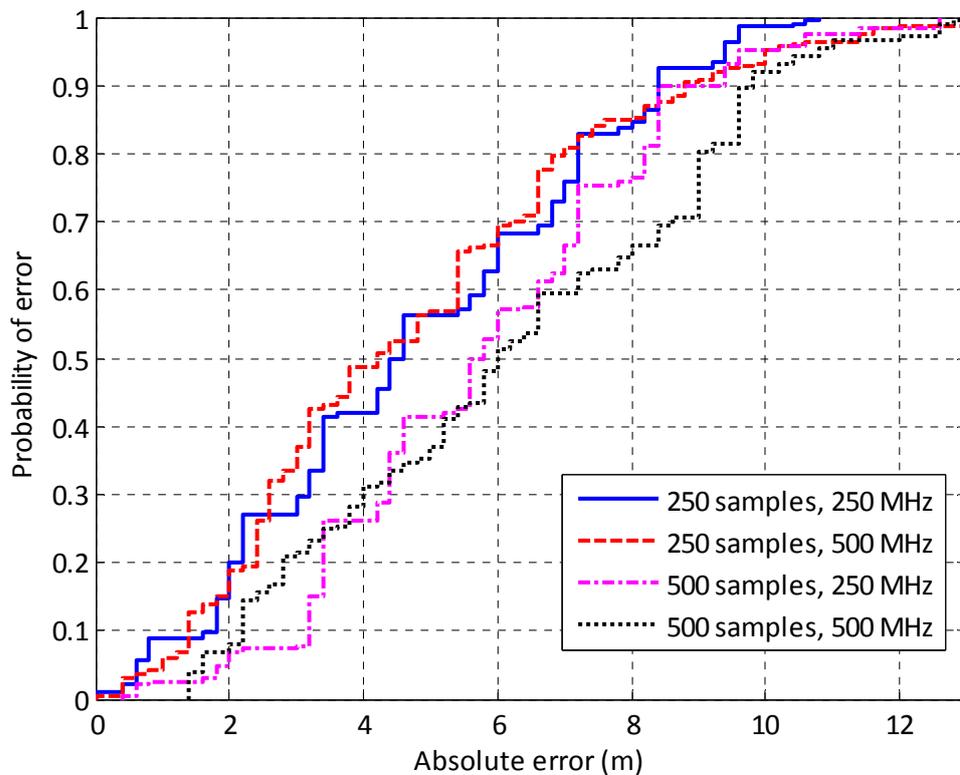


Figure 4.32– CDF of the absolute error, Barker-IF approach, limit used

The results obtained are quite unaccurate. However, there are some very big errors in some measurements which could be a reason for that. In order to test so, the CDF of the same data sets is plot in Figure 4.32, but every estimation of d_1 whose absolute value is larger than 10 m will not be

used, so that there is a limit in the possible distance obtained and so, it is possible to check whether this causes this algorithm to locate unaccurately.

Some conclusions that arise here are:

- The error is high, even if the distance is limited. It is greater than 6 m at 80% of measurements, in all cases, which is a relative error of 100%. So, the accuracy of this method is poor, even though a big improvement has been achieved when limiting the maximum distance
- In order to solve this problem, a more thorough study should be done in order to detect why this method is not giving the expected results. Subsequently, further investigation may explain why the absolute error is so big and thus, a solution could be provided.
- It could be possible that the error is not strongly related to the distance between antennas and so, it should be tested in an environment with a larger distance, since the absolute error might be fix in 6 meters and so, it could be a solution to locate roughly in a large environment.

4.5.5.5.2 - SINUSOID-IF APPROACH

In this approach, using the parameters described in section 4.5.5.4, a distance offset of 14 cm, a distance modulus of 12.5 cm for calculus purposes (exactly as in Section 4.5.4.4) and an envelope detector based on the FFT approach described in section 4.5.5.4.2, with a decimation factor $N = 10$ and using 8 FFT samples, the obtained results are in Table 4.14, the obtained CDF is in Figure 4.33, and a comparison between real and estimated distance is in Figure 4.34.

$d_1(cm)$	$f_s = 250 MHz$			$f_s = 500 MHz$		
	$d_1 estimated(cm)$	Absolute error (cm)	Relative error (%)	$d_1 estimated(cm)$	Absolute error (cm)	Relative error (%)
0	12.3259	12.3259	98.6	12.4254	12.4254	99.4
1	0.6468	0.3532	2.82	0.4975	0.5025	4.02
2	2.7861	0.7861	6.29	1.1194	0.8806	7.04
3	4.7761	1.7761	14.21	1.8781	1.1219	8.98
4	5.3483	1.3483	10.79	2.7861	1.2139	9.71
5	6.393	1.393	11.14	3.6862	1.1318	9.05
6	2.9726	3.0274	24.22	4.9005	1.0995	8.8
7	9.5522	2.5522	20.42	6.7537	0.2463	1.97
8	9.7761	1.7761	14.21	7.2761	0.7239	5.79
9	6.6542	2.3458	18.77	8.1841	0.8159	6.53
10	7.8234	2.1766	17.41	8.8682	1.1318	9.06
11	10.1866	0.8134	6.51	10.0373	0.9627	7.7
12	11.505	0.495	3.96	11.6294	0.3706	2.97

Table 4.14 - Results of sinusoid-IF approach, LOS setup

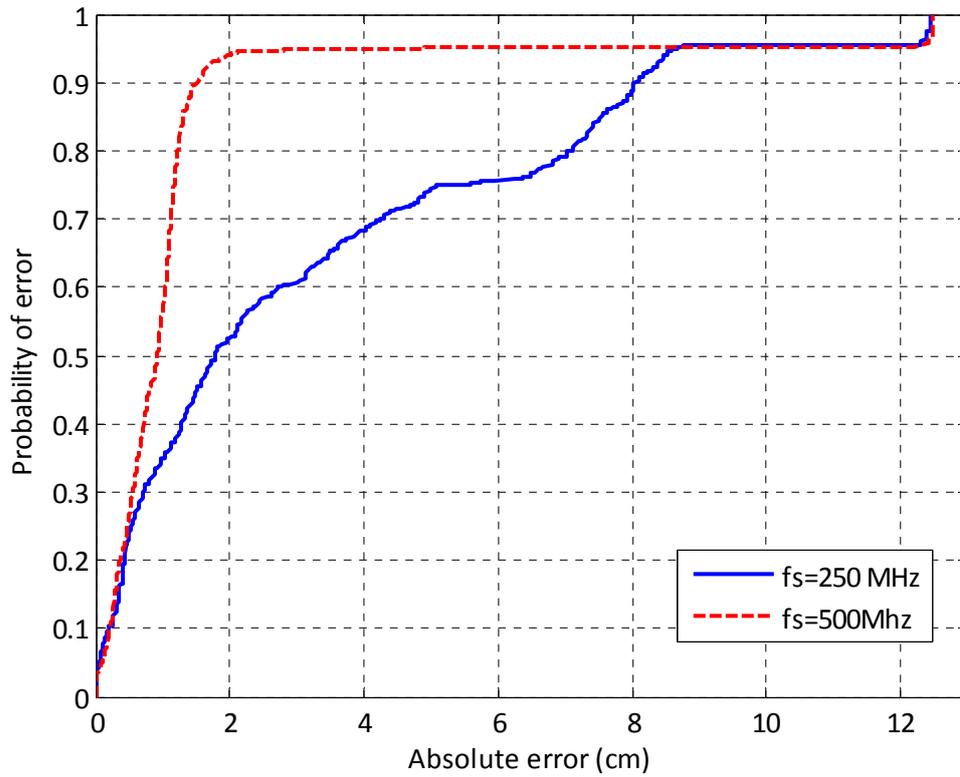


Figure 4.33 - CDF of the absolute error, Sinusoid-IF approach

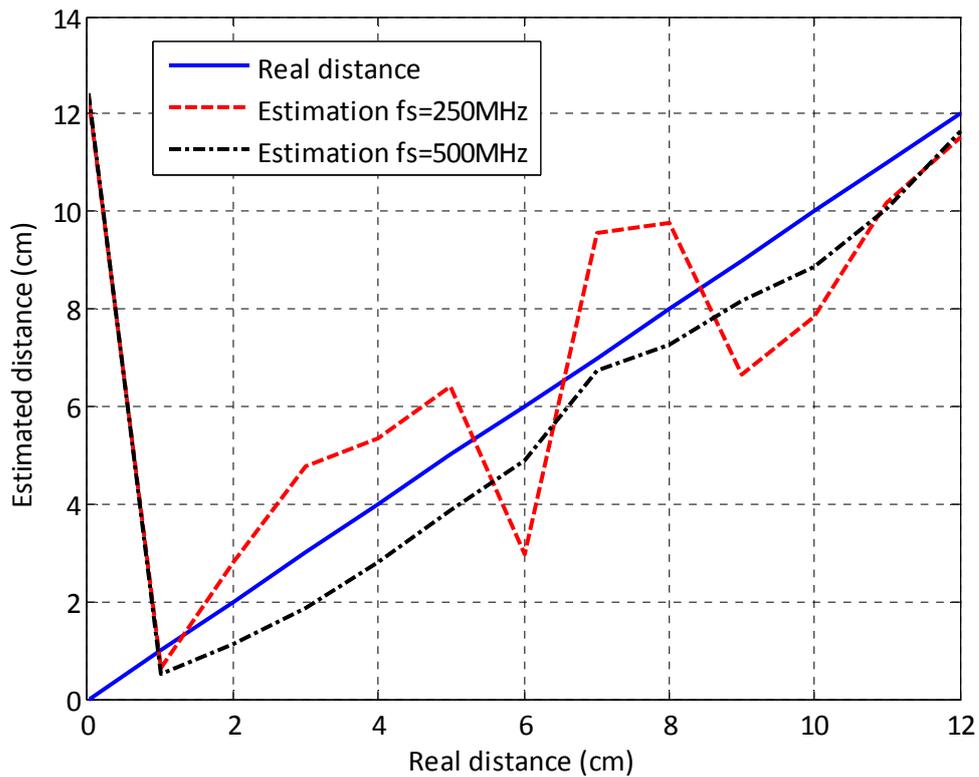


Figure 4.34– Real and estimated distance, Sinusoid-IF approach

Some conclusions that can be observed are:

- The results obtained with this method are nearly the same as the one described in section 4.5.4. Even though the intermediate frequency and the method of computing the distance estimation are different, the results are quite similar. That indicates that the Sinusoid-IF method performs quite good independently of the method used
- The results are better when $f_s = 500 \text{ MHz}$ are used. The main reason might be the envelope detector used, since the same number of samples are used with both sampling frequencies, even though that means that different frequencies are eliminated in each setup.
- Also, it is remarkable that the envelope detector used, based on FFT, gives so accurate results. It is very likely that an improvement in this envelope detector, using a better method, will improve the accuracy obtained
- However, the main problem of this method remains: its low unambiguity range is a big limitation that cannot be avoided and it is not a really valid approach in order to get the absolute position for environments in real life

4.6 - COMPARATIVE

Four different approaches have been tested in order to determine whether they are valid or not for LOS location, using the WLAN 802.11b specifications. Some of the conclusions reached are summed up in Table 4.15.

Method	Specifications	Absolute error @ 80%	Relative error @ 80%	Advantages	Disadvantages
Frequency sweep	11 Channels, phase unwrapped	80 cm	13.3 %	High accuracy High unambiguity range (60 m)	Slow Expensive downconverter
Frequency sweep	3 Channels, phase unwrapped	250 cm	41.7 %	Faster than 11 channels	Medium unambiguity range (12 m) Bad accuracy Expensive downconverter
FFT	6 meters antenna distance	-	-	Very high unambiguity range (>1 km) Cheap downconverter	Unknown problems cause a poor location
Power level	Channel 1, offset used	90 cm	15 %	Cheap downconverter Very fast Good accuracy	Strongly dependent on parameter estimation
Power level	Channel 1, no offset used	110 cm	18.3 %	Cheap downconverter Very fast Medium accuracy	Strongly dependent on parameter estimation
IF Cross correlation	Modulus used	2 cm	16.6 %	Very high accuracy Cheap downconverter Fast	Very low unambiguity range (12 cm)

Near-Zero-IF, Barker-IF approach	250 samples, 10 m limit	7 m	116.67 %	Cheap downconverter Fast High unambiguity range	Very inaccurate
Near-Zero-IF, sinusoid-IF approach	$f_s = 500 \text{ MHz}$	1.6 cm	13.33 %	Very high accuracy Cheap downconverter Fast	Very low unambiguity range (12 cm)

Table 4.15 - Comparison of proposed location methods

The methods observed could be classified attending to the next parameters:

- Accuracy: In relative terms, all the methods used give a similar accuracy, between 10% and 20%, except Barker-IF, FFT and 3 channels frequency sweep. However, in absolute values, the IF Cross correlation and Sinusoid-IF give the highest accuracy (absolute error of 1.5 -2 cm), when compared to any other method. Then, the power level method and the frequency sweep give a good accuracy, below 1 meter.
- Range of measure: Frequency sweep, FFT and cross correlation (sinusoid-IF and high-IF) yield a periodical result in the location calculus, which means that there is a range within the measure will be unambiguous and so; there is a maximum location range. The first has the possibility to adjust that range, so that it does not become a problem, at a cost of time, and so a tradeoff between time and range is possible to be achieved; the second offers also the possibility of adjusting parameters (number of samples and sampling frequency) in order to obtain a tradeoff between computational cost and unambiguity range; whereas the last has a measuring range constrained by the carrier wavelength and so, in 2.4 GHz frequency, the maximum range is quite small, about 12 cm. Barker-IF also has an unambiguity range due to the signal periodicity, big enough not to become a problem (300 m)
- Speed: FFT and power level approaches are really fast in their calculation process, and so they can yield a few positioning estimations per second. Also, cross correlation could be quite fast, since not quite a lot samples are needed. However, in the implementation made, the frequency sweep method was really slow since the WiFi channels were manually changed. Moreover, since this approach is based on measuring a few times, whereas the other methods could work with only one measure, that will cause this method to be inherently slower than the others.
- Cost: The main parameter that differs from one method to another, in terms of cost, is the downconverter. All methods use a normal downconverter except for the frequency sweep, where an IQ signal is needed, and so this will be more expensive than the others. This is something that must be taken into account as well when designing a real life system.

