UNIVERSITÉ CATHOLIQUE DE LOUVAIN

ECOLE POLYTECHNIQUE DE LOUVAIN

TELE-LAB

Code design for TDOA UWB based positioning of multiple sources

---

Supervisor
Luc Vandendorpe

Adrià Gusi i Amigó

August 25, 2010

Acknowledgments

First, I would like to thank my supervisor, Luc Vandendorpe, for giving me the opportunity and all the support to realize this master thesis.

My gratitude also goes to Achraf Mallat for his support and guidance all through the duration of this thesis work.

I would also like to acknowledge the encouragement provided by my friends from Barcelona and Louvain-La-Neuve and specially by my girlfriend Cristina.

Finally, I dedicate this thesis to my mother and my grandfather for their unconditional support.

Abstract

In this thesis we study the feasibility of obtaining acceptable variance values in TDOA UWB based positioning of multiple sources. We define the conditions a time-hopping code must satisfy to mitigate multi-user interference. We design two time-hopping code generators. We propose a TOA estimator. We show simulation results and we conclude that both generated time-hopping code and the proposed estimator reduce the TOA estimation variance and that it is possible to reach the CRB when multiple sources are transmitting. Finally, we prove that TDOA based positioning can achieve a standard deviation of the position error in the order of mm.

Contents

1 Introduction **1**
- 1.1 State of the art . . . 1
  - 1.1.1 Introduction . . . 1
  - 1.1.2 Ultra-Wideband technology . . . 1
    - 1.1.2.1 Shannon's theory and UWB . . . 1
    - 1.1.2.2 History . . . 2
    - 1.1.2.3 Regulation . . . 3
    - 1.1.2.4 Characteristics . . . 3
  - 1.1.3 Localization with UWB . . . 7
    - 1.1.3.1 Localization techniques . . . 7
    - 1.1.3.2 Position derivation with TDOA . . . 11
    - 1.1.3.3 Proved Applications . . . 12
  - 1.1.4 Conclusion . . . 13
- 1.2 Motivation . . . 13
- 1.3 Outline . . . 13

2 Impulse-Radio Ultra-Wideband signal **15**
- 2.1 Introduction . . . 15
- 2.2 Time-Hopping Pulse Position Modulation . . . 15
  - 2.2.1 Pseudo-random time-hopping . . . 16
  - 2.2.2 Data modulation . . . 17
- 2.3 Received signal . . . 18
  - 2.3.1 Estimator . . . 19
- 2.4 Propagation model . . . 20
- 2.5 Conclusion . . . 21

3 Optimum Time-Hopping code **23**
- 3.1 Introduction . . . 23
- 3.2 Time-hopping code notation . . . 23
- 3.3 Correlation output . . . 24
- 3.4 Conditions for minimum correlation THC . . . 26
  - 3.4.1 Maximum correlation matrix . . . 27
- 3.5 Conclusion . . . 28

4 Time-hopping code generators **29**
4.1 Introduction . . . 29
4.2 Backtracking generator . . . 29
4.2.1 Procedure . . . 29
4.2.2 Example of a backtracking generated code . . . 33
4.3 Fast generator . . . 36
4.3.1 Procedure . . . 36
4.4 Conclusion . . . 37

5 Proposed estimator **38**
5.1 Introduction . . . 38
5.2 Basic estimator . . . 38
5.3 Maximum cancellation estimator . . . 38
5.3.1 Procedure . . . 39
5.4 Iterative cancellation estimator . . . 40
5.4.1 Procedure . . . 40
5.5 Conclusion . . . 42

6 Simulation results **43**
6.1 Introduction . . . 43
6.2 Method . . . 43
6.2.1 Introduction . . . 43
6.2.2 Error calculation . . . 44
6.2.3 TOA Variance calculation . . . 45
6.2.4 TDOA positioning variance . . . 45
6.3 Results presentation . . . 46
6.4 Results . . . 46
6.4.1 One user . . . 46
6.4.2 Multiple users . . . 47
6.4.3 TDOA simulation results . . . 52
6.5 Conclusion . . . 56

7 Conclusions and future work **57**
7.1 Conclusions . . . 57
7.2 Future work . . . 58

8 Annexe **I**

List of Figures

1.1 Bandwidth usage of different wireless technologies 2
1.2 FCC spectral mask for indoor applications 4
1.3 ECC spectral mask 4
1.4 Second derivative of a Gaussian 6
1.5 Normalized spectrum of a UWB pulse 6
1.6 CRB versus SNR for different pulse widths. 10
1.7 TDOA positioning with two hyperbolas 11
1.8 TDOA positioning with two hyperbolas 12

2.1 Transmitted signal from one user. 18
2.2 (a) Transmitted signals (b) Received signal without noise (c) Received signal with noise 19
2.3 Correlation between received signal and one user template 21
2.4 SNR vs distance 22

4.1 Example of a tree structure 33
4.2 Tree structure for $N_h = 3$ and $N_c = 3$ 36

6.1 Variance for one user 47
6.2 Variance of $N_u = 4$, $N_h = 6$, $N_c = 3$ 48
6.3 Variance of $N_u = 4$, $N_h = 8$, $N_c = 4$ vs $N_u = 4$, $N_h = 3$, $N_c = 3$ 49
6.4 Variance of $N_u = 4$, $N_h = 15$, $N_c = 5$ 50
6.5 Variance for $N_c = 3$, $N_c = 4$ and $N_c = 5$ 51
6.6 Variance of 1 user and 4 users, Nc=5 52
6.7 Position estimation for SNR=15 53
6.8 Position estimation for SNR=15 54
6.9 Position estimation for (a) SNR=20 (b) SNR=25 (c) SNR=30 . . 54
6.10 Variance of the TOA estimation and TDOA estimated position . 55

Nomenclature

AOA Angle Of Arrival

AWGN Additive White Gaussian Noise

ECC Electronic Communications Committee

FCC Federal Communications Commission

IR-UWB Impulse-Radio UWB

ISM Industrial, Scientic and Medical

LOS Line Of Sight

MAI Multi-Access Interference

MPC Multipath Component

MSE Mean Square Error

OOK On-Off Keying

PAM Pulse Amplitude Modulation

PPM Pulse Position Modulation

PSD Power Spectral Density

RSS Received Signal Strength

SNR Signal to Noise Ratio

TDOA Time Difference Of Arrival

TH Time-Hopping

THC Time-Hopping Code

TOA Time Of Arrival

UWB Ultra-Wideband

Notation

Constants

π	Pi = 3.14159
c	Speed of light
∞	Infinity

Functions

e^x	Exponential Function
$\frac{d^n y}{dx^n}$	n^{th}derivative of $y = f(x)$
$\arg\max_x f(x)$	Argument of the maximum

Statistical parameters

$\mathrm{E}\{x\}$	Expected value of x
$\mathrm{Var}\{x\}$	Variance of x
$\sigma\{x\}$	Standard deviation of x

Matrices and vectors

$\mathbf{v}$	Column vector
$\mathbf{v^T}$	Row vector
$\mathbf{M}$	Matrix

Chapter 1

Introduction

1.1 State of the art

1.1.1 Introduction

Ultra-Wideband (UWB) is an emerging wireless technology that has proved excellent performance in precision localization [8]. In this chapter we will see an introduction to Ultra-Wideband technology and we will discuss the characteristics that justify its use for localization applications.

1.1.2 Ultra-Wideband technology

UWB technology is defined by the US Frequency Regulator (Federal Communications Commission, FCC) as any wireless scheme that occupies a fractional bandwidth larger than 20 % or more than 500 MHz of absolute bandwidth [10]. So, an UWB signal is characterized by its very large bandwidth compared to other narrowband systems, as we can see in figure 1.1 [12].

The absolute bandwidth is calculated as the difference between the upper frequency f_H of the -10 dB emission point and the lower frequency f_L of the -10 dB emission point . This bandwidth is also defined -10 dB bandwith

$$B = f_H - f_L$$

On the other hand, fractional bandwidth is defined as

$$B_{frac} = \frac{2(f_H - f_L)}{f_H + f_L}$$

1.1.2.1 Shannon's theory and UWB

We can understand the benefits of using a very large bandwidth by examining Shannon's capacity equation for an Additive White Gaussian Noise (AWGN)

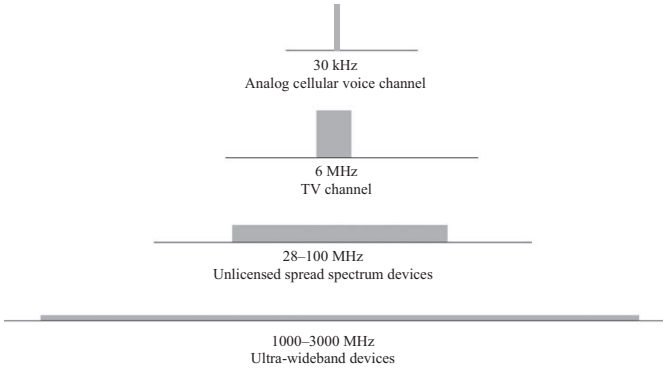


Figure 1.1: Bandwidth usage of different wireless technologies

channel [3]

$$C = B \log(1 + \frac{S}{N})$$

where C is the maximum channel capacity in [bits/second], B is the channel bandwith in hertz [Hz] and S/N is the signal to noise ratio (SNR), where S is the signal power in watts [W] and N is the noise power in watts. We can see that the maximum capacity of the channel grows linearly when bandwith B increases but grows logarithmically with signal to noise ratio SNR. Given this, the expansion of operating bandwidth allows very low signal to noise ratio systems with the same capacity. Consequently, UWB devices can have a much lower operating power than conventional narrowband communication systems.

1.1.2.2 History

Although, most people consider UWB as a recent breakthrough technology in wireless communications, UWB is more precisely a new engineering technology in that no new physical properties have been discovered. In 1890s Marconi and Hertz designed a spark-gap transmission which consisted in a spark discharge to produce electromagnetic waves. That was the first wireless communication system and, in fact, was based on UWB [10].

However, owing to the technical limitations, narrowband communication was preferred to UWB, and consequently the dominant form of wireless communications became sinusoidal. UWB was destined for years to systems designed for military covert radar and communication. It was not until 1960s that interest was focused again on UWB, when Dr. Gerald F. Ross proved the feasibility of

UWB systems in communications and radar. Years after, in 1990s, pioneer work on impulse radio UWB (IR-UWB) from Win and Scholtz made UWB again a point of attraction. UWB received the definitive impulse in 2002 when FCC decided to allow operations without license for UWB in United States. Until that moment, its emissions were limited to the ISM band (Industrial, Scientific and Medical) [11].

1.1.2.3 Regulation

The reason of regulation is to assure that the power spectral density (PSD) of an UWB satisfies the spectral masks specified by frequency regulating agencies. The power spectral density is defined as

$$PSD = \frac{P}{B}$$

where P is the transmitted power in watts [W] and B the bandwidth used by the signal in hertz [Hz], so PSD unit is watts/hertz [W/Hz].

In order to benefit from advantages of UWB without degrading the performance of other systems, the FCC in the United States specified an spectral mask for indoor applications which allowed UWB operations. According to this spectral mask, UWB systems must transmit below certain power levels so they don't cause significant interference to the other systems in the same frequency spectrum. The spectral mask allows UWB transmissions on an unlicensed band from 3.1 to 10.6 GHz with a PSD lower than to -41.3 dBm/MHz. It also allows UWB communications in other bands but with an even lower limit[8]. So, the recommended band is the 7.5 GHz from 3.1 to 10.6 GHz, as we can see in figure 1.2.

On account of European regulation, it was not until 2006 that Electronic Communications Committee (ECC) from European Conference of Communications allowed UWB operations without license in Europe. The allowed band for a power spectral density of -41.3 dBm/MHz is only a portion from the one in United States and it belongs to the band from 6 to 8.5 GHz. UWB systems can also transmit -41.3 dBm/MHz in the 4.2–4.8 GHz band until the end of 2010. After then, a limit of -70 dBm/MHz will be imposed on that band. So, it should be noted that the ECC Decision intends to deliver a clear message that the 6–8.5 GHz band is identified in Europe for long-term UWB operation[8]. ECC emission limits can be seen in figure 1.3.

1.1.2.4 Characteristics

UWB communications come in single-band or either multiband. In this thesis we focus on impulse radio UWB, which is a single-band UWB system. Impulse radio UWB systems are characterized by the transmission of sequences of waveforms with very short duration and low power in a very fine time precision,

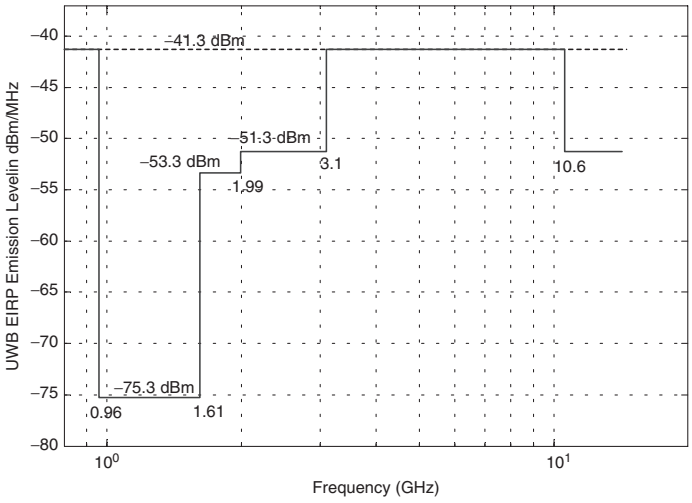


Figure 1.2: FCC spectral mask for indoor applications

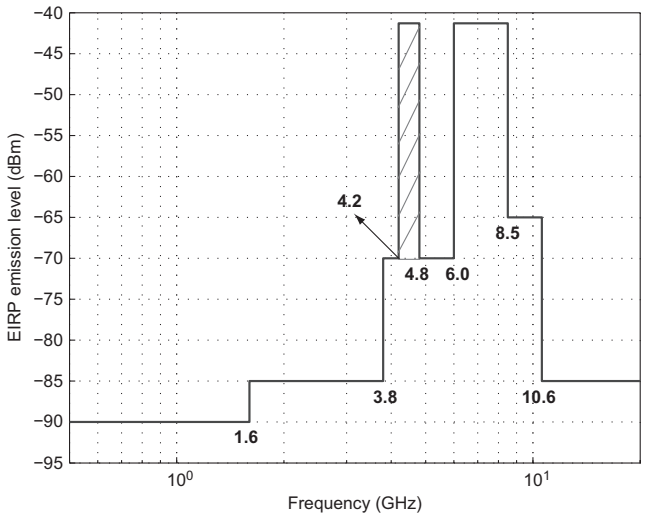


Figure 1.3: ECC spectral mask

usually on the order of a nanosecond. In other words, the emitter converts data and transmits it by sending thousands of pulses through a wide spectrum of a few GHz, then the receiver detects the transmitted pulses and tries to match a certain sequence of pulses. Also, note that since the bandwidth is very large, better resolution of multipath is achieved and signal power can be kept low to increase the battery life of the system and to minimize the interference to the other systems in the same frequency spectrum. Moreover, a UWB system can be operated in baseband, meaning that UWB pulses can be transmitted without a sine-wave carrier ("carrier-free"). In that case, the system does not require IF processing, which facilitates low cost implementations.