Chapter 5 - SUMMARY AND OUTLOOK

The objective of this work was to implement and compare four different indoor location methods that would work using the 802.11b standard signal. The four methods were implemented using MatLAB and put to the test in two different environments: a cable based one and a wireless, LOS setup, and the results obtained, including accuracy, cost, speed and advantages and disadvantages of each method were analyzed as well.

5.1 - SUMMARY OF THE WORK

The four methods yield good results in the cable environment setup, what means that they are all feasible to be implemented. However, this environment is a nearly-ideal one and subsequently, when testing each method in a real, wireless setup, the results vary widely. In general, a reduction in the accuracy in all the methods was observed when going from the cable to the LOS setup, as expected, but some methods like FFT suffered greatly, so good results could not be obtained. Others, however, give good results in a LOS environment, like frequency sweep or power level, which makes them more promising for being implemented in a real life indoor location system. And finally, the cross correlation based approach showed an excellent accuracy when a high IF was used or the envelope was studied with a near-zero-IF approach, in the range of centimeters, but a worse accuracy is obtained when the Barker-IF approach is used to compute the absolute distance.

Three of the methods described, FFT, IF cross correlation and frequency sweep, have an unambiguity range that can be adjusted in some of the methods described because it depends on system parameters, but in others, it only depends on the wavelength of the transmitted signal. Thus, this limit imposes a limitation on the maximum range of location of each method.

5.2 - FUTURE WORK PROPOSALS

Some final thoughts on each method are:

- Frequency sweep: As it was seen, the main disadvantages of this method are its speed and cost. However, a code automatization could make this method feasible to be implemented in real life application. To solve the cost problem, it is possible to implement a normal downconverter and obtain a phase estimation of the signal using some of the phase estimators that are already well known, as the ones described in [29]. So, the main idea behind this method could be adapted in order to obtain a different implementation method, which may overcome the problems described before.

This method was run using 11 or 3 WiFi Channels. Even though the 3 Channel approach had the advantage of being faster than the 11 Channel one, it really does not bring any significant improvement in accuracy, speed or cost when compared to other methods. So, it seems to be more logical working to improve the 11 Channel approach rather than doing the same with the 3 Channels one.

In order to improve this method, four main paths could be followed:

- i. A first way of significantly improving the speed of this method would be forcing the emitter to scan all the WiFi Channels available, that is, forcing it to actively scan the available WiFi networks [31]. That means that it will send requests in all the 11 WiFi Channels automatically, and these frames could be captured and used to locate. Depending on the system speed, if it is fast enough, it could listen to the transmission and compute the phase difference while the next channel frame it is being captured, or

the signals propagating through the air could be captured and stored, and after the scanning is finished, the locations calculus could be done.

The implementations detail may vary, but the main idea is using this command available in the 802.11 standard in order to force the emitter to automatically change the carrier frequency and so, being able to compute the distance estimation in a shorter time.

- ii. A second way would be using a different physical layer specification present in the 802.11 standard: the FHSS [32]. Until this point, the physical layer specification used was DSSS, but there are more, and FHSS specification is interesting because it is based in a pseudorandom frequency hopping, which means that the carrier frequency jumps needed in this algorithm to do the sweep are done automatically in the physical layer (Figure 5.1). That means that the algorithm may easily be speeded up using this approach. However, this solution might not be really feasible because this physical layer specification is already obsolete.

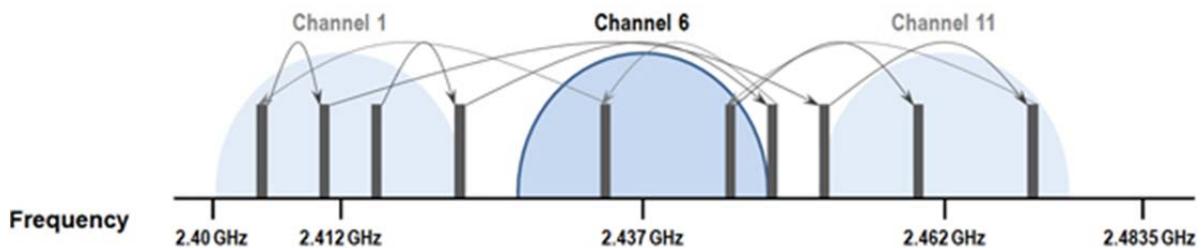


Figure 5.1 - Comparison of FHSS and DSSS specifications

- iii. A third way would be using another physical layer specification: OFDM , pp 1442-1488
 - iv. [33]. It is based in the transmission of 52 different subcarriers (an illustration with only four subcarriers in in Figure 5.2), which are sent in time during a limited time (known as T_{FFT} , between 3.2 and 12.8 μs) plus a guard interval that protects the signal against ISI, so it can be possible to obtain the phase difference between the different subcarriers within one WiFi Channel and so achieving a location estimation. Since only one WiFi channel would be used, the measuring process would be faster.
 - v. Finally, a fourth way to increase the speed would be using only a WiFi Channel and filtering it in different frequencies, using a filter bank (Figure 5.3): that way, a similar result to the OFDM proposed method would be achieved, since the signal would be split into different frequencies “subcarriers”.
- FFT: This method would be the ideal one, if it were not for the location problems. It is really fast, it is feasible and the accuracy given in the cable setup is good. So, it seems to be a good idea wondering whether it is possible to overcome the problems in any way, since by doing that, a good alternative for location is obtained. If the problems in location were due to multipath propagation, some possible ways to overcome this problem could be:
- i. Using spatial diversity techniques, such as RAKE receptors [24],which are specifically designed to be used in channels where multipath propagation is present.
 - ii. Using a dynamic filter in the receptor. Since the signal to be received is well known (the synchronization preamble is always the same shape of signal), the channel impulse response can be estimated with the received signal, and so, a dynamic filter to prevent multipath could be implemented using this information.

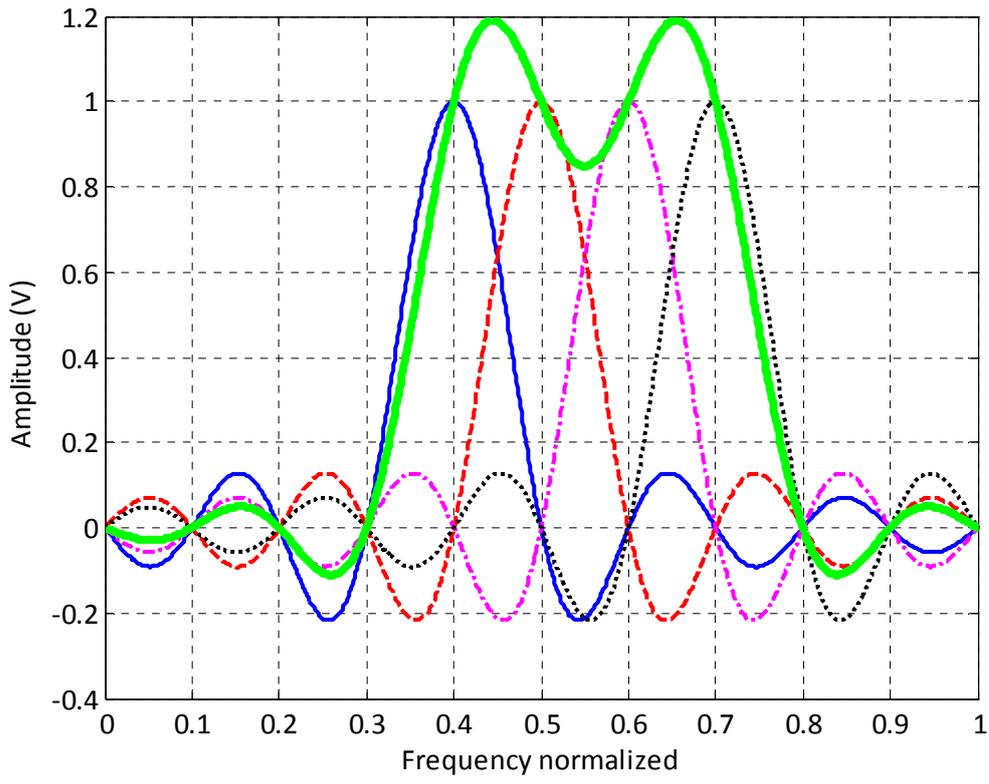


Figure 5.2 – OFDM four-subcarriers spectrum

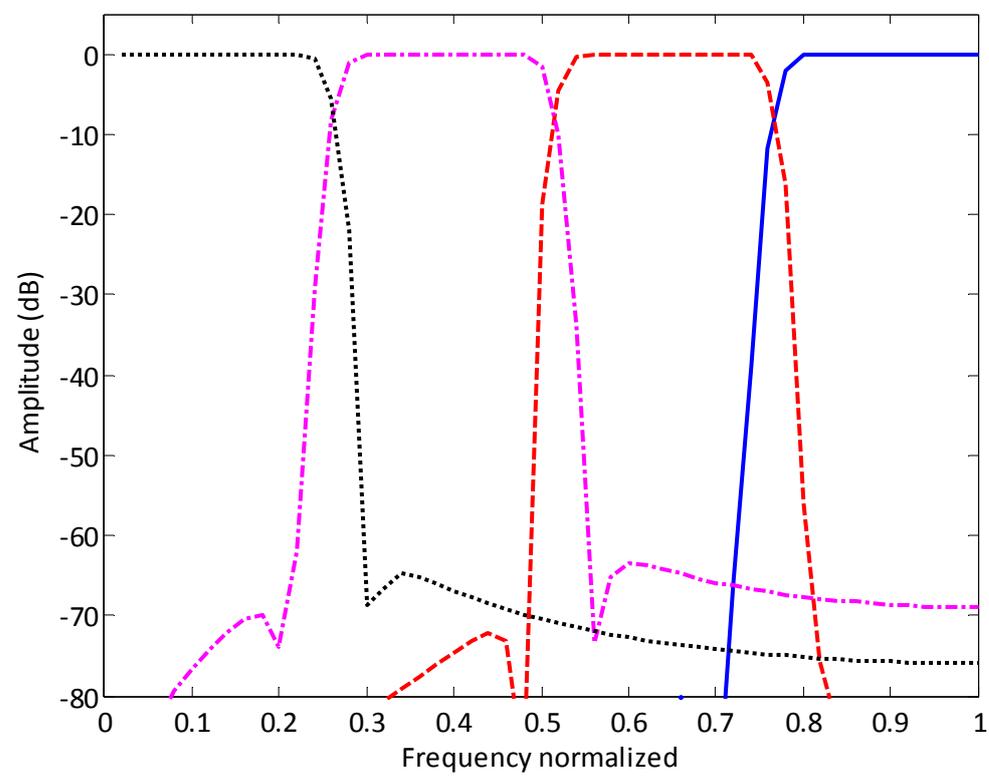


Figure 5.3 - Filter bank transfer function

- iii. Also, another solution could be implementing this method with closely locate antennas, but that would increase significantly the cost of the approach as well.
- Power level: This method, which has already been deeply studied and developed, shows that in a LOS environment, with estimated parameters, can locate surprisingly accurately. It is quite fast and cheap, and the hardware necessary is a power level meter. However, the differences observed between WiFi channel measurements seem to indicate that the parameters estimation will play a major role on the system accuracy, and so, any change in the environment that brings a change in the propagation conditions will, surely, have an impact on the location accuracy.
- IF Cross Correlation: It is already known that RF cross correlation can yields good results in LOS environments, however, the cost of operating in the 2.4 GHz band are really high and so, this approach was proposed in order to obtain a tradeoff between resolution and cost. And that tradeoff was obtained, since the highest accuracy in absolute terms is obtained here, but the main limitation is the periodicity due to the carrier wavelength. That causes that this method cannot be used to locate, unless it is combined with another method whose error is smaller than the unambiguity range: then, it would be a great way to improve fine location. However, the information provided by this method can also be used in order to obtain some additional information that can be used in order to help locating with another method. A speed estimator that took advantage of the high accuracy of this method could be implemented, for example, and it might be helpful for locating purposes. Also, it would be interesting improving the modulus operations and slope detections, not only in this approach, but in all that are periodical, since a significant improvement in that part of the algorithm would increase accuracy. Thus, a better implementation of the unwrapping algorithm will surely yield a significant improvement in the accuracy obtained in every method that depends on a periodical measure to locate.
- Sinusoid-IF: As with the before method, the low unambiguity range is a problem, but it can both be used together with another method whose error is slower than the unambiguity range of this method or be used to obtain side information, such as speed, that might be helpful to improve the location estimation. Also, the envelope detector can be improved. Four different envelope detectors were proposed, and the theoretically worst was implemented, and yet the results were accurate. Subsequently, implementing another one might increase the accuracy of this method.