Pulse shape We can choose between different pulse shapes in UWB, but the most popular one is the Gaussian pulse and its derivatives owing to its simplicity and the fact that satisfies the spectral masks mentioned above. Its shape is easily generated since it is just a square pulse that has been shaped by the limited rise and fall times of the pulse and the filtering effects of the transmit and receive antennas. Moreover, a square pulse can be generated easily by switching on and off a transistor.

So, we can use Gaussian pulses from zero to fifth order. All of them are assumed to zero-mean and they can also be called Gaussian pulse, monocycle or doublet. The zero order Gaussian pulse is defined in [2] as follows

$$w_0(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}} \tag{1.1}$$

where σ is the variance and it determines the pulse width T_w. In order the obtain different order waveforms, equation 1.1 must be derived to the desired order

$$w_n(t) = \frac{d^n}{dt^n} w_0(t)$$

All the simulations of this thesis have been carried out using the second derivative of a Gaussian. Its shape can be observed in figure 1.4. The pulse width T_w is equal to $\sqrt{2\pi}\sigma \simeq 2.5\sigma$.

$$w_2(t) = \left[1 - 4\pi \left(\frac{t}{\sigma}\right)^2\right] e^{-2\pi\left(\frac{t}{\sigma}\right)^2}$$

In figure 1.5 we can observe the spectrum of a Gaussian pulse. The center frequency is approximately 5 GHz, with the 3 dB bandwidth extending over several GHz. In comparison with narrowband or even wideband communication systems the large bandwidth is very noticeable.

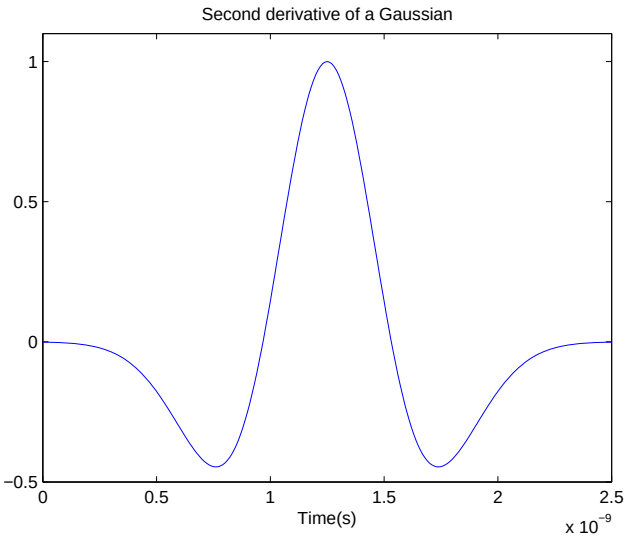


Figure 1.4: Second derivative of a Gaussian

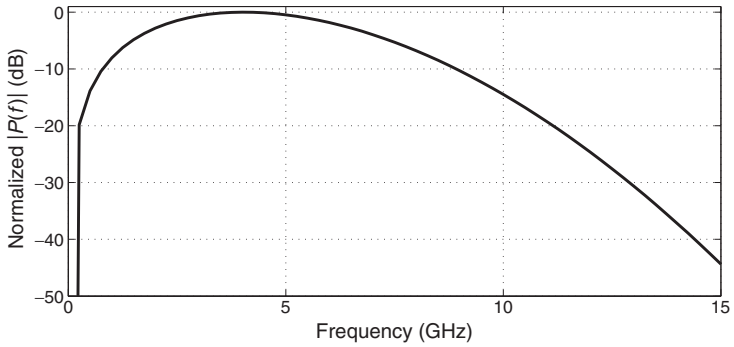


Figure 1.5: Normalized spectrum of a UWB pulse

Summary As a summary, the most important characteristics of UWB technology are:

- High time resolution due to its short duration pulses
- Penetration through obstacles given that energy is spread in different frequencies.
- High ranging, hence positioning, accuracy
- Robust against multipath propagation owing to its high temporal resolution
- Multiple access possibility given its wide bandwidth
- High data transmission rate
- Safe transmissions since UWB signal to noise ratio is so low that transmissions are easily seen as noise
- Low cost and low power implementation.
- Low rate of interference

1.1.3 Localization with UWB

As we have seen, UWB signals have an excellent time resolution due to its short duration pules. Exploiting this resolution, precision localization becomes one of the application areas for impulse radio ultra-wideband (IR-UWB) technology. However, UWB is not a localization technology that satisfies all applications. Theoretically, UWB range could cover distances in the order of hundreds of meters, but practically , the standard limits its radius of coverage to tens of meters in order to avoid possible interferences. This limitation reduces the number of applications that this wireless technology may have. Consequently, UWB localization is only suitable for short and medium range applications like reduced space scenarios, and more specifically, indoor applications.

Even this limitation seems to be a negative point, it can turn to a benefit since it clarifies the type of applications and lets focus on them. So, whereas GPS is an excellent choice for many scenarios and typically has fine precision outdoors, it does not obtain good accuracies in indoor environments due to multipath effects and blocked line of sight (LOS). Moreover, GPS devices are usually too expensive. On the other hand, UWB is an excellent signaling choice for high accuracy localization in short to medium distances due to its high time resolution and inexpensive circuitry. Some varieties of UWB localization techniques can be used to determine the range between UWB radios indoors.

1.1.3.1 Localization techniques

The parameters that have been most commonly used for positioning are the received signal strength (RSS), the angle of arrival (AOA), time of arrival (TOA) and a variant of this last, time difference of arrival (TDOA) [8].

RSS With this technique, the signal strength of the transmitting source is measured at several static receivers. The main idea is that if the relation between distance and power loss is known, the received signal strength measured at a node can be used to estimate the distance between that node and the transmitting node, assuming that the transmit power is known. Ideally, each measurement will generate a circle centered at the corresponding receiver with a radius that represents the distance between the object and that receiver. By intersecting several circles the object position can be obtained. At least three receivers are required in two-dimensional positioning in order to resolve the intersection ambiguities. Moreover, this technique requires a site-specific path loss to obtain an accurate estimation of the RSS. Given that this approach does not exploit the fine time resolution of UWB short duration pulses this is the least adequate parameter to localize in the case of UWB.

AOA This estimation measures the angle of arrival of the signal sent by the object to be positioned at several receivers. The receivers obtain the estimation by steering the main lobe of a directional antenna or an adaptive antenna array. Each measurement forms a radial line from the receiver to the object to be positioned. In two-dimensional positioning the position of the object is defined at the intersection of two directional lines of bearing. In practice, more than two receivers may be employed to combat inaccuracies introduced by multipath propagation effects. Moreover, this method has the advantage of not requiring synchronization of the receivers nor an accurate timing reference. However, this estimation requires multiple antennas or a beamforming antenna and their regular calibration due to temperature variations and antenna mismatches. This requirement implies relative high size and complexity receivers, necessity that UWB is not specially suited for given its low power, low cost and small size constraints. Also, in [5] they conclude that the CRB from TOA always outperforms the one from AOA.

TOA With this approach the time of flight between two nodes is estimated. Since the propagation time of the signal is known, the measured time can provide a radius of a circle centered at the receiver. In two-dimensional localization, we need at least three circles to solve the intersection, which means three receivers. Moreover, synchronization between transmitter and each receiver is mandatory, either by having common clocks or changing timing information via certain protocols. TOA estimation is the most attractive parameter for positioning with UWB since it exploits the fine time resolution.

Typically, TOA estimation technique is done by matched filtering or correlation operations [8]. The received signal at a node can be expressed as

$$r(t) = \alpha s(t-\tau) + n(t) \tag{1.2}$$

where τ represents the TOA, α is the channel coefficient, and $n(t)$ is AWGN with zero mean and a spectral density of $N_0/2$. Then, a correlator based receiver searches for the peak of the correlation of $r(t)$ with a reference signal or template

signal from $s(t)$, defined as $v(t)$, for different delays $\hat{\tau}$. So, let the estimated time of arrival, $\hat{\tau}_{TOA}$, be

$$\hat{\tau}_{TOA} = \arg\max_{\tau} \{r(\tau) * v(-\tau)\} = \arg\max_{\tau} \int_{t_i} r(t)v(t+\tau)dt \tag{1.3}$$

where t_i is the observation interval. This kind of receiver is optimal for single-path AWGN channels.

The accuracy of a delay estimator can be estimated by the Cramér-Rao bound (CRB). The CRB provides us the lower bound of the mean square error (MSE) of any unbiased estimate $\hat{\tau}$ of τ

$$\mathrm{Var}\{\hat{\tau}\} = \mathrm{E}\{(\hat{\tau} - \tau)^2\} \geq CRB$$

The Cramér-Rao lower bound is provided by

$$\mathrm{CRB} = \frac{N_o/2}{(2\pi)^2 E_p \beta^2} = \frac{1}{4\pi^2 \beta^2 \mathrm{SNR}} \tag{1.4}$$

where $E_p$is the average received energy, $N_o/2$ is the two-sided power spectral density from AWGN, $\mathrm{SNR} = \frac{E_p}{N_o/2}$ and β^2 is the second moment of the spectrum $W(f)$ of the transmitted pulse $w(t)$, and is equal to

$$\beta^2 = \frac{\int_{\infty}^{\infty} f^2 |W(f)|^2 df}{\int_{\infty}^{\infty} |W(f)|^2 df}$$

For Gaussian pulses, which are the most commonly used in UWB, we can find in [1] the β^2 corresponding to the nth order derivative of $w_o(t)$ with

$$\beta^{2^{(n)}} = \frac{2n+1}{2\pi\sigma^2}$$

where σ is the variance of the Gaussian pulse and n the order of derivative. Given that in this thesis the pulse used is the second derivative of a Gaussian, the CRB we will use is the following

$$\mathrm{CRB} = \frac{2\pi\sigma^2}{4\pi^2(4+1)\mathrm{SNR}} = \frac{\sigma^2}{10\pi\mathrm{SNR}} \tag{1.5}$$

Note that TOAs accuracy can be improved by increasing SNR or decreasing the pulse width σ. Also, we can express the CRB in distance as

$$\mathrm{CRB}(m) = \sqrt{CRB}c$$

where c is the speed of light .We can observe the relationship between CRB in distance and SNR for various pulse widths in figure 1.6.

As we have seen, TOA is the most suitable technique for UWB positioning. However, in the system discussed in this thesis we can't assume synchronization between transmitter and each receiver. So, we need a technique that exploits fine time resolution and doesn't need synchronization between source and destination. This technique is time difference of arrival (TDOA).

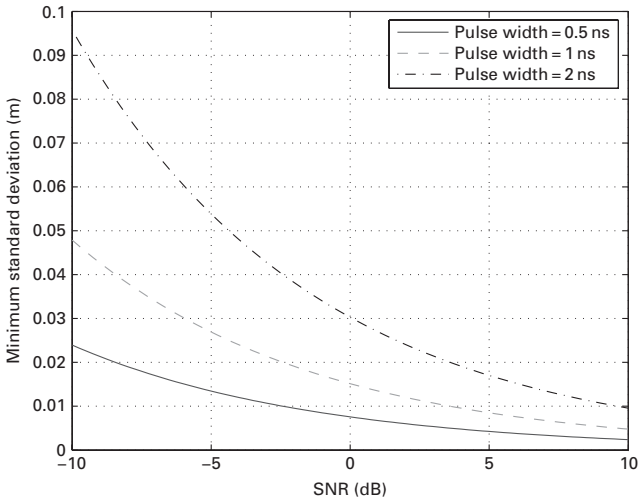


Figure 1.6: CRB versus SNR for different pulse widths.

TDOA In this technique the time difference at which the signal from the transmitter arrives at two different receivers is measured. Given that the difference is measured between transmitters, this technique only requires synchronization between receivers, which is easier to achieve compared to synchronization between transmitters and receivers. Each time difference is converted into a hyperboloid with the constant distance between both receivers. The estimated position is the intersection of a number of hyperboloids. In two-dimensional positioning, at least two hyperboloid are required. This means that at least two pairs of receivers are required and this is achievable with three receivers.

We can obtain a TDOA measurement by estimating relative TOA at each reference node and then computing difference between the two estimated TOA. Since reference nodes are synchronized, the timing offset is the same for each TOA estimation. So, let $r_1(t)$ and $r_2(t)$ be the received signals at receiver 1 and 2, τ_1 the estimated relative TOA from $r_1(t)$ and τ_2 the estimated from $r_2(t)$. Given this, the TDOA measurement can be obtained as

$$\hat{\tau}_{TDOA} = \hat{\tau}_1 - \hat{\tau}_2$$

The accuracy of the TDOA estimation is the same than the accuracy from TOA measurements seen in 1.3.

Another way to obtain a TDOA measurement by developing cross-correlations between received signals $r_1(t)$ and $r_2(t)$. So, $\hat{\tau}_{TDOA}$ can also be obtained as

$$\hat{\tau}_{TDOA} = \arg\max_{\tau} \{r_1(\tau) * r_2(-\tau)\} = \arg\max_{\tau} \int_{t_i} r_1(t) r_2(t+\tau) dt \qquad (1.6)$$

Now, we will see one method to estimate the position assuming we have three receivers

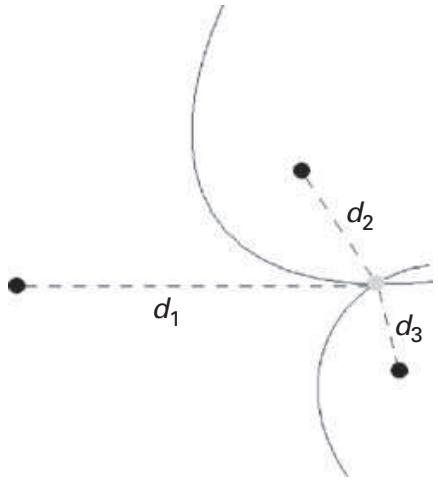


Figure 1.7: TDOA positioning with two hyperbolas

1.1.3.2 Position derivation with TDOA

In [3] we find a procedure to derive the position. Lets the coordinates from three receivers be, without loss of generality

$$R_1 = (x_1,\, x_2) = (0,\, 0)$$
$$R_2 = (y_1,\, y_2) = (0,\, y_2)$$
$$R_3 = (x_3,\, y_3)$$

and the coordinates from the object to be positioned

$$T = (x, y)$$

Given $\hat{t}_1$, $\widehat{t_2}$ and $\hat{t}_3$the relative TOA estimated at receivers 1, 2 and 3 respectively, we can define

$$d_{n,m} = d_m - d_n = c(\hat{t}_m - \hat{t}_n) = c\hat{\tau}_{TDOA_{m,n}}$$

where d_n is the distance from receiver n to the target.