BIBLIOGRAPHY

- [1] "What is a GPS? How does it work?" The Library of Congress [Online]. Available <http://www.loc.gov/rr/scitech/mysteries/global.html> (Accessed in 12/06/2014)
- [2] A. de Angelis et al, "Experimental Radio indoor positioning systems based on Round-trip Measurement" in *Advances in measurements systems*, Chapter 8, [Online]. Available at www.intechopen.com (Accessed in 12/06/2014)
- [3] Zwirello L et al, "UWB Localization System for Indoor Applications: Concept, Realization and Analysis" in *Journal of Electrical and Computer Engineering*, Volume 2012, Article ID 849638, Hindawi Publishing Corporation
- [4] J. Nilsson et al, "Signal processing issues in indoor positioning by ultra wide band radio aided inertial navigation", in *17th European Signal Processing Conference*, Glasgow, 2009
- [5] H. Reddish et al, "An improved Time-of-Arrival Estimation for WLAN-Based Local Positioning" in *Communications Systems Software and Middleware*, 2007, pp 1-5
- [6] C. Wong et al, "Using WLAN infrastructure for Angle-of-Arrival indoor user location", in *Vehicular Technology Conference*, 2008, Calgary, pp 1-5
- [7] R. Zekavat, R. Buehrer, "RF Fingerprinting location techniques" in *Handbook of Position Location: Theory, Practice and Advances*, Wiley-IEEE Press, 2011, pp 487-520
- [8] S. Schwarzer et al, "Precise distance measurement with IEEE 802.15.4 (ZigBee) devices" in *IEEE Wireless and Radio Symposium*, 2008, pp 779-782
- [9] B. Sklar, *Digital Communications, fundamentals and applications*, 2nd ed, Prentice Hall, 2004, pp 30-33
- [10] B. Sklar, *Digital Communications, fundamentals and applications*, 2nd ed, Prentice Hall, 2004, pp 944-1011
- [11] T. Rappaport, *Wireless Communications, principles and practice*, 2nd ed, Prentice Hall, 2002, pp 178-179
- [12] J. Proakis, *Digital Communications*, 4th ed, McGraw Hill, 2001, pages 160-163
- [13] T. Rappaport, *Wireless Communications, principles and practice*, 2nd ed, Prentice Hall, 2002, pp 205-208
- [14] T. Rappaport, *Wireless Communications, principles and practice*, 2nd ed, Prentice Hall, 2002, pp 197-210
- [15] M. Weber et al, "Comparison of various methods for indoor RF fingerprinting using leaky feeder cable", in *7th Workshop on Position Navigation and Communication*, 2010, pp 291-298
- [16] IEEE Global History Network, "Wireless LAN 802.11 Wi-Fi" [Online] Available: http://www.ieeeahn.org/wiki/index.php/Wireless_LAN_802.11_Wi-Fi (Accessed in 12/06/2014)
- [17] IEEE Standards Assotiation [Online] Available: <http://standards.ieee.org/about/get/802/802.11.html> (Accessed in 12/06/2014)
- [18] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" in *IEEE Standard 802.11-2012*, IEEE, New York, NY, USA, 1999-2012, part 16, pp 1504-1535
- [19] Agilent Technologies, Appl Note 1380-1, pp 10-12
- [20] J. Proakis, *Digital Communications*, 4th ed, McGraw Hill, 2001, pp 797-805
- [21] Huber & Suhner, datasheet RF Coaxial Cable type RG_214_/U-01 [Online] Available at www.hubersuhner.com (Accessed in 12/06/2014)
- [22] Rohde & Schwarz, Oscilloscope R&S RTO 1022, datasheet [Online] available: http://www.rohde-schwarz.com/en/product/rto-products_63492-10790.html (Accessed in 12/06/2014)
- [23] Linksys, Wireless Broadband Router WRT54GL, datasheet [Online] available at www.linksys.com (Accessed in 12/06/2014)

- [24] J. Proakis, *Digital Communications*, 4th ed, McGraw Hill, 2001, pp 758-769
- [25] VARIA, 5 dBi antenna, datasheet [Online] Available <http://varia-store.com/Antennas/Antennas-2-4-2-5GHz/Antenne-Magnet-Foot-2-4-2-5GHz/VARIA-magnet-foot-with-5dBi-antenna-RP-SMA-connector::612.html> (Accessed in 12/06/2014)
- [26] E. O. Brigham, "Leakage Reduction" in *The Fast Fourier Transform*, Prentice Hall, 1974, pp 140-146
- [27] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", in *Proc. IEEE*, Vol. 66, no. 1, Jan 1978, pp.51-83
- [28] B. Sujak *et al*, "Indoor propagation channel models for WLAN 802.11b at 2.4 GHz ISM band" in *Conference on applied Electromagnetics*, 2005, Johor
- [29] R. S. Lawrence *et al*, "Digital methods for the extraction of phase and amplitude information from a modulated signal" in *Journal of Research of the National Bureau of Standards – D. Radio Propagation*, Vol 65D, No 4 July-August 1961, pp 351-356
- [30] R. H. Barker, "Group Synchronizing of Binary Digital Sequences" in *Communication Theory*, Butterworth, London, pp. 273–287.
- [31] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" in *IEEE Standard 802.11-2012*, IEEE, New York, NY, USA, 1999-2012, pp 978-980
- [32] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" in *IEEE Standard 802.11-2012*, IEEE, New York, NY, USA, 1999-2012, pp 1442-1488
- [33] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" in *IEEE Standard 802.11-2012*, IEEE, New York, NY, USA, 1999-2012, pp 1583-1630

APPENDIX 1: ANTENNA CALIBRATION PROCESS

The proposed steps are the following: the antennas are placed side-by-side, at a short distance from the emitter (but out of the near field of the antenna) and are connected with two identical cables to the oscilloscope, whose propagation speed is known (Section 3.2.1). Afterwards, the distance of the emitter respect to the antennas is computed; in this case, it does not matter which total distance is used, but the result must be $d_1 \approx d/2$, independently on the total distance d used. If the antennas are equal, omnidirectional and everything is working as it should, the result before should be obtained, but if something is failing, anything else will be obtained and so, it will be necessary to determine which part of the system is failing.

This test was done to the antennas in order to determine whether there is an important phase shift between them or not. The chosen antennas are [25] and their results (Figure 0.1) when an imaginary distance of $d = 4m$ is used to measure the phase shift between antennas is of only 7 cm (since the returned distance estimation the algorithm gives is $d_1 = 2.0723 m$). This is a relative error of 1.8%, thus the phase shift between antennas can be ignored.

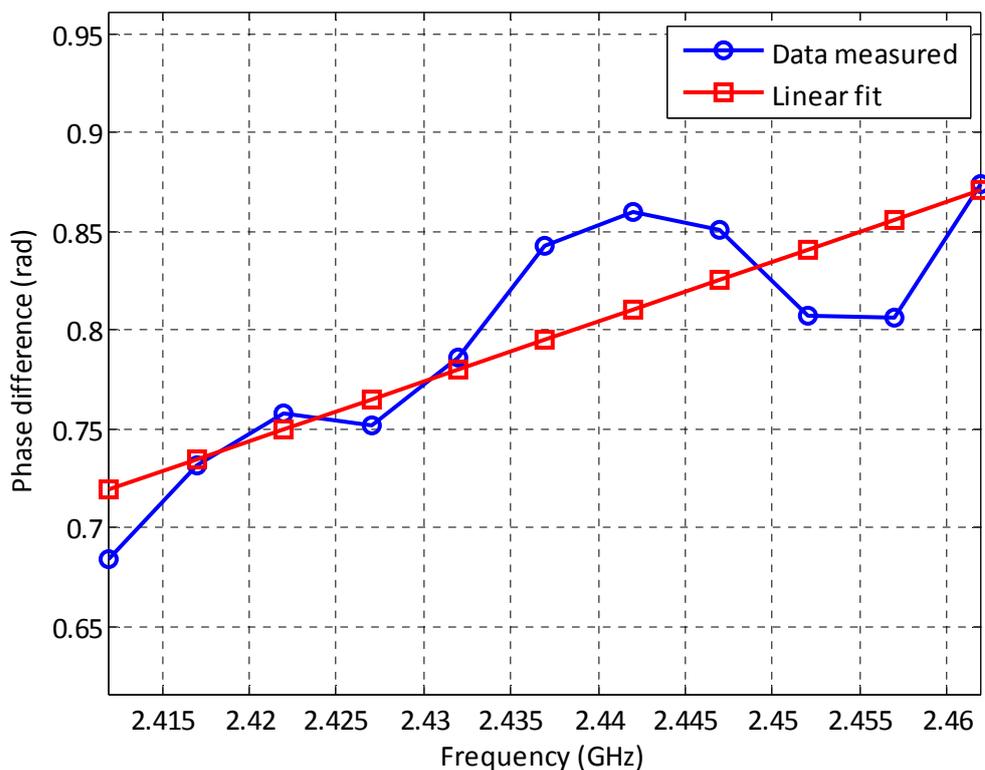


Figure 0.1 – Phase slope offset in WiFi antennas

When performing those measurements, the modulus operation performed in the calculations sometimes introduced wrong jumps in the phase difference, so a π shift is being computed and added to the algorithm in order to eliminate partly this problem. However, this is a solution valid only with one jump: in order to get the right phase, this phase must be unwrapped.

In case that there is a phase shift between antennas, whatever reason it is due, another way to proceed is measure that phase shift in different frequencies and propagation conditions, so that it is possible to check whether that shift depends only on frequency or not, and if it did, it would be

possible to add those phase terms to the algorithm, so that it corrects the phase shift due to the antennas and so, obtain a measure which would be antenna-independent. Also, this method could be used to compute the constant phase shifts in an empirical form in case that cables with different propagation speeds and/or lengths are used, and in general, to test any phase shift introduced by the system. The ideal situation is that this correction is not needed and it should be aimed at when the system is being designed, since it will be easier and more accurate (in the end, there is no warranty that the phase shift measurements performed remain still with time and changing of environmental conditions)

APPENDIX 2: FFT METHOD SPEED

One of the main advantages of this method is the fact that no frequency sweep has to be performed in order to get the distance estimation, and that means that the time cost of this method is going to be significantly small, because with only a measurement process, a distance estimation is obtained. In order to get a first glance of this, a time measure has been done to test what is the time it really takes this setup to work.

The measure has been done as follows: the full algorithm has been computed, which includes roughly two parts: the oscilloscope setup and the data extraction and calculation. So, if the calculi are repeated ten times, there will be one oscilloscope setup and ten data extraction and calculations.

Computing fifty times the full algorithm, it will be possible to obtain good statistics about the time spent in calculations. The results are:

- a) The time used in data extractions and calculations has a mean of 15.8 ms, a median of 11.4 ms and a standard deviation of 10.3 ms. It is possible to see that the measures are expected to be in the 10 – 100 ms range.
- b) The time used to set up the Oscilloscope is quite high, compared with the time used in data extractions and calculations. The time mean is 1.1265 s, the median is 1.1147 s and the standard deviation is 86.6 ms, so this time is the most expensive, in time cost.

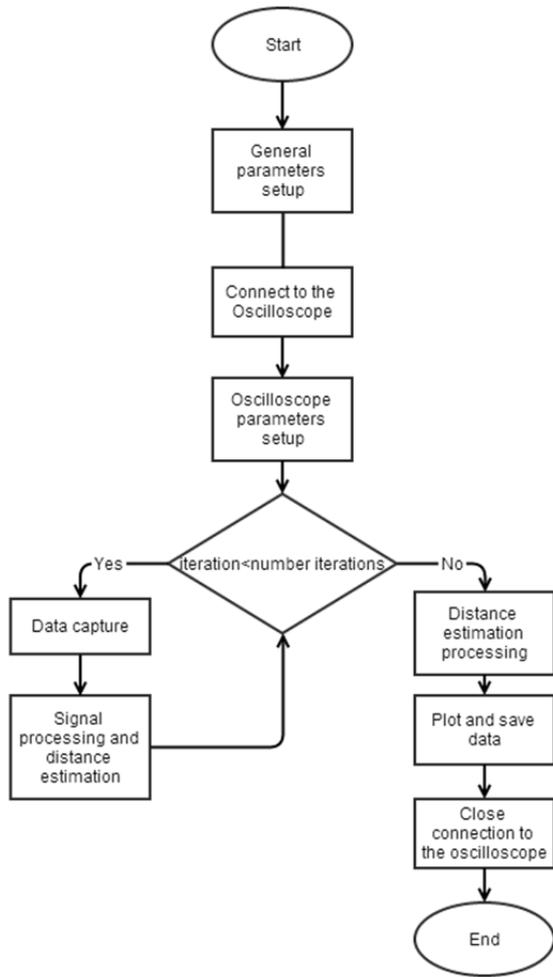
The results indicate that the FFT algorithm, including the data extraction, processing and distance calculation, takes a very short time to be performed. This can be used to, once the oscilloscope is setup, to perform various measurements and so, averaging the results obtained in order to increase the accuracy, at a little cost of time.

APPENDIX 3: CODE STRUCTURE EXPLANATION

In this and the following appendixes, the code used is shown. In all the approaches, the main changes were found in the signal processing part, so a big part of the code remains the same in all cases.