With this, we can define two hyperbolas, for example the one generated by receiver 1 and receiver 2 and the one by receiver 1 and receiver 3. So, we have

$$d_{1,2} = \sqrt{x^2 + (y - y_2)^2} - \sqrt{x^2 + y^2}$$
$$d_{1,3} = \sqrt{(x - x_3)^2 + (y - y_3)^2} - \sqrt{x^2 + y^2}$$

At this point, we can plot two hyperbolas as seen in figure 1.7.

In order to resolve the ambiguity from the two interception points, we can add a third hyperbola, as seen in figure 1.8.

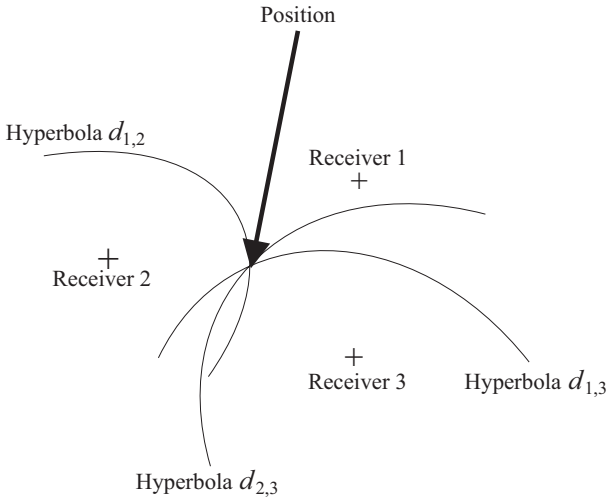


Figure 1.8: TDOA positioning with two hyperbolas

$$d_{2,3} = \sqrt{(x - x_3)^2 + (y - y_3)^2} - \sqrt{x^2 + (y - y_2)^2}$$

or either the distance between receiver 1 and the object

$$d_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2}$$

Any of these combinations, lead to quadratic equation problem. This can be solved by non-linear least-square methodology.

1.1.3.3 Proved Applications

Work in impulse radio UWB (IR-UWB) for accurate positioning has already been done, like the testbed in [4]. In this testbed, conducted at Université Catholique de Louvain (UCL), ranging and positioning experimental measurements have been carried. Ranging experiments have been done with one transmit antenna and one receive antenna, while positioning experiments have been carried out out with one transmit antenna and three receive antennas. Their results show that variances for both ranging and positioning are close to CRLBs when the first multipath component (MPC) is the strongest one and the mainlobe of the pulse is stronger than sidelobes. When the strongest MPC is not the first, which means a false MPC, notable positioning errors are obtained. To detect the first MPC correctly, they propose a threshold based TOA estimator. Also, they have concluded that positioning accuracy can always be improved by increasing the number of receive antennas.

1.1.4 Conclusion

In this section we have seen that UWB consumes very low power and transmits radio signals of very short duration pulses. It has a huge potential for localization techniques due to the simple fact of working with bandwidths in the order of GHz. Moreover, given its lower power requirements its an easy and economical technology. We have also seen that given the spectrum limitation from frequency regulating agencies, the use of UWB is more suitable in indoor scenarios. On the other hand, in terms of localization techniques, TOA seems to be the approach that best exploits the qualities of UWB. In addition, if we can not achieve synchronization between receiver and transmitter, we can use TDOA estimation.

1.2 Motivation

Ultra-wideband has proved to be an interesting choice for positioning applications thanks to its wide spectrum. Also, the FCC regulation in 2002 that allowed the unlicensed use from 3.1 to 10.6 GHz in the United States and, afterward, the ECC regulation in 2006 that allowed the same use from 6 to 8.5 GHz, have definitely impulsed UWB utilization in localization.

As we have seen, localization of one transmitting object is feasible via IR-UWB, although localizing multiple sources might be also possible. Therefore, the aim of this project is to determine whether it is possible to locate multiple objects via IR-UWB .

So, the main motivation of this thesis is to determine the feasibility of obtaining good variance values in UWB based positioning of multiple sources. On the one hand, we need to study a multiple access UWB modulation. Specifically, we will use a time-hopping (TH) pulse position modulation (PPM) that uses a pseudo-random code or time-hopping code (THC) to distinguish users. So, one objective of this thesis is to design an optimum THC that improves the variance of the estimation. On the other hand, we need to analyze the way the estimation is done and improve it. Given this, another objective is to improve the estimator so better variances at the output are obtained. The final step is to carry out simulations and compare results to CRB, and, with this information, to evaluate the feasibility of multiple sources positioning in UWB.

1.3 Outline

This document has been divided in the following chapters. In **chapter 2** the TH-PPM modulation is introduced, the transmitted signal that we will use is defined, the received signal and a basic estimator are also shown and finally the propagation model is exposed. **Chapter 3** focuses on the characteristics of THC with the aim of defining conditions optimal THC must satisfy. In **chapter 4** we propose two different generators of optimal codes. In **chapter 5** we propose an improved estimator, which is formed by two concatenated estimators. In

chapter 6 the simulation results are presented. Finally, in **chapter 7** we expose the conclusions of the thesis and suggest future work.

Chapter 2

Impulse-Radio Ultra-Wideband signal

2.1 Introduction

IR-UWB is a type of UWB communications system that transmits UWB pulses with a low duty cycle. The main advantage of IR is that it does not need RF components in transceivers, consequently implementation is simple and economical. In an IR-UWB system, a number of pulses are transmitted per information symbol and information is usually determined by the positions or the polarities of the pulses, such as pulse amplitude modulation (PAM), pulse position modulation (PPM), on-off keying (OOK), etc. In this chapter we will introduce TH-PPM modulation.

2.2 Time-Hopping Pulse Position Modulation

Now, we will explain step by step how a TH-PPM UWB signal is formed [9]. In TH-PPM one pulse $w(t)$ is transmitted in each frame of time T_f

$$s(t) = \sum_{i=-\infty}^{+\infty} w(t - iT_f)$$

To modulate data we add a small time shift δ in the pulse position when we want to transmit the symbol 1 and no time shift if the desired symbol is 0. Also, since this is an oversampled modulation system with Ns pulses transmitted per symbol, the modulating data symbol changes only every Ns hops

$$s(t) = \sum_{i=-\infty}^{+\infty} w(t - iT_f - \delta d_n([i/N_s]))$$

Finally, to avoid spectral lines and to distinguish users we add a time-hopping code c_n where n represents the n^{th} user. With this, the frame time T_f is divided in a certain number of chip slots of duration Tc. So, the time-hopping code (THC) can switch the position of our pulse between any of the chip slots. Also, a THC has a period N_c, that should be equal to the number of pulses transmitted per symbol N_s. So, the final expression of a TH-PPM modulation is

$$s_n(t) = \sum_{i=-\infty}^{+\infty} w\left(t - iT_f - c_n(i)T_c - \delta d_n([i/N_c])\right) \tag{2.1}$$

where:

- n is the n^{th}user
- $w(t)$is the transmitted Gaussian pulse or monocycle
- T_f is the frame time or pulse repetition time
- c is the periodic pseudo-random code
- T_c is the chip duration. T_c should satisfy $T_c > T_w$
- δ is the added time shift when the symbol is 1
- d is the transmitted symbol. It belongs to a binary data sequence

$$d_{n(i)} \in \{0, 1\}$$

Given that the signal repeats itself each N_c frame times, we can re-express equation 2.1 as

$$s_n(t) = \sum_{i=-\infty}^{+\infty} b_n\left(t - iN_cT_f - \delta d_n(i)\right)$$

$$b_n(t) = \sum_{j=0}^{N_c-1} w(t - jT_f - c_n(i)T_c)$$

where $b_n(t)$ is the spreading of the N_c pules that contain one code.