The general structure description of the code used, which can be seen in the figure below, is the next:

1. First, the general parameters are set up, like the WiFi Channels which is being used, the sampling frequency and intermediate frequency to use and depending on the method used to locate, the parameters that it may need in order to work.
2. Secondly, a connection to the oscilloscope is opened. The connection is checked, in order to test whether there is a connectivity problem or not.
3. Next, the oscilloscope configuration parameters are sent to the oscilloscope, and it is set up according to what it is needed.
4. After everything is set up, the main loop starts, and for a number of iterations which was defined in the general parameters, the computer captures the data from the oscilloscope and processes them so that a distance estimation is obtained from them.
5. After the main loop, the distance estimations obtained are averaged according to the method that was set up in the general parameters (mean or median).
6. Next, the data is plot and/or saved.
7. Finally, the connection to the oscilloscope is closed and the program ends.



APPENDIX 4: CODE OF FREQUENCY SWEEP APPROACH IN A LOS SETUP

```
function []=Optimized_LOS()
% Function that computes the distance estimation using the frequency sweep
% approach in a LOS environment.
% Created: April 2014, Juan Parras Moral
% IMPORTANT: Distance estimation can be d1 or d2. Check just in case!

%% Disconnect and delete all instrument objects
instrreset
clear all

%% Set up the general parameters needed

pl=1; %Logical value used to plot or not the graphs
Nrep=10; %Number of iterations in the main loop
fd=5e6; %Frequency gap between channels
freq=2.412e9:fd:2.462e9; %WiFi Channels central freq
c1=3e8; %Propagation speed channel 1
c2=3e8; %Propagation speed channel 2
d=6; %Total distance separation between antennas
stat=1; %Statistical used to average: 1 is median, 2 is mean
name=input('Enter name of the file to be saved (use '' ) ');
th=2e-4; %Noise threshold used

%Oscilloscope parameters setup

% Physical channels (input channels)
% First entry: Coupling = 50 Ohm and Trigger on this channel
nChannelNo = [1 2];
nNofActiveCHs = length (nChannelNo); % Number of active channels

% Define number of samples (record length)
nNofSamples = 2e4;

% Set carrier frequency (center frequency)
nCarrierFrequency = 2.395e9;

% Set sample rate
nSampleRate = 250e6;

% Set parameters for IQ mode
sInputType = 'REAL';
sInputMode = 'RFIF';
sSideband = 'NORMal';
nRelBW = 0.4;

%% Connection to the oscilloscope

% Create a VISA connection using HiSLIP protocol to the specified IP
address

IPAddress = '169.254.82.246';
RTO = visa('ni', ['TCPIP::' IPAddress '::inst0::INSTR']);
```

```

% increase the buffer size to, e.g., transport IQ data (in bytes)
RTO.InputBufferSize = 40e6;

% Open the instrument connection
fopen(RTO);

% Clear error queue
systemErrorQueue(RTO);

% Query and plot the connected instrument ID String
fprintf(RTO, '*IDN?');
disp(['    ID: ' fscanf(RTO)])

% Activate View-Mode in Remote Mode
fprintf(RTO, 'SYST:DISP:UPD ON');

%% Oscilloscope parameters setup

% Preset the RTO and wait till action is finished
fprintf(RTO, '*CLS');
fprintf(RTO, '*RST; *OPC?');
fscanf(RTO);

% Activate channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':STATE ON']);
end

% Set Coupling to 50 Ohm
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':COUP DC']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':COUP DC']);

% Set vertical scale
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':SCAL 1e-3']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':SCAL 1e-3']);

% Set horizontal scale and position
fprintf(RTO, 'ACQ:POIN:AUTO RES'); % resolution remains constant
fprintf(RTO, 'ACQ:POIN:MAX 2e6'); % max record length = 2 MSA
fprintf(RTO, 'TIM:SCAL 40e-6'); % horizontal scale = 400 us
fprintf(RTO, 'ACQ:RES 200e-12'); % sample rate = 5GSa/s
fprintf(RTO, 'TIM:REF 10'); % Ref. position = 10%

% Set up trigger (source, type, level and mode)
fprintf(RTO, 'TRIGger:SOURce CHAN' num2str(nChannelNo(1)));
fprintf(RTO, 'TRIGger:LEV 4.5e-3');
fprintf(RTO, 'TRIGger:LEV:NOISE MAN'); % trigger hysteresis = manual
fprintf(RTO, 'TRIGger:LEV:NOISE:MODE ABS'); % trigger hysteresis mode
fprintf(RTO, 'TRIGger:LEV:NOISE:ABS 10e-6'); % trigger hyst. = 10 uV
fprintf(RTO, 'TRIGger:MODE NORM');
fprintf(RTO, 'TRIGger:TYPE EDGE');

% Check error queue
[~] = systemError(RTO);

% Activate IQ mode
fprintf(RTO, 'IQ:STATe ON');

```

```

%Set the input signal, input mode and sideband for active channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPType '
sInputType]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPMode '
sInputMode]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:SBRF '
sSideband]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:CFRequency '
num2str(nCarrierFrequency)]);
end

% Record Length to be used
fprintf(RTO, ['IQ:RLEN ' num2str(nNofSamples)]);
[~] = systemError(RTO);

% Set the correct sampling rate
fprintf(RTO, ['IQ:SRATE ' num2str(nSampleRate)]);
[~] = systemError(RTO);

% Set the relative Bandwidth
fprintf(RTO, ['IQ:RBwidth ' num2str(nRelBW)]);
[~] = systemError(RTO);

%% Main loop

%The phase difference is contained in aux

aux=0;
for i=1:length(freq)
    display(['Processing channel ' num2str(i)]);
    input('Press enter once the freq is adjusted');
    for j=1:Nrep;
        aux(i,j)=phase_calc(RTO,nNofSamples,pl,stat,th);
    end
end
save(name);

%% Distance estimation processing

% Two different methods are used: 1 and 2 use all data in all frequencies,
% whereas methods 3 and 4 only use data in selected frequencies
% Also, methods 1 and 3 use data wrap, whereas 2 and 4 use them unwrap

%Method 1 used: each phase measurement at each frequency will be used
for j=1:Nrep
    display(['Measure ' num2str(j) ' out of ' num2str(Nrep)]);
    phase=aux(:,j);
    phase=phase';
    display('11 Channel results');

    %Method 1: Wrap
    au=polyfit(freq/1e6,phase,1);
    m=au(1)*1e-6;
    c0=3e8;
    a1=c1/c0;
    a2=c2/c0;
    d1=(m*a1*a2*c0/(2*pi)+d*a1)/(a1+a2);
    display(['Results wrap: m= ' num2str(m) ';d1= ' num2str(d1)]);

```

```

%Method 2: unwrap
au2=polyfit(freq/1e6,unwrap(phase),1);
m2=au2(1)*1e-6;
d12=(m2*a1*a2*c0/(2*pi)+d*a1)/(a1+a2);
display(['Results unwrap: m= ' num2str(m2) ';d1= ' num2str(d12)]);

% 3 Channels calculations
display('3 Channel results');
freq2=[freq(1),freq(6),freq(11)];
phase2=[phase(1),phase(6), phase(11)];

%Method 1: Wrap
au3=polyfit(freq2/1e6,phase2,1);
m3=au3(1)*1e-6;
d13=(m3*a1*a2*c0/(2*pi)+d*a1)/(a1+a2);
display(['Results wrap: m= ' num2str(m3) ';d1= ' num2str(d13)]);

%Method 2: unwrap
au4=polyfit(freq2/1e6,unwrap(phase2),1);
m4=au4(1)*1e-6;
d14=(m4*a1*a2*c0/(2*pi)+d*a1)/(a1+a2);
display(['Results unwrap: m= ' num2str(m4) ';d1= ' num2str(d14)]);

vector_d1(j)=d1;
vector_d12(j)=d12;
vector_d13(j)=d13;
vector_d14(j)=d14;
end

%% Data plot and save

% figure(2);
% plot(freq,phase,'b',freq,unwrap(phase),'k',freq,freq*au(1)*1e-
6+au(2),'r',freq,freq*au2(1)*1e-6+au2(2),'g'),l=legend('Data
measured','Data measured unwrap','Linear fit wrap','Linear fit
unwrap');title(['Phase difference d=' num2str(d) 'm, 11
Channel']),xlabel('Frequency (Hz)');ylabel('Phase difference (rad)');
% figure(3);
% plot(freq2,phase2,'b',freq2,unwrap(phase2),'k',freq2,freq2*au3(1)*1e-
6+au3(2),'r',freq2,freq2*au4(1)*1e-6+au4(2),'g'),l=legend('Data
measured','Data measured unwrap','Linear fit wrap','Linear fit
unwrap');title(['Phase difference d=' num2str(d) 'm, 3
Channel']),xlabel('Frequency (Hz)');ylabel('Phase difference (rad)');

save(name);

%% Close oscilloscope connection

% Close instrument connections
fclose(RTO);
delete(RTO);

end

function[phase_d]=phase_calc(RTO,nNofSamples,pl,stat,th)

```

```

% Capture data, avoiding noise!

aux=1;

while(aux)

[ iqDataCh1, iqDataCh2, nTimeAcqIQDataCh1, nTimeReadIQDataCh1 ] =
ReadRTOIQBin2(RTO, 'Ch1', nNofSamples);

ich1=real(iqDataCh1);
qch1=imag(iqDataCh1);
ich2=real(iqDataCh2);
qch2=imag(iqDataCh2);

% Calculate phase and modulus of each channel data

abs1=abs(ich1+j*qch1);
abs2=abs(ich2+j*qch2);
phase1=unwrap(angle(ich1+j*qch1));
phase2=unwrap(angle(ich2+j*qch2));

%Compute phase difference between channels

phase_dif=phase1-phase2;
phase_dif=mod(phase_dif,2*pi); %Phase between 0 and 2pi

%Average distances using the proper statistic

if stat==1
    phase_d=median(phase_dif)
else
    phase_d=mean(phase_dif)
end

% Plot if that option is active

t=0:1/250:(length(abs1)-1)/250;
if pl
    figure(1);
    subplot(411);plot(t,phase_dif); title('Phase difference');xlabel ('Time
(us)');ylabel('Phase (rad)');
    subplot(412);plot(t,phase1,'b',t,phase2,'g');title('Phase each
channel');xlabel ('Time (us)');ylabel('Phase (rad)');
    subplot(413);plot(t,abs1,'b',t,abs2,'g');title('Modulus each
channel');xlabel ('Time (us)');ylabel('Tension (V)');

    subplot(414);plot(t,real(ich1+j*qch1), 'b',t,real(ich2+j*qch2), 'g',t,th*ones
(length(t),1), 'r');title('Real signals');xlabel ('Time
(us)');ylabel('Tension (V)');
end

if(mean(abs1))>th %Don't accept a measure which is noise!!
    aux=0;
end

end

end

```

```

function [ iqData1, iqData2, tAcq, tReadIQ ] = ReadRTOIQBin2( instrHandle,
chNr, nsamples )
%ReadRTOwfmbin Summary of this function goes here
% Detailed explanation goes here

%% Check number of arguments
error(nargchk(1, 3, nargin, 'struct'));    %#ok mlint

% Initialize optional parameters
if ~exist('chNr', 'var'),
    chNr = 'Ch1';
end;

if ~exist('numAcq', 'var'),
    numAcq = 10;
end;

switch lower(chNr)
    case 'ch1'
        chReadCmd = 'CHAN1:IQ:DATA:VALues?';

    case 'ch2'
        chReadCmd = 'CHAN2:IQ:DATA:VALues?';

    case 'ch3wfml'
        chReadCmd = 'CHAN3:IQ:DATA:VALues?';

    case 'ch4wfml'
        chReadCmd = 'CHAN4:IQ:DATA:VALues?';

    otherwise
        error('Illegal channel name');

end

%% Select binary format of the transmission [ASC/UINT/REAL]
fprintf(instrHandle, 'FORM REAL,32');
numAcq=2; %Always, to have 2 channels read
fprintf(instrHandle, 'SING;*OPC?');
fscanf(instrHandle);

sBinaryFormatString = 'float';
nSizeType = 4;

tAcq = 0;
tReadIQ = 0;
%     fprintf(instrHandle, 'CHAN1:IQ:HIST:STAT ON');
%     fprintf(instrHandle, 'CHAN2:IQ:HIST:STAT ON');

% *** Initiate acquisition and wait for completion ***
tic
%fprintf(instrHandle, 'SING;*WAI');
fprintf(instrHandle, 'SING;*OPC?'); %Starts single acquisition and waits
till it is completed
fscanf(instrHandle);
tAcq = tAcq + toc;