In the next sections we discuss the time-hopping code and the data modulation.

2.2.1 Pseudo-random time-hopping

In order to distinguish between users, a THC is employed. So, the n^{th} user will use code c_n. These codes are defined as pseudo-random codes and have a period equal to N_c, as seen in equation 2.2. This means that each code has a length equal to N_c and consequently each transmitted symbol is conformed by N_c pulses[7].

$$c_n(i + jN_c) = c_n(i) \ \forall i, j \epsilon \mathbb{Z} \tag{2.2}$$

The sequence $c_n(i)$ values are integers in the range

$$0 \leq c_n(i) < N_h - 1$$

Given this, time-hopping codes add a time shift from 0 to $N_h T_c$seconds. Consequently, the time frame, T_f must satisfy $T_f \geq N_h T_c$. It is not recommended to use a T_f equal to $N_h T_c$, due to guard time requirements. However, to simplify, in this thesis we use $T_f = N_h T_c$.

Also, the use of this codes determines the period of the UWB signal. In fact, since the hopping code has a period N_c, the transmitted signal in equation 2.1 is periodic with period

$$T_p = N_c T_f = N_c N_h T_c \tag{2.3}$$

2.2.2 Data modulation

The binary data sequence $d_n(i)$ is assumed to be independent and identically distributed. Also, given that this is an oversampled modulation each symbol is repeated N_c times with one pulse in each time frame. In this modulation, when the desired transmitted symbol is 1 an additional time shift of value equal to δ is added to the pulse, and when the transmitted symbol is 0 there is no added delay.

Since the objective of the system discussed in this thesis is to localize the position of n transmitters, there is no need to modulate data. Being this so, the transmission of the parameter $\delta d_n(\lfloor i/N_c \rfloor)$ in equation 2.1 can be omitted. The definitive transmitted signal is

$$s_n(t) = \sum_{i=-\infty}^{+\infty} w\left(t - iT_f - c_n(i)T_c\right) \tag{2.4}$$

an the re-expressed version is

$$s_n(t) = \sum_{i=-\infty}^{+\infty} b_n\left(t - iN_c T_f\right) \tag{2.5}$$

$$b_n(t) = \sum_{j=0}^{N_c - 1} w(t - jT_f - c_n(i)T_c) \tag{2.6}$$

In figure 2.1 we can observe the transmitted signal from one user with this configuration: $N_c = 3$, $N_h = 5$, $T_f = 6T_c$ and time-hopping code $c_1 = [0, 4, 1]$.

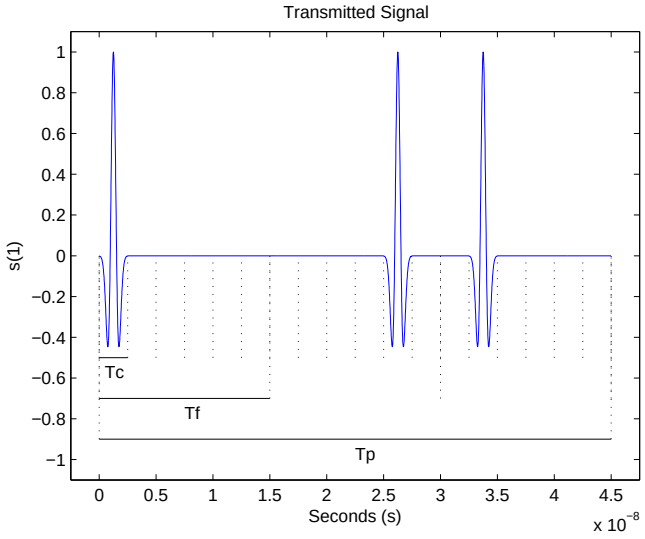


Figure 2.1: Transmitted signal from one user.

2.3 Received signal

The received signal in a given antenna when N_u users are active can be defined as

$$r(t) = \sum_{n=1}^{N_u} A_n s_n \left(t - \theta_n\right) + n(t)$$

where A_n represents the path attenuation of user n^{th} transmitted signal, θ_n accounts for the time asynchronism between the different users, and $n(t)$ is Additive White Gaussian Noise (AWGN). The time asynchronism θ_n is defined as a uniformly distributed variable over a code period [7].

$$0 \leq \theta_n < T_p$$

If we are interested in the data transmitted by user 1, we can re-write the received signal as

$$r(t) = A_1 s_1(t - \theta_1) + \sum_{k=2}^{N_u} A_k s_k \left(t - \theta_k\right) + n(t) = A_1 s_1(t - \theta_1) + n_{tot}(t)$$

where $n_{tot}(t) = \sum_{k=2}^{N_u} A_k s_k \left(t - \theta_k\right) + n(t)$. The term $\sum_{k=2}^{N_u} A_k s_k \left(t - \theta_k\right)$ represents the multi-access interference (MAI).

An example of a received signal can be seen in figure 2.2. In these pictures, the number of users is $N_h = 4$, 3 periods T_p are represented and asynchronism

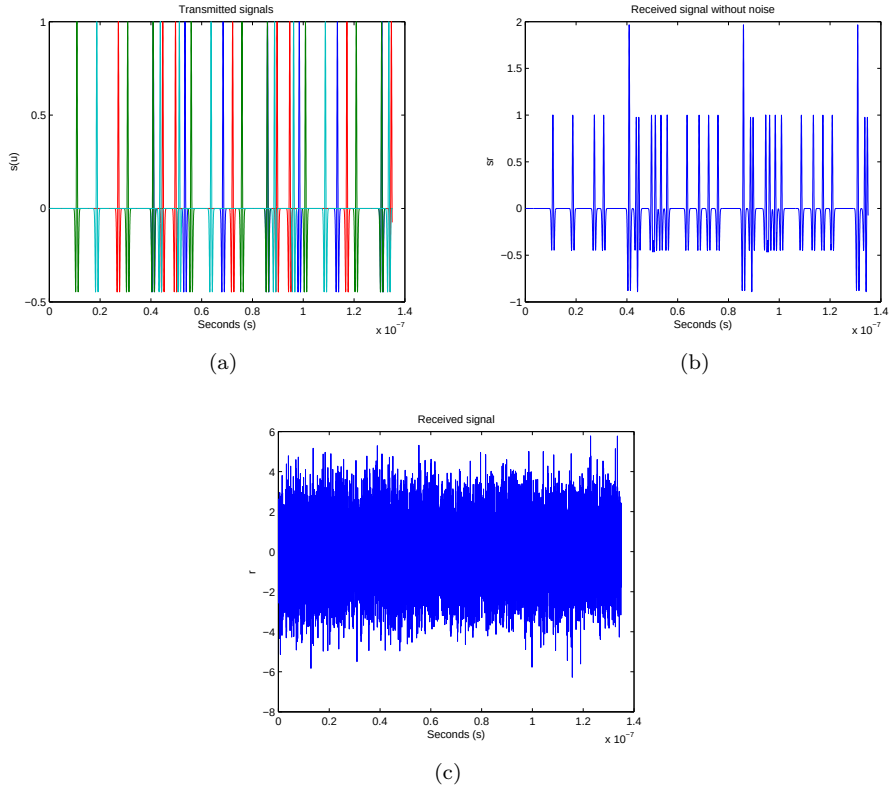


Figure 2.2: (a) Transmitted signals (b) Received signal without noise (c) Received signal with noise

has been included. We can observe in the received signal without noise that two pulses are colliding, and this can be seen three times due to the number of periods we are observing.