```

```

for i=1:numAcq

    % *** Get IQ data ***
    tic
    if i==1
        ch='CHAN1';
    else
        ch='CHAN2';
    end

%     fprintf(instrHandle, [ch ':IQ:DATA:HIST 0,' num2str(-nsamples)]);
    fprintf(instrHandle, [ch ':IQ:DATA:VALues?']);

    % check the return beginning with a hash '#'
    sStartIndicator = fread(instrHandle,1,'char');
    if sStartIndicator ~= '#'; fprintf('oops!\n'); end;

    % check the length of the length field in units
    nLengthOfLengthfield = fread(instrHandle,1,'char');
    nLengthOfLengthfield = str2double(char(nLengthOfLengthfield));

    % check the length of the data record
    nBlockLength = fread(instrHandle, nLengthOfLengthfield , 'char');
    nBlockLength = str2double(char(nBlockLength)) / nSizeType;

    % to make this work, the endianness endian must be considered!
    % the RTO supports little endian byte order
    iqData2 = fread(instrHandle, nBlockLength, sBinaryFormatString);
    iqData2 = iqData2(1:2:end) + 1i*iqData2(2:2:end);

    tReadIQ = tReadIQ + toc;
    if i==1
        iqData1=iqData2;
    end
end

end

function [nErrorNr] = systemErrorQueue(instrHandle)
    nErrorNr = -1;
    while nErrorNr < 0
        nErrorNr = systemError(instrHandle);
    end
    disp('Error queue empty')
end

function [nErrorNr] = systemError(instrHandle)
    fprintf(instrHandle, 'SYSTEM:ERROR?');
    sAux = fscanf(instrHandle);
    C = strsplit(sAux, ',');
    sErrorNr = C{1};
    nErrorNr = str2double(sErrorNr);
    disp(sAux)
end

```

APPENDIX 5: CODE OF FFT APPROACH, LOS SETUP

```
function []=Optimized_FFT_LOS()

% Function that computes the distance estimation using the FFT approach in
% a LOS environment.
% Created: April 2014, Juan Parras Moral

%% Disconnect and delete all instrument objects

instrreset
clear all

%% Set up the general parameters needed

Nrep=50; %Number of iterations in the main loop
Nsamples_w=2000; %Number of samples used to windowing
fs=250e6; %Sampling frequency
fc=2.412;%Wifi channels: 1 - 2.412, 6 - 2.437, 11 - 2.462
fc=fc*1e9-2.395e9; %Frequency in the IF
B=15e6; %Bandwidth of the signal used to compute phase difference
c1=3e8; %Propagation speed channel 1
c2=3e8; %Propagation speed channel 2
th=2e-3; %Noise threshold

d=6; %Total distance difference between antennas
name=input('Enter name of the file to be saved (use '' ) ');
stat=2; %Statistical used to average: 1 is median, 2 is mean

% Oscilloscope parameters setup

% Physical channels (input channels)
% First entry: Coupling = 50 Ohm and Trigger on this channel
nChannelNo = [1 2];
nNofActiveCHs = length (nChannelNo); % Number of active channels

% Define number of samples (record length)
nNofSamples = 2e4;

% Set carrier frequency (center frequency)
nCarrierFrequency = 2.395e9;

% Set sample rate
nSampleRate = 250e6;

% Set parameters for IQ mode
sInputType = 'REAL';
sInputMode = 'RFIF';
sSideband = 'NORMal';
nRelBW = 0.4;

%% Connection to the oscilloscope

% Create a VISA connection using HiSLIP protocol to the specified IP
address

IPAddress = '169.254.82.246';
```

```

RTO = visa('ni', ['TCPIP::' IPAddress '::inst0::INSTR']);

% increase the buffer size to, e.g., transport IQ data (in bytes)
RTO.InputBufferSize = 40e6;

% Open the instrument connection
fopen(RTO);

% Clear error queue
systemErrorQueue(RTO);

% Check the connection
fprintf(RTO, '*IDN?');
disp([' ID: ' fscanf(RTO)])

% Activate View-Mode in Remote Mode
fprintf(RTO, 'SYST:DISP:UPD ON');

%% Oscilloscope parameters setup

% Preset the RTO and wait till action is finished
fprintf(RTO, '*CLS');
fprintf(RTO, '*RST; *OPC?');
fscanf(RTO);

% Activate channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':STATE ON']);
end

% Set Coupling to 50 Ohm
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':COUP DC']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':COUP DC']);

% Set vertical scale
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':SCAL 1e-3']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':SCAL 1e-3']);

% Set horizontal scale and position
fprintf(RTO, 'ACQ:POIN:AUTO RES'); % resolution remains constant
fprintf(RTO, 'ACQ:POIN:MAX 2e6'); % max record length = 2 MSa
fprintf(RTO, 'TIM:SCAL 40e-6'); % horizontal scale = 400 us
fprintf(RTO, 'ACQ:RES 200e-12'); % sample rate = 5GSa/s
fprintf(RTO, 'TIM:REF 10'); % Ref. position = 10%

% Set up trigger (source, type, level and mode)
fprintf(RTO, ['TRIGger:SOURce CHAN' num2str(nChannelNo(1))]);
fprintf(RTO, 'TRIGger:LEV 4.5e-3');
fprintf(RTO, 'TRIGger:LEV:NOISE MAN'); % trigger hysteresis = manual
fprintf(RTO, 'TRIGger:LEV:NOISE:MODE ABS'); % trigger hysteresis mode
fprintf(RTO, 'TRIGger:LEV:NOISE:ABS 10e-6'); % trigger hyst. = 10 uV
fprintf(RTO, 'TRIGger:MODE NORM');
fprintf(RTO, 'TRIGger:TYPE EDGE');

% Check error queue
[~] = systemError(RTO);

```

```

% Activate IQ mode
fprintf(RTO, 'IQ:STATE ON');

%Set the input signal, input mode and sideband for active channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPType '
sInputType]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPMode '
sInputMode]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:SBRF '
sSideband]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:CFrequency '
num2str(nCarrierFrequency)]);
end

% Record Length to be used
fprintf(RTO, ['IQ:RLEN ' num2str(nNofSamples)]);
[~] = systemError(RTO);

% Set the correct sampling rate
fprintf(RTO, ['IQ:SRATE ' num2str(nSampleRate)]);
[~] = systemError(RTO);

% Set the relative Bandwidth
fprintf(RTO, ['IQ:RBWidth ' num2str(nRelBW)]);
[~] = systemError(RTO);

%% Main loop

% aux1 contains distance estimation wrap, aux2 contains the unwrap ones
for j=1:Nrep;
    display(['Processing iteration ' num2str(j) ' of ' num2str(Nrep)]);
    [phase_d(j,:) aux1(j)
aux2(j)]=d1_calc(RTO,nNofSamples,Nsamples_w,fs,fc,B,d,c1,c2,th);
end

%% Distance estimation processing

if stat==1
    d1=median(aux1);
    d12=median(aux2);
else
    d1=mean(aux1);
    d12=mean(aux2);
end

%% Plot and save data

plot(1:length(aux1),aux1,'b',1:length(aux2),aux2,'r');
display(['Wrap: Mean d1 = ' num2str(mean(aux1)) ' Median = '
(num2str(median(aux1))) ' Std desviation = ' num2str(sqrt(var(aux1)))]);
display(['Unwrap: Mean d12 = ' num2str(mean(aux2)) ' Median = '
(num2str(median(aux2))) ' Std desviation = ' num2str(sqrt(var(aux2)))]);
save(name, 'aux1','aux2','d1','std','fc','B','Nsamples_w','phase_d');

%% Close oscilloscope connection

% Close instrument connections

```

```

fclose(RTO);
delete(RTO);

end

function [phase_d, d1, d12] = d1_calc(RTO, nNofSamples, Nsamples_w, fs, fc, B, d, c1, c2, th)

%Capture data, avoiding noise

i=0;
while i==0
    [iqDataCh1, iqDataCh2, nTimeAcqIQDataCh1, nTimeReadIQDataCh1] =
    ReadRTOIQBin2(RTO, 'Ch1', nNofSamples);
    x1=iqDataCh1(:);
    if abs(x1'*x1)>th
        i=1;
    end
    display('Noise detected, proceeding to repeat measurement...');
end
ich1=real(iqDataCh1);
qch1=imag(iqDataCh1);
ich2=real(iqDataCh2);
qch2=imag(iqDataCh2);
x1=real(ich1+j*qch1);
x2=real(ich2+j*qch2);

%Select the samples needed, from the midst of the vector on

x1=x1(nNofSamples/2:nNofSamples/2+Nsamples_w-1);
x2=x2(nNofSamples/2:nNofSamples/2+Nsamples_w-1);

%Using hamming window
w=hamming(Nsamples_w);
X1=fft(x1.*w);
X2=fft(x2.*w);

%Select the samples to use and compute phase difference
sample_low_limit=Nsamples_w*(fc-B/2)/fs;
sample_high_limit=Nsamples_w*(fc+B/2)/fs;
phase_1=angle(X1(sample_low_limit:sample_high_limit));
phase_2=angle(X2(sample_low_limit:sample_high_limit));
phase_d=mod(phase_2-phase_1,2*pi);%Missing minus sign somewhere

%Create a frequencies vector
freq=linspace(fc-B/2,fc+B/2,length(phase_d));

%Method 1: Wrap
a=polyfit(freq/1e6,phase_d,1);
m=a(1)*1e-6;
if c1~=c2; display('Warning: Wrong conditions in d1 calculations (c1~=c2)');end;
d1=0.5*(d+m*c1/(2*pi))

%Method 2: Unwrap
a=polyfit(freq/1e6,unwrap(phase_d),1);
m=a(1)*1e-6;

```

```

if c1~=c2; display('Warning: Wrong conditions in d1 calculations
(c1~=c2)');end;
d12=0.5*(d+m*c1/(2*pi))

end
function [ iqData1, iqData2, tAcq, tReadIQ ] = ReadRTOIQBin2( instrHandle,
chNr, nsamples )
%ReadRTOWfmBin Summary of this function goes here
% Detailed explanation goes here

%% Check number of arguments
error(nargchk(1, 3, nargin, 'struct'));    %#ok mlint

% Initialize optional parameters
if ~exist('chNr', 'var'),
    chNr = 'Ch1';
end;

if ~exist('numAcq', 'var'),
    numAcq = 10;
end;

switch lower(chNr)
    case 'ch1'
        chReadCmd = 'CHAN1:IQ:DATA:VALues?';

    case 'ch2'
        chReadCmd = 'CHAN2:IQ:DATA:VALues?';

    case 'ch3wfm1'
        chReadCmd = 'CHAN3:IQ:DATA:VALues?';

    case 'ch4wfm1'
        chReadCmd = 'CHAN4:IQ:DATA:VALues?';

    otherwise
        error('Illegal channel name');

end

%% Select binary format of the transmission [ASC/UINT/REAL]
fprintf(instrHandle,'FORM REAL,32');
numAcq=2; %Always, to have 2 channels read
fprintf(instrHandle,'SING;*OPC?');
fscanf(instrHandle);

sBinaryFormatString = 'float';
nSizeType = 4;

tAcq = 0;
tReadIQ = 0;
% fprintf(instrHandle, 'CHAN1:IQ:HIST:STAT ON');
% fprintf(instrHandle, 'CHAN2:IQ:HIST:STAT ON');

% *** Initiate acquisition and wait for completion ***
tic
%fprintf(instrHandle,'SING;*WAI');

```

```

    fprintf(instrHandle, 'SING;*OPC?'); %Starts single acquisition and waits
till it is completed
    fscanf(instrHandle);
    tAcq = tAcq + toc;

    for i=1:numAcq

        % *** Get IQ data ***
        tic
        if i==1
            ch='CHAN1';
        else
            ch='CHAN2';
        end

%         fprintf(instrHandle, [ch ':IQ:DATA:HIST 0,' num2str(-nsamples)]);
        fprintf(instrHandle, [ch ':IQ:DATA:VALues?']);

        % check the return beginning with a hash '#'
        sStartIndicator = fread(instrHandle,1,'char');
        if sStartIndicator ~= '#'; fprintf('oops!\n'); end;

        % check the length of the length field in units
        nLengthOfLengthfield = fread(instrHandle,1,'char');
        nLengthOfLengthfield = str2double(char(nLengthOfLengthfield));

        % check the length of the data record
        nBlockLength = fread(instrHandle, nLengthOfLengthfield , 'char');
        nBlockLength = str2double(char(nBlockLength)) / nSizeType;

        % to make this work, the endianness endian must be considered!
        % the RTO supports litte endian byte order
        iqData2 = fread(instrHandle, nBlockLength, sBinaryFormatString);
        iqData2 = iqData2(1:2:end) + 1i*iqData2(2:2:end);

        tReadIQ = tReadIQ + toc;
        if i==1
            iqData1=iqData2;
        end
    end

end

function [nErrorNr] = systemErrorQueue(instrHandle)
    nErrorNr = -1;
    while nErrorNr < 0
        nErrorNr = systemError(instrHandle);
    end
    disp('Error queue empty')
end

function [nErrorNr] = systemError(instrHandle)
    fprintf(instrHandle, 'SYSTem:ERRor?');
    sAux = fscanf(instrHandle);
    C = strsplit(sAux, ',');
    sErrorNr = C{1};
    nErrorNr = str2double(sErrorNr);
    disp(sAux)
end

```


APPENDIX 6: CODE OF SIGNAL POWER APPROACH, LOS SETUP

```
function []=Optimized_pow_cable()

% Function that computes the distance estimation using the power approach
% in a LOS environment.
% Created: April 2014, Juan Parras Moral

%% Disconnect and delete all instrument objectsinstrreset
clear all

%% Set up the general parameters needed

Nrep=50; %Number of iterations in the main loop
c1=0.66*3e8; %Propagation speed channel 1
c2=0.66*3e8; %Propagation speed channel 2
stat=1; %Statistical used to average: 1 is median, 2 is mean
d=6; %Total distance between antennas
alfa=input('Enter attenuation to be used'); %Attenuation constant to be
used
offset=input('Enter offset to be used'); %Attenuation offset to be used
name=input('Enter name of the file to be saved (use '' ) ');

%Oscilloscope parameters setup

% Physical channels (input channels)
% First entry: Coupling = 50 Ohm and Trigger on this channel
nChannelNo = [1 2];
nNofActiveCHs = length (nChannelNo); % Number of active channels

% Define number of samples (record length)
nNofSamples = 2e4;

% Set carrier frequency (center frequency)
nCarrierFrequency = 2.395e9;

% Set sample rate
nSampleRate = 250e6;

% Set parameters for IQ mode
sInputType = 'REAL';
sInputMode = 'RFIF';
sSideband = 'NORMal';
nRelBW = 0.4;

%% Connection to the oscilloscope

% Create a VISA connection using HiSLIP protocol to the specified IP
address

IPAddress = '169.254.82.246';
RTO = visa('ni', ['TCPIP::' IPAddress '::inst0::INSTR']);

% increase the buffer size to, e.g., transport IQ data (in bytes)
RTO.InputBufferSize = 40e6;

% Open the instrument connection
```

```

fopen(RTO);

% Clear error queue
systemErrorQueue(RTO);

% Query and plot the connected instrument ID String
fprintf(RTO, '*IDN?');
disp(['    ID: ' fscanf(RTO)])

%Activate View-Mode in Remote Mode
fprintf(RTO,'SYST:DISP:UPD ON');

%% Oscilloscope parameters setup

% Preset the RTO and wait till action is finished
fprintf(RTO,'*CLS');
fprintf(RTO,'*RST; *OPC?');
fscanf(RTO);

% Activate channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':STATE ON']);
end

% Set Coupling to 50 Ohm
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':COUP DC']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':COUP DC']);

% Set vertical scale
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':SCAL 1e-3']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':SCAL 1e-3']);

% Set horizontal scale and position
fprintf(RTO, 'ACQ:POIN:AUTO RES'); % resolution remains constant
fprintf(RTO, 'ACQ:POIN:MAX 2e6'); % max record length = 2 MSa
fprintf(RTO, 'TIM:SCAL 40e-6'); % horizontal scale = 400 us
fprintf(RTO, 'ACQ:RES 200e-12'); % sample rate = 5GSa/s
fprintf(RTO, 'TIM:REF 10'); % Ref. position = 10%

% Set up trigger (source, type, level and mode)
fprintf(RTO, ['TRIGger:SOURce CHAN' num2str(nChannelNo(1))]);
fprintf(RTO, 'TRIGger:LEV 3.5e-3');
fprintf(RTO, 'TRIGger:LEV:NOISE MAN'); % trigger hysteresis = manual
fprintf(RTO, 'TRIGger:LEV:NOISE:MODE ABS'); % trigger hysteresis mode
fprintf(RTO, 'TRIGger:LEV:NOISE:ABS 10e-6'); % trigger hyst. = 10 uV
fprintf(RTO, 'TRIGger:MODE NORM');
fprintf(RTO, 'TRIGger:TYPE EDGE');

% Check error queue
[~] = systemError(RTO);

% Activate IQ mode
fprintf(RTO,'IQ:STATe ON');

%Set the input signal, input mode and sideband for active channels
for nIndex = 1 : nNofActiveCHs

```

```

        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPType '
sInputType]);
        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPMode '
sInputMode]);
        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:SBRF '
sSideband]);
        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:CFRequency '
num2str(nCarrierFrequency)]);
end

% Record Length to be used
fprintf(RTO, ['IQ:RLEN ' num2str(nNofSamples)]);
[~] = systemError(RTO);

% Set the correct sampling rate
fprintf(RTO, ['IQ:SRATE ' num2str(nSampleRate)]);
[~] = systemError(RTO);

% Set the relative Bandwidth
fprintf(RTO, ['IQ:RBwidth ' num2str(nRelBW)]);
[~] = systemError(RTO);

%% Main loop

% att contains the attenuation differences between channels, and aux
% contains the distance estimation obtained in each iteration

aux=0;
for j=1:Nrep;
    display(['Processing iteration ' num2str(j) ' of ' num2str(Nrep)]);
    [att(j) aux(j)]=d1_calc(RTO,nNofSamples,alfa,d,c1,c2,offset);
end

%% Distance estimation processing

if stat==1
    d1=median(aux)
else
    d1=mean(aux)
end

%% Plot and save data

%plot(aux);
display(['Mean d1 = ' num2str(mean(aux)) ' Median = '
(num2str(median(aux))) ' Std desviation = ' num2str(sqrt(var(aux)))]);
save(name);
%% Close oscilloscope connections

% Close intrument connections
fclose(RTO);
delete(RTO);

end

function[att,d1]=d1_calc(RTO,nNofSamples,alfa,d,c1,c2,offset)

%Capture data, avoiding noise

```

```

th=2e-4;
k=1;
while k
    [iqDataCh1, iqDataCh2, nTimeAcqIQDataCh1, nTimeReadIQDataCh1] =
ReadRTOIQBin2(RTO, 'Ch1', nNofSamples);
    ich1=real(iqDataCh1);
    qch1=imag(iqDataCh1);
    ich2=real(iqDataCh2);
    qch2=imag(iqDataCh2);
    x1=real(ich1+j*qch1);
    x2=real(ich2+j*qch2);

    if (size(x1,1))~=1; x1=x1';end;
    if (size(x2,1))~=1; x2=x2';end;

    pow_1=x1*x1'
    if pow_1>th
        k=0
    else
        display('Noise detected');
    end
end
pow_2=x2*x2';

%Compute the attenuation difference between channels

att=10*log(abs(pow_2/pow_1))

%Compute the distance difference estimation

if c1~=c2; display('Warning: d1 measure wrong (c1~=c2)'); end;
att=att-offset;
a=10.^(att/(10*alfa));
d1=d.*(a./(1+a));

end
function [ iqData1, iqData2, tAcq, tReadIQ ] = ReadRTOIQBin2( instrHandle,
chNr, nsamples )
%ReadRTOwfmbin Summary of this function goes here
% Detailed explanation goes here

%% Check number of arguments
error(nargchk(1, 3, nargin, 'struct'));    %#ok mlint

% Initialize optional parameters
if ~exist('chNr', 'var'),
    chNr = 'Ch1';
end;

if ~exist('numAcq', 'var'),
    numAcq = 10;
end;

switch lower(chNr)
    case 'ch1'
        chReadCmd = 'CHAN1:IQ:DATA:VALues?';

    case 'ch2'

```

```

        chReadCmd = 'CHAN2:IQ:DATA:VALues?';

    case 'ch3wfm1'
        chReadCmd = 'CHAN3:IQ:DATA:VALues?';

    case 'ch4wfm1'
        chReadCmd = 'CHAN4:IQ:DATA:VALues?';

    otherwise
        error('Illegal channel name');

end

%% Select binary format of the transmission [ASC/UINT/REAL]
fprintf(instrHandle, 'FORM REAL,32');
numAcq=2; %Always, to have 2 channels read
fprintf(instrHandle, 'SING;*OPC?');
fscanf(instrHandle);

sBinaryFormatString = 'float';
nSizeType = 4;

tAcq = 0;
tReadIQ = 0;
%     fprintf(instrHandle, 'CHAN1:IQ:HIST:STAT ON');
%     fprintf(instrHandle, 'CHAN2:IQ:HIST:STAT ON');

% *** Initiate acquisition and wait for completion ***
tic
    fprintf(instrHandle, 'SING;*WAI');
    fprintf(instrHandle, 'SING;*OPC?'); %Starts single acquisition and waits
till it is completed
fscanf(instrHandle);
tAcq = tAcq + toc;

for i=1:numAcq

    % *** Get IQ data ***
    tic
    if i==1
        ch='CHAN1';
    else
        ch='CHAN2';
    end

%     fprintf(instrHandle, [ch ':IQ:DATA:HIST 0,' num2str(-nsamples)]);
    fprintf(instrHandle, [ch ':IQ:DATA:VALues?']);

% check the return beginning with a hash '#'
sStartIndicator = fread(instrHandle,1,'char');
if sStartIndicator ~= '#'; fprintf('oops!\n'); end;

% check the length of the length field in units
nLengthOfLengthfield = fread(instrHandle,1,'char');
nLengthOfLengthfield = str2double(char(nLengthOfLengthfield));

% check the length of the data record
nBlockLength = fread(instrHandle, nLengthOfLengthfield , 'char');

```

```

nBlockLength = str2double(char(nBlockLength)) / nSizeType;

% to make this work, the endianness endian must be considered!
% the RTO supports little endian byte order
iqData2 = fread(instrHandle, nBlockLength, sBinaryFormatString);
iqData2 = iqData2(1:2:end) + 1i*iqData2(2:2:end);

tReadIQ = tReadIQ + toc;
if i==1
    iqData1=iqData2;
end
end

end

function [nErrorNr] = systemErrorQueue(instrHandle)
nErrorNr = -1;
while nErrorNr < 0
    nErrorNr = systemError(instrHandle);
end
disp('Error queue empty')
end

function [nErrorNr] = systemError(instrHandle)
fprintf(instrHandle, 'SYSTEM:ERROR?');
sAux = fscanf(instrHandle);
C = strsplit(sAux, ',');
sErrorNr = C{1};
nErrorNr = str2double(sErrorNr);
disp(sAux)
end

```

APPENDIX 7: CODE TO ESTIMATE ATTENUATION CONSTANT, POWER SIGNAL APPROACH

```
function []=Att_est_LOS()

% Function that estimates the attenuation constant to be used in the power
% attenuation approach, in a LOS environment.
% Created: May 2014, Juan Parras Moral

%% Disconnect and delete all instrument objects -----instrreset
clear all

%% Set up the general parameters needed

Nrep=50; %Number of repetition to estimate the att constant
dist=[1 2 3 4 5 6]; %Distances vector, where measures are taken (in meters)
name=input('Enter name of the file to be saved (use ') ');

% Oscilloscope parameters setup

% Physical channels (input channels)
% First entry: Coupling = 50 Ohm and Trigger on this channel
nChannelNo = [1 2];
nNofActiveCHs = length (nChannelNo); % Number of active channels

% Define number of samples (record length)
nNofSamples = 2e4;

% Set carrier frequency (center frequency)
nCarrierFrequency = 2.395e9;

% Set sample rate
nSampleRate = 250e6;

% Set parameters for IQ mode
sInputType = 'REAL';
sInputMode = 'RFIF';
sSideband = 'NORMal';
nRelBW = 0.4;

%% Connection to the oscilloscope

% Create a VISA connection using HiSLIP protocol to the specified IP
address

IPAddress = '169.254.82.246';
RTO = visa('ni', ['TCP/IP::' IPAddress '::inst0::INSTR']);

% increase the buffer size to, e.g., transport IQ data (in bytes)
RTO.InputBufferSize = 40e6;

% Open the instrument connection
fopen(RTO);

% Clear error queue
systemErrorQueue(RTO);
```

```

% Check the connection
fprintf(RTO, '*IDN?');
disp([' ID: ' fscanf(RTO)])

%Activate View-Mode in Remote Mode
fprintf(RTO, 'SYST:DISP:UPD ON');

%% Oscilloscope parameters setup

% Preset the RTO and wait till action is finished
fprintf(RTO, '*CLS');
fprintf(RTO, '*RST; *OPC?');
fscanf(RTO);

% Activate channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':STATE ON']);
end

% Set Coupling to 50 Ohm
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':COUP DC']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':COUP DC']);

% Set vertical scale
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':SCAL 1e-3']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':SCAL 1e-3']);

% Set horizontal scale and position
fprintf(RTO, 'ACQ:POIN:AUTO RES'); % resolution remains constant
fprintf(RTO, 'ACQ:POIN:MAX 2e6'); % max record length = 2 Msa
fprintf(RTO, 'TIM:SCAL 40e-6'); % horizontal scale = 400 us
fprintf(RTO, 'ACQ:RES 200e-12'); % sample rate = 5GSa/s
fprintf(RTO, 'TIM:REF 10'); % Ref. position = 10%

% Set up trigger (source, type, level and mode)
fprintf(RTO, ['TRIGger:SOURce CHAN' num2str(nChannelNo(1))]);
fprintf(RTO, 'TRIGger:LEV 4.5e-3');
fprintf(RTO, 'TRIGger:LEV:NOISE MAN'); % trigger hysteresis = manual
fprintf(RTO, 'TRIGger:LEV:NOISE:MODE ABS'); % trigger hysteresis mode
fprintf(RTO, 'TRIGger:LEV:NOISE:ABS 10e-6'); % trigger hyst. = 10 uV
fprintf(RTO, 'TRIGger:MODE NORM');
fprintf(RTO, 'TRIGger:TYPE EDGE');

% Check error queue
[~] = systemError(RTO);

% Activate IQ mode
fprintf(RTO, 'IQ:STATe ON');

%Set the input signal, input mode and sideband for active channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPType '
sInputType]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPMode '
sInputMode]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:SBRF '
sSideband]);
end