2.3.1 Estimator

In order to detect a transmitted signal from an user n, a receiver with the criteria shown in equation 1.3 can be used

$$\hat{\tau}_{TOA_n} = \arg\max_{\tau} R_{r\upsilon_1}(\tau) = \arg\max_{\tau} \{r(\tau) * \upsilon_n(-\tau)\} = \arg\max_{\tau} \int_{t_i} r(t)\upsilon_n(t+\tau)dt \tag{2.7}$$

where $\upsilon_n(t)$ is the receiver template for user n, t_i is the observation interval $[-T_c, T_c]$. $\upsilon_n(t)$ can be expressed as

$$\upsilon_n(t) = \sum_{i=0}^{N_c-1} w\left(t - iT_f - c_n(i)T_c\right)$$

Note that $\upsilon_n(t)$ is exactly the same as $b_n(t)$ seen in equation 2.6, but we name them differently due to their different purposes.

We can expand the correlation as

$$R_{r\upsilon_n}(\tau) = \{r(\tau) * \upsilon_n(-\tau)\} = \int_{t_i} r(t)\upsilon_n(t+\tau)$$

$$= \int_{t_i} \left(\sum_{n=1}^{N_u} A_n s_n\left(t - \theta_n\right) + n(t) \right) \upsilon_n(t+\tau)$$

and if we need to estimate TOA from user 1

$$R_{r\upsilon_1}(\tau) = \int_{t_i} A_1 s_1(t-\theta_1)\upsilon_1(t+\tau)dt + \int_{t_i} \sum_{k=2}^{N_u} A_k s_k\left(t-\theta_k\right)\upsilon_1(t+\tau)dt + \int_{t_i} n(t)\upsilon_1(t+\tau)dt$$

$$R_{r\upsilon_1}(\tau) = A_1 R_{s_1\upsilon_1}(\tau+\theta_1) + \sum_{k=2}^{N_u} A_k R_{s_k\upsilon_1}(\tau+\theta_k) + R_{n\upsilon_1}(\tau) \tag{2.8}$$

By doing this correlation, we are looking for the grade of relationship between all transmitted signals plus noise and user n's template. If the system has been well designed and noise is not too high, we should be able to find a maximum in the interval t_i that indicates with a certain accuracy the instant of time of arrival.

In figure 2.3 we can see an example of the output of the correlator receiver, corresponding to the example seen in figure 2.2. It can be noticed than only two major peaks are observed, the third is not seen due to the time asynchronism.

2.4 Propagation model

The propagation model we use is the single free-space model, like the shown in [6], where the transmitted signal reaches the receiver through a direct path. The path loss can be written as

$$Pl = \left(\frac{c}{4\pi f_c}\right)^2 \tag{2.9}$$

where c is the speed of light and f_c is the central frequency.

We can express the SNR of the system as

$$SNR = \frac{P_t Pl T}{N_0/2\upsilon_{nf}\upsilon_{il}} \tag{2.10}$$

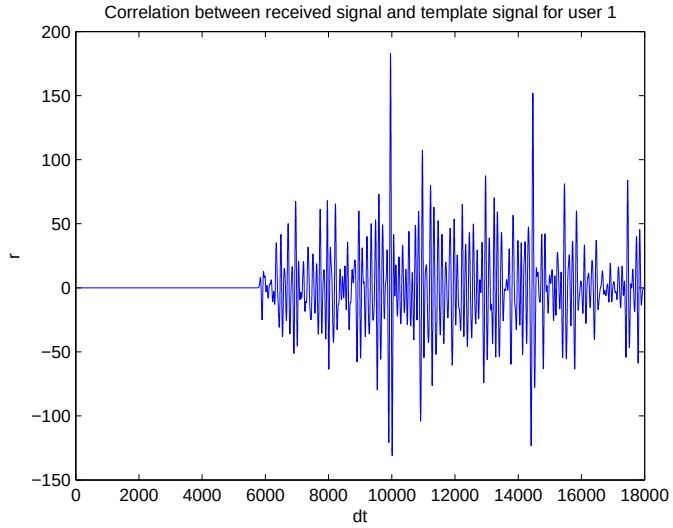


Figure 2.3: Correlation between received signal and one user template

where P_t is the transmitted power, T is the repetition period, $N_0/2$ is the bilateral PSD of $n(t)$, v_{nf} the noise figure and v_{il} the implementation loss.

Note that we have not taken into account multi-path propagation. Although UWB is naturally more robust to multi-path than other technologies it does have an impact. Given that the objective of this work is to determine the feasibility of multiple source UWB based positioning in a first step, we have decided to use a single-path model in order to see if results are satisfactory.

The maximum allowed transmitter power is given by the spectral mask specified by frequency regulating agencies. In Europe, as seen in 1.3, we can transmit in an unlicensed band of 2.5 GHz with a maximum PSD of -41.3 dBm/MHz ($7.4131 \cdot 10^{-14}$ W/Hz). Consequently

$$P_t \leq 7.4131 \cdot 10^{-14} \text{ W/Hz} \cdot 2.5 \text{ GHz} = 0.18533 \text{ mW}$$

The repetition period T must be lower or equal than $1/1$ MHz $= 1\,\mu$s. In figure 2.4 we can see SNR with respect to d. We use $Pt = 0.18533$ mW, $T = 200\,\mu$s, $f_c = 1.29$ GHz, $N_0/2 = -114$ dBm/MHz, $v_{nf} = 6.6$ dB and $v_{il} = 2.5$ dB.

2.5 Conclusion

In this chapter we have learned TH-PPM modulation. Also, we have defined the transmitted signal that will be used in the rest of the thesis. We have show the received signal and we have been able to separate the estimator output in terms, letting us to know is the useful information, which is the multiple user

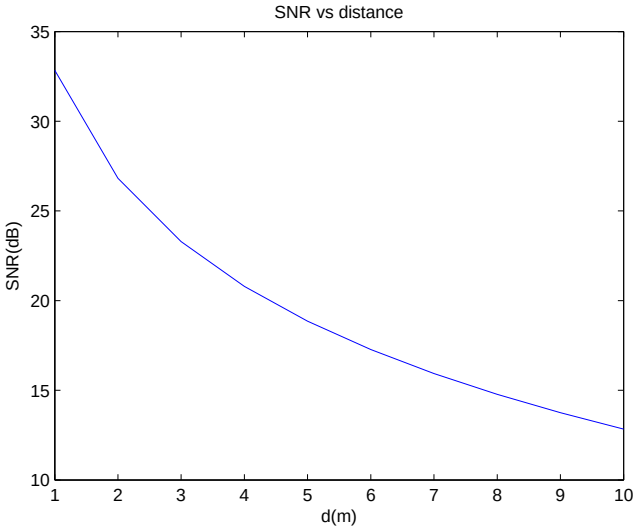


Figure 2.4: SNR vs distance

interference and which is the noise. Finally, we have given the propagation model and computed the maximum transmitter power.