```

```

    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:CFrequency '
num2str(nCarrierFrequency)]);
end

% Record Length to be used
fprintf(RTO, ['IQ:RLEN ' num2str(nNofSamples)]);
[~] = systemError(RTO);

% Set the correct sampling rate
fprintf(RTO, ['IQ:SRATE ' num2str(nSampleRate)]);
[~] = systemError(RTO);

% Set the relative Bandwidth
fprintf(RTO, ['IQ:RWidth ' num2str(nRelBW)]);
[~] = systemError(RTO);

%% Main loop
power=0;
for i=1:length(dist);
    input(['Press enter once the antenna is in distance d = '
num2str(dist(i)) ' m']);
    for j=1:Nrep;
        display(['Distance= ' num2str(dist(i)) '; Processing iteration '
num2str(j) ' of ' num2str(Nrep)]);
        [power(i,j)]=power_calc(RTO,nNofSamples);
    end
end

%% Attenuation constant estimation processing

pot=10*log10(mean(power)); %Compute the mean power, as a function of
distance
a=polyfit(dist,pot,1) %Fit used to estimate the attenuation constant

%% Plot and save data

plot(dist,pot,'b',dist,a(1)*dist+a(2),'r');title('Data fit
computed');xlabel('Distance (cm)');ylabel('Power received
(dB)');legend('Data obtained','Data fit');
save(name);

%% Close oscilloscope connection

% Close instrument connections
fclose(RTO);
delete(RTO);

end

function[power_ch1]=power_calc(RTO,nNofSamples)
a=1;
th=2e-3;
while a
    [iqDataCh1, iqDataCh2, nTimeAcqIQDataCh1, nTimeReadIQDataCh1] =
ReadRTOIQBin2(RTO, 'Ch1', nNofSamples);
    x1=real(iqDataCh1);
    x1=x1(:);
    power_ch1=x1'*x1;
    if power_ch1>th
        a=0;

```

```

else
    display('Noise detected');
end
end

end
function [ iqData1, iqData2, tAcq, tReadIQ ] = ReadRTOIQBin2( instrHandle,
chNr, nsamples )
%ReadRTOwfmbin Summary of this function goes here
% Detailed explanation goes here

%% Check number of arguments
error(nargchk(1, 3, nargin, 'struct'));    %#ok mlint

% Initialize optional parameters
if ~exist('chNr', 'var'),
    chNr = 'Ch1';
end;

if ~exist('numAcq', 'var'),
    numAcq = 10;
end;

switch lower(chNr)
    case 'ch1'
        chReadCmd = 'CHAN1:IQ:DATA:VALues?';

    case 'ch2'
        chReadCmd = 'CHAN2:IQ:DATA:VALues?';

    case 'ch3wfm1'
        chReadCmd = 'CHAN3:IQ:DATA:VALues?';

    case 'ch4wfm1'
        chReadCmd = 'CHAN4:IQ:DATA:VALues?';

    otherwise
        error('Illegal channel name');

end

%% Select binary format of the transmission [ASC/UINT/REAL]
fprintf(instrHandle, 'FORM REAL,32');
numAcq=2; %Always, to have 2 channels read
fprintf(instrHandle, 'SING;*OPC?');
fscanf(instrHandle);

sBinaryFormatString = 'float';
nSizeType = 4;

tAcq = 0;
tReadIQ = 0;
%     fprintf(instrHandle, 'CHAN1:IQ:HIST:STAT ON');
%     fprintf(instrHandle, 'CHAN2:IQ:HIST:STAT ON');

% *** Initiate acquisition and wait for completion ***
tic

```

```

    fprintf(instrHandle,'SING;*WAI');
    fprintf(instrHandle,'SING;*OPC?');%Starts single acquisition and waits
till it is completed
    fscanf(instrHandle);
    tAcq = tAcq + toc;

    for i=1:numAcq

        % *** Get IQ data ***
        tic
        if i==1
            ch='CHAN1';
        else
            ch='CHAN2';
        end

%         fprintf(instrHandle, [ch ':IQ:DATA:HIST 0,' num2str(-nsamples)]);
        fprintf(instrHandle, [ch ':IQ:DATA:VALues?']);

        % check the return beginning with a hash '#'
        sStartIndicator = fread(instrHandle,1,'char');
        if sStartIndicator ~= '#'; fprintf('oops!\n'); end;

        % check the length of the length field in units
        nLengthOfLengthfield = fread(instrHandle,1,'char');
        nLengthOfLengthfield = str2double(char(nLengthOfLengthfield));

        % check the length of the data record
        nBlockLength = fread(instrHandle, nLengthOfLengthfield , 'char');
        nBlockLength = str2double(char(nBlockLength)) / nSizeType;

        % to make this work, the endianness endian must be considered!
        % the RTO supports litte endian byte order
        iqData2 = fread(instrHandle, nBlockLength, sBinaryFormatString);
        iqData2 = iqData2(1:2:end) + 1i*iqData2(2:2:end);

        tReadIQ = tReadIQ + toc;
        if i==1
            iqData1=iqData2;
        end
    end
end

function [nErrorNr] = systemErrorQueue(instrHandle)
    nErrorNr = -1;
    while nErrorNr < 0
        nErrorNr = systemError(instrHandle);
    end
    disp('Error queue empty')
end

function [nErrorNr] = systemError(instrHandle)
    fprintf(instrHandle,'SYSTem:ERRor?');
    sAux = fscanf(instrHandle);
    C = strsplit(sAux,',');
    sErrorNr = C{1};
    nErrorNr = str2double(sErrorNr);
    disp(sAux)
end

```

APPENDIX 8: CODE OF CROSS-CORRELATION HIGH-IF APPROACH, LOS SETUP

```
function []=Optimized_corr_LOS()
% Function that computes the distance estimation using the High-IF
% approach in a LOS environment.
% Created: May 2014, Juan Parras Moral

%% Disconnect and delete all instrument objects -----
instrreset
clear all

%% Set up the general parameters needed

Nrep=50; %Number of iterations in the main loop
Nsamples=500; %Samples to be used to compute (-1 to use all samples)
fs=250e6; % Sampling frequency
fc=2.412e9; %Carrier frequency (WiFi channel 1)
c1=3e8; %Propagation speed channel 1
c2=3e8; %Propagation speed channel 2
stat=1; %Statistical used to average: 1 is median, 2 is mean
th=2e-3; %Power threshold to discriminate noise
name=input('Enter name of the file to be saved (use '' ) ');

%Oscilloscope parameters setup

% Physical channels (input channels)
% First entry: Coupling = 50 Ohm and Trigger on this channel
nChannelNo = [1 2];
nNofActiveCHs = length (nChannelNo); % Number of active channels

% Define number of samples (record length)
nNofSamples = 2e4;

% Set carrier frequency (center frequency)
nCarrierFrequency=2.395e9;
% Set sample rate
nSampleRate = 250e6;

% Set parameters for IQ mode
sInputType = 'REAL';
sInputMode = 'RFIF';
sSideband = 'NORMal';
nRelBW = 0.4;

%% Connection to the oscilloscope

% Create a VISA connection using HiSLIP protocol to the specified IP
address

IPAddress = '169.254.82.246';
RTO = visa('ni', ['TCPIP::' IPAddress '::inst0::INSTR']);

% increase the buffer size to, e.g., transport IQ data (in bytes)
RTO.InputBufferSize = 40e6;
```

```

% Open the instrument connection
fopen(RTO);

%% Clear error queue
systemErrorQueue(RTO);

% Query and plot the connected instrument ID String
fprintf(RTO, '*IDN?');
disp(['    ID: ' fscanf(RTO)])

%Activate View-Mode in Remote Mode
fprintf(RTO, 'SYST:DISP:UPD ON');

%% Oscilloscope parameters setup

% Preset the RTO and wait till action is finished
fprintf(RTO, '*CLS');
fprintf(RTO, '*RST; *OPC?');
fscanf(RTO);

% Activate channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':STATE ON']);
end

% Set Coupling to 50 Ohm
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':COUP DC']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':COUP DC']);

% Set vertical scale
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':SCAL 20e-3']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':SCAL 20e-3']);

% Set horizontal scale and position
fprintf(RTO, 'ACQ:POIN:AUTO RES'); % resolution remains constant
fprintf(RTO, 'ACQ:POIN:MAX 2e6'); % max record length = 2 MSa
fprintf(RTO, 'TIM:SCAL 40e-6'); % horizontal scale = 400 us
fprintf(RTO, 'ACQ:RES 200e-12'); % sample rate = 5GSa/s
fprintf(RTO, 'TIM:REF 10'); % Ref. position = 10%

% Set up trigger (source, type, level and mode)
fprintf(RTO, ['TRIGger:SOURce CHAN' num2str(nChannelNo(1))]);
fprintf(RTO, 'TRIGger:LEV 3.5e-3');
fprintf(RTO, 'TRIGger:LEV:NOISE MAN'); % trigger hysteresis = manual
fprintf(RTO, 'TRIGger:LEV:NOISE:MODE ABS'); % trigger hysteresis mode
fprintf(RTO, 'TRIGger:LEV:NOISE:ABS 10e-6'); % trigger hyst. = 10 uV
fprintf(RTO, 'TRIGger:MODE NORM');
fprintf(RTO, 'TRIGger:TYPE EDGE');

% Check error queue
[~] = systemError(RTO);

% Activate IQ mode
fprintf(RTO, 'IQ:STATe ON');

%Set the input signal, input mode and sideband for active channels
for nIndex = 1 : nNofActiveCHs

```

```

        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPType '
sInputType]);
        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPMode '
sInputMode]);
        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:SBRF '
sSideband]);
        fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:CFrequency '
num2str(nCarrierFrequency)]);
end

% Record Length to be used
fprintf(RTO, ['IQ:RLEN ' num2str(nNofSamples)]);
[~] = systemError(RTO);

% Set the correct sampling rate
fprintf(RTO, ['IQ:SRATE ' num2str(nSampleRate)]);
[~] = systemError(RTO);

% Set the relative Bandwidth
fprintf(RTO, ['IQ:RBwidth ' num2str(nRelBW)]);
[~] = systemError(RTO);

%% Main loop

dist=0; %Where the distance estimation is going to be stored
for j=1:Nrep;
    display(['Processing iteration ' num2str(j) ' of ' num2str(Nrep)]);
    [samples(j)
dist(j)]=d1_calc(RTO,nNofSamples,Nsamples,fs,c1,c2,nCarrierFrequency,fc,th)
;
end

%% Distance estimation processing

if stat==1
    ddif1=median(dist)
else
    ddif1=mean(dist)
end

%% Plot and save data

plot(1:length(dist),dist,'b');
display(['Mean dd = ' num2str(mean(dist)) ' Median = '
(num2str(median(dist))) ' Std desviation = ' num2str(sqrt(var(dist)))]);
save(name, 'dist','d','fc','samples','ddif1');

%% Close oscilloscope connection

% Close intrument connections
fclose(RTO);
delete(RTO);

end

function[samples_shift,dist_dif]=d1_calc(RTO,nNofSamples,Nsamples,fs,c1,c2,
fol,fc,th)

% Capture data, avoiding noise

```

```

a=1;
while a
    [iqDataCh1, iqDataCh2, nTimeAcqIQDataCh1, nTimeReadIQDataCh1] =
ReadRTOIQBin2(RTO, 'Ch1', nNofSamples);
    x1=real(iqDataCh1);
    x2=real(iqDataCh2);
    x1=x1(:);
    figure(1);
    plot(x1);
    if x1'*x1>th
        a=0;
    else
        display('Noise detected');
    end
end

% Select the samples to be used

if Nsamples~-1 %If Nsamples== -1, take all samples
    x1=x1(nNofSamples/2:nNofSamples/2+Nsamples-1);
    x2=x2(nNofSamples/2:nNofSamples/2+Nsamples-1);
end

% Compute the distance estimation using cross correlation

y=xcorr(x1,x2);
[val,ind]=max(y);
samples_shift=ind-length(x1)
if c1~=c2; display('Warning: d1 measure wrong (c1~=c2)'); end;
dist_dif=samples_shift*c1*abs(fc-fol)/(fs*fc)

end

function [ iqData1, iqData2, tAcq, tReadIQ ] = ReadRTOIQBin2( instrHandle,
chNr, nsamples )
%ReadRTOwfmbin Summary of this function goes here
% Detailed explanation goes here

%% Check number of arguments
error(nargchk(1, 3, nargin, 'struct'));    %#ok mlint

% Initialize optional parameters
if ~exist('chNr', 'var'),
    chNr = 'Ch1';
end;

if ~exist('numAcq', 'var'),
    numAcq = 10;
end;

switch lower(chNr)
    case 'ch1'
        chReadCmd = 'CHAN1:IQ:DATA:VALues?';

    case 'ch2'
        chReadCmd = 'CHAN2:IQ:DATA:VALues?';

    case 'ch3wfml'

```

```

        chReadCmd = 'CHAN3:IQ:DATA:VALues?';

    case 'ch4wfml'
        chReadCmd = 'CHAN4:IQ:DATA:VALues?';

    otherwise
        error('Illegal channel name');

end

%% Select binary format of the transmission [ASC/UINT/REAL]
fprintf(instrHandle, 'FORM REAL,32');
numAcq=2; %Always, to have 2 channels read
fprintf(instrHandle, 'SING;*OPC?');
fscanf(instrHandle);

sBinaryFormatString = 'float';
nSizeType = 4;

tAcq = 0;
tReadIQ = 0;
%     fprintf(instrHandle, 'CHAN1:IQ:HIST:STAT ON');
%     fprintf(instrHandle, 'CHAN2:IQ:HIST:STAT ON');

% *** Initiate acquisition and wait for completion ***
tic
    %fprintf(instrHandle, 'SING;*WAI');
    fprintf(instrHandle, 'SING;*OPC?'); %Starts single acquisition and waits
till it is completed
fscanf(instrHandle);
tAcq = tAcq + toc;

for i=1:numAcq

    % *** Get IQ data ***
    tic
    if i==1
        ch='CHAN1';
    else
        ch='CHAN2';
    end

%     fprintf(instrHandle, [ch ':IQ:DATA:HIST 0,' num2str(-nsamples)]);
    fprintf(instrHandle, [ch ':IQ:DATA:VALues?']);

    % check the return beginning with a hash '#'
    sStartIndicator = fread(instrHandle,1,'char');
    if sStartIndicator ~= '#'; fprintf('oops!\n'); end;

    % check the length of the length field in units
    nLengthOfLengthfield = fread(instrHandle,1,'char');
    nLengthOfLengthfield = str2double(char(nLengthOfLengthfield));

    % check the length of the data record
    nBlockLength = fread(instrHandle, nLengthOfLengthfield , 'char');
    nBlockLength = str2double(char(nBlockLength)) / nSizeType;

    % to make this work, the endianness endian must be considered!

```

```

    % the RTO supports little endian byte order
    iqData2 = fread(instrHandle, nBlockLength, sBinaryFormatString);
    iqData2 = iqData2(1:2:end) + 1i*iqData2(2:2:end);

    tReadIQ = tReadIQ + toc;
    if i==1
        iqData1=iqData2;
    end
end

end

function [nErrorNr] = systemErrorQueue(instrHandle)
    nErrorNr = -1;
    while nErrorNr < 0
        nErrorNr = systemError(instrHandle);
    end
    disp('Error queue empty')
end

function [nErrorNr] = systemError(instrHandle)
    fprintf(instrHandle, 'SYSTEM:ERROR?');
    sAux = fscanf(instrHandle);
    C = strsplit(sAux, ',');
    sErrorNr = C{1};
    nErrorNr = str2double(sErrorNr);
    disp(sAux)
end

```

APPENDIX 9: CODE OF CROSS-CORRELATION NEAR-ZERO-IF APPROACH, LOS SETUP

```
function []=Optimized_corr_LOS()

% Function that computes the distance estimation using the Near-Zero-IF
% approach in a LOS environment.
% Created: May 2014, Juan Parras Moral

%% Disconnect and delete all instrument objects
instrreset
clear all

%% Set up the general parameters needed

Nrep=50;%Number of iterations in the main loop
Nsamples=250; %Samples to be used to compute (reduce load, increase error)
fs=input('Enter fs ');% Sampling frequency
fc=2.412e9;%Carrier frequency (WiFi channel 1)
c1=3e8; %Propagation speed channel 1
c2=3e8; %Propagation speed channel 2
stat=1; %Statistical used to average: 1 is median, 2 is mean
f_offset=100e3;%Frequency offset used
df_of=0;%Distance offset used in fine approach
d=6; %Total distance between antennas
th=2e-3; %Power threshold to discriminate noise
n=8; %Samples of the FFT to compute the offset
dt=input('Enter d for title ');
name=['fine_' num2str(dt) '_' num2str(fs/1e6) 'k']; %Name to save data

%Oscilloscope parameters setup

% First entry: Coupling = 50 Ohm and Trigger on this channel
nChannelNo = [1 2];
nNofActiveCHs = length (nChannelNo); % Number of active channels

% Define number of samples (record length)
nNofSamples = 2e4;

% Set carrier frequency (center frequency)
nCarrierFrequency=fc;
% Set sample rate
nSampleRate = fs;

% Set parameters for IQ mode
sInputType = 'REAL';
sInputMode = 'RFIF';
sSideband = 'NORMal';
nRelBW = 0.4;

%% Connection to the oscilloscope

% Create a VISA connection using HiSLIP protocol to the specified IP
address

IPAddress = '169.254.82.246';
RTO = visa('ni', ['TCPIP::' IPAddress '::inst0::INSTR']);
```

```

% increase the buffer size to, e.g., transport IQ data (in bytes)
RTO.InputBufferSize = 40e6;

% Open the instrument connection
fopen(RTO);

%Clear error queue
systemErrorQueue(RTO);

% Check the connection
fprintf(RTO, '*IDN?');
disp([' ID: ' fscanf(RTO)])

%Activate View-Mode in Remote Mode
fprintf(RTO,'SYST:DISP:UPD ON');

%% Oscilloscope parameters setup

% Preset the RTO and wait till action is finished
fprintf(RTO,'*CLS');
fprintf(RTO,'*RST; *OPC?');
fscanf(RTO);

% Activate channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':STATE ON']);
end

% Set Coupling to 50 Ohm
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':COUP DC']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':COUP DC']);

% Set vertical scale
fprintf(RTO, ['CHAN' num2str(nChannelNo(1)) ':SCAL 1e-3']);
fprintf(RTO, ['CHAN' num2str(nChannelNo(2)) ':SCAL 1e-3']);

% Set horizontal scale and position
fprintf(RTO, 'ACQ:POIN:AUTO RES'); % resolution remains constant
fprintf(RTO, 'ACQ:POIN:MAX 2e6'); % max record length = 2 Msa
fprintf(RTO, 'TIM:SCAL 40e-6'); % horizontal scale = 400 us
fprintf(RTO, 'ACQ:RES 200e-12'); % sample rate = 5GSa/s
fprintf(RTO, 'TIM:REF 10'); % Ref. position = 10%

% Set up trigger (source, type, level and mode)
fprintf(RTO, ['TRIGger:SOURce CHAN' num2str(nChannelNo(1))]);
fprintf(RTO, 'TRIGger:LEV 4.5e-3');
fprintf(RTO, 'TRIGger:LEV:NOISE MAN'); % trigger hysteresis = manual
fprintf(RTO, 'TRIGger:LEV:NOISE:MODE ABS'); % trigger hysteresis mode
fprintf(RTO, 'TRIGger:LEV:NOISE:ABS 10e-6'); % trigger hyst. = 10 uV
fprintf(RTO, 'TRIGger:MODE NORM');
fprintf(RTO, 'TRIGger:TYPE EDGE');

% Check error queue
[~] = systemError(RTO);

% Activate IQ mode

```

```

fprintf(RTO, 'IQ:STATE ON');

%Set the input signal, input mode and sideband for active channels
for nIndex = 1 : nNofActiveCHs
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPType '
sInputType]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:INPMode '
sInputMode]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:SBRF '
sSideband]);
    fprintf(RTO, ['CHAN' num2str(nChannelNo(nIndex)) ':IQ:CFRequency '
num2str(nCarrierFrequency-f_offset)]);
end

% Record Length to be used
fprintf(RTO, ['IQ:RLEN ' num2str(nNofSamples)]);
[~] = systemError(RTO);

% Set the correct sampling rate
fprintf(RTO, ['IQ:SRate ' num2str(nSampleRate)]);
[~] = systemError(RTO);

% Set the relative Bandwidth
fprintf(RTO, ['IQ:RBwidth ' num2str(nRelBW)]);
[~] = systemError(RTO);

%% Main loop

for j=1:Nrep;
    display(['Processing iteration ' num2str(j) ' of ' num2str(Nrep)]);

[d_f(j),d_r(j),ns_fine(j),ns_brut(j)]=d1_calc(RTO,nNofSamples,Nsamples,fs,c
1,c2,fc,th,n,f_offset,df_of);
end

%% Distance estimation processing

if stat==1
    df=median(d_f);
    dr=median(d_r);
else
    df=mean(d_f);
    dr=mean(d_r);
end
df=mod(df*100,12.5)
dr=0.5*(dr+d);

%% Plot and save data

figure(1);
subplot(211);plot(d_f);title('Distance fine');
subplot(212);plot(d_r);title('Distance rough');
save(name, 'd_f', 'd_r', 'ns_fine', 'ns_brut', 'fc', 'Nsamples', 'fs');

%% Close oscilloscope connection

% Close intrument connections
fclose(RTO);
delete(RTO);

```

```

end

function[d_fine,d_rough,ns_fine,ns_brut]=d1_calc(RTO,nNofSamples,Nsamples,f
s,c1,c2,fc,th,n,f_offset,df_of)

if c1~=c2; display('Warning: d1 measure wrong (c1~=c2)'); end;

% Capture data, using the noise threshold to avoid noise

a=1;
while a
    [iqDataCh1, iqDataCh2, nTimeAcqIQDataCh1, nTimeReadIQDataCh1] =
ReadRTOIQBin2(RTO, 'Ch1', nNofSamples);
    x1=real(iqDataCh1);
    x2=real(iqDataCh2);
    x1=x1(:);
    figure(2);
    subplot(211);plot(1:length(x1),x1,'b',1:length(x2),x2,'r');
    if x1'*x1>th
        a=0;
    else
        display('Noise detected');
    end
end

%Signal normalization
x1=x1/max(x1);
x2=x2/max(x2);

%Fine approach calculus

% Decimation of the signals
N=10;
xa=resample(x1,1,N);
Xa=fft(xa);
Xa(n:end-n)=0;
sa=real(ifft(Xa));

xb=resample(x2,1,N);
Xb=fft(xb);
Xb(n:end-n)=0;
sb=real(ifft(Xb));

subplot(212);plot(1:length(sa),sa/max(sa),'b',1:length(sa),sb/max(sb),'r');

% Correlation and distance estimation

y=xcorr(sb,sa);
[val ind]=max(y);
ns_fine=ind-length(xa)
d_fine=c1*f_offset*ns_fine/(fc*fs/N)-df_of;

%Rough approach

%Since the signals are normalized, the first sample will be that which a
%20% of diynamic range - That way initial noise is avoided!

% Selection of the samples and computation of distance estimation

```

```

aux=find(x1>0.2);

if Nsamples==-1 %Use all samples
    y=abs(xcorr(x1(aux(1):end),x2(aux(1):end)));
    ncorr=length(x1(aux(1):end));
else
    y=abs(xcorr(x1(aux(1):aux(1)+Nsamples),x2(aux(1):aux(1)+Nsamples)));
    ncorr=Nsamples;
end

[val, ind]=max(y);

ns_brut=ind-ncorr
d_rough=ns_brut*c1/fs;

end

function [ iqData1, iqData2, tAcq, tReadIQ ] = ReadRTOIQBin2( instrHandle,
chNr, nsamples)
%ReadRTOWfmBin Summary of this function goes here
% Detailed explanation goes here

%% Check number of arguments
error(nargchk(1, 3, nargin, 'struct'));    %#ok mlint

% Initialize optional parameters
if ~exist('chNr', 'var'),
    chNr = 'Ch1';
end;

if ~exist('numAcq', 'var'),
    numAcq = 10;
end;

switch lower(chNr)
    case 'ch1'
        chReadCmd = 'CHAN1:IQ:DATA:VALues?';

    case 'ch2'
        chReadCmd = 'CHAN2:IQ:DATA:VALues?';

    case 'ch3wfm1'
        chReadCmd = 'CHAN3:IQ:DATA:VALues?';

    case 'ch4wfm1'
        chReadCmd = 'CHAN4:IQ:DATA:VALues?';

    otherwise
        error('Illegal channel name');

end

%% Select binary format of the transmission [ASC/UINT/REAL]
fprintf(instrHandle,'FORM REAL,32');
numAcq=2; %Always, to have 2 channels read

```

```

fprintf(instrHandle, 'SING;*OPC?');
fscanf(instrHandle);

sBinaryFormatString = 'float';
nSizeType = 4;

tAcq = 0;
tReadIQ = 0;
%     fprintf(instrHandle, 'CHAN1:IQ:HIST:STAT ON');
%     fprintf(instrHandle, 'CHAN2:IQ:HIST:STAT ON');

% *** Initiate acquisition and wait for completion ***
tic
%fprintf(instrHandle, 'SING;*WAI');
fprintf(instrHandle, 'SING;*OPC?'); %Starts single acquisition and waits
till it is completed
fscanf(instrHandle);
tAcq = tAcq + toc;

for i=1:numAcq

    % *** Get IQ data ***
    tic
    if i==1
        ch='CHAN1';
    else
        ch='CHAN2';
    end

    %     fprintf(instrHandle, [ch ':IQ:DATA:HIST 0,' num2str(-nsamples)]);
    fprintf(instrHandle, [ch ':IQ:DATA:VALUES?']);

    % check the return beginning with a hash '#'
    sStartIndicator = fread(instrHandle,1,'char');
    if sStartIndicator ~= '#'; fprintf('oops!\n'); end;

    % check the length of the length field in units
    nLengthOfLengthfield = fread(instrHandle,1,'char');
    nLengthOfLengthfield = str2double(char(nLengthOfLengthfield));

    % check the length of the data record
    nBlockLength = fread(instrHandle, nLengthOfLengthfield , 'char');
    nBlockLength = str2double(char(nBlockLength)) / nSizeType;

    % to make this work, the endianness endian must be considered!
    % the RTO supports little endian byte order
    iqData2 = fread(instrHandle, nBlockLength, sBinaryFormatString);
    iqData2 = iqData2(1:2:end) + 1i*iqData2(2:2:end);

    tReadIQ = tReadIQ + toc;
    if i==1
        iqData1=iqData2;
    end
end

end

function [nErrorNr] = systemErrorQueue(instrHandle)
nErrorNr = -1;

```

```
while nErrorNr < 0
    nErrorNr = systemError(instrHandle);
end
disp('Error queue empty')
end

function [nErrorNr] = systemError(instrHandle)
    fprintf(instrHandle, 'SYSTEM:ERROR?');
    sAux = fscanf(instrHandle);
    C = strsplit(sAux, ',');
    sErrorNr = C{1};
    nErrorNr = str2double(sErrorNr);
    disp(sAux)
end
```