Chapter 3

Optimum Time-Hopping code

3.1 Introduction

In the previous chapter we have mentioned that time-hopping codes should be pseudo-random codes in order to avoid collisions in multiple access. In this chapter we will focus on the characteristics of this codes, we will see their influence to the correlation output and we will extract conditions to obtain optimal codes.

3.2 Time-hopping code notation

In order to understand how collisions occur due to time-hopping codes we need to establish some new definitions.

As we have seen, time-hopping codes, $c_n(i)$ have a length N_c and possible values from 0 to $N_h - 1$. Also, we can have up to N_u users. Being this so, we can arrange horizontally each user code in the matrix $\mathbf{C}$ of a THC with N_u rows and N_c columns.

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1^T \\ \mathbf{c}_2^T \\ \vdots \\ \mathbf{c}_{N_u}^T \end{bmatrix} = \begin{bmatrix} c_1(1) & c_1(2) & \dots & c_1(N_c) \\ c_2(1) & c_2(2) & \dots & c_2(N_c) \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_u}(1) & c_{N_u}(2) & \dots & c_{N_u}(N_c) \end{bmatrix} \tag{3.1}$$

As an example, a matrix of a time-hopping code with a number of users $N_u = 3$, up to $N_h = 5$ possible values and a length $N_c = 3$ is

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1^T \\ \mathbf{c}_2^T \\ \mathbf{c}_3^T \end{bmatrix} = \begin{bmatrix} c_1(1) & c_1(2) & c_1(3) \\ c_2(1) & c_2(2) & c_2(3) \\ c_3(1) & c_3(2) & c_3(3) \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 4 & 1 \end{bmatrix}$$

We can re-express the representation of these codes using the so-called *developed time-hopping codes* [7] with the following equivalence

$$c'_n(j) = \begin{cases} 1 & \text{if } j = c_n(i) + iN_h,\ 0 \le i < N_c - 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } j = 0,\ 1,\ \cdots,\ N_h N_c - 1 \tag{3.2}$$

With this new definition we can define a matrix $\mathbf{C'}$ as

$$\mathbf{C'} = \begin{bmatrix} \mathbf{c'_1}^T \\ \mathbf{c'_2}^T \\ \vdots \\ \mathbf{c'_3}^T \end{bmatrix}$$

The previous example would be represented as

$$\mathbf{C'} = \begin{bmatrix} \mathbf{c'_1}^T \\ \mathbf{c'_2}^T \\ \mathbf{c'_3}^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Using this notation, we can re-write the equation 2.6 as

$$b_n(t) = \sum_{j=0}^{N_c-1} w(t - jT_f - c_n(i)T_c) = \sum_{j=0}^{N_h N_c - 1} c'_n(j) w(t - jT_c) \tag{3.3}$$

Note that this can also be applied to the template signal $\upsilon_n(t)$.

3.3 Correlation output

Let us retake the expression of the correlation output for user 1 seen in 2.8.

$$R_{r\upsilon_1}(\tau) = A_1 R_{s_1\upsilon_1}(\tau + \theta_1) + \sum_{k=2}^{N_u} A_k R_{s_k\upsilon_1}(\tau + \theta_k) + R_{n\upsilon_1}(\tau)$$

For sake of simplicity, let us suppose that the path A_n attenuation is the same for all users and equal to 1, that the transmitted signal is just a basic spread of N_c pulses, $s_n(t) = b_n(t)$, and that the time synchronism is the same for all users, $\theta_n = 0$. Given this, the correlation output would be as

$$R_{r\upsilon_1}(\tau) = R_{b_1\upsilon_1}(\tau) + \sum_{k=2}^{N_u} R_{b_k\upsilon_1}(\tau) + R_{n\upsilon_1}(\tau) \tag{3.4}$$

Now, using the equation in 3.3, we can compute the correlation of user n^{th} $b_n(t)$ with user m^{th} template signal $\upsilon_m(t)$.

$$R_{b_n \upsilon_m}(\tau) = \int_{-T_p}^{T_p} b_n(t)\upsilon_m(t+\tau) = \int_{-T_p}^{T_p} \sum_{j=0}^{N_h N_c - 1} c'_n(j) w(t - jT_c) \sum_{i=0}^{N_h N_c - 1} c'_m(i) w(t - iT + \tau) dt$$

$$= \sum_{j=0}^{N_h N_c - 1} \sum_{i=0}^{N_h N_c - 1} \int_{-T_p}^{T_p} c'_n(j) w(t - jT_c) c'_m(i) w(t - iT_c + \tau) dt$$

$$= \sum_{j=0}^{N_h N_c - 1} \sum_{i=0}^{N_h N_c - 1} c'_n(j) c'_m(i) \int_{-T_p}^{T_p} w(t - jT_c) w(t - iT_c + \tau) dt$$

$$= \sum_{j=0}^{N_h N_c - 1} \sum_{i=0}^{N_h N_c - 1} c'_n(j) c'_m(i) R_w(\tau - (i-j)T_c)$$

Now we can discretize the correlation using the change of variable $\tau = kT_c$

$$R_{b_n \upsilon_m}[kT_c] = \sum_{j=0}^{N_h N_c - 1} \sum_{i=0}^{N_h N_c - 1} c'_n(j) c'_m(i) R_w[kT_c - (i-j)T_c]$$

$$R_{b_n \upsilon_m}[k] = \sum_{j=0}^{N_h N_c - 1} \sum_{i=0}^{N_h N_c - 1} c'_n(j) c'_m(i) R_w[k - (i-j)] = \{l = i - k\}$$

$$R_{b_n \upsilon_m}[k] = \sum_{j=0}^{N_h N_c - 1} \sum_{l=-j}^{N_h N_c - 1 - j} c'_n(j) c'_m(j+l) R_w[k - l] \tag{3.5}$$

We can assume without loss of generality that the energy of the pulse is one, $R_w[0] = 1$, consequently $R_w[m]$ is as

$$R_w[m] = \begin{cases} 1 & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases}$$

Given that in equation 3.5 pulse auto-correlation is evaluated at $k - l$, the second summation $\sum_{l=-j}^{N_h N_c - 1 - j}$ will only lead to a possible nonzero value when $l = k$. So,

$$R_{b_n \upsilon_m}[k] = \sum_{j=0}^{N_h N_c - 1} \sum_{l=-j}^{N_h N_c - 1 - j} c'_n(j) c'_m(j+l) R_w[k - l] = \{l = k\}$$

$$R_{b_n \upsilon_m}[k] = \sum_{j=0}^{N_h N_c - 1} c'_n(j) c'_1(j+k)$$

$$R_{b_n \upsilon_m}[k] = R_{c'_n c'_m}[k] \tag{3.6}$$

As we can see, the discrete correlation of two periods of N_c pulses from an UWB is the same as the correlation between their developed time-hopping codes.

From 3.4 and 3.6 we can write

$$R_{rv_1}[k] = R_{b_1 v_1}[k] + \sum_{n=2}^{N_u} R_{b_n v_1}[k] + R_{nv_1}[k]$$

$$R_{rv_1}[k] = R_{c'_1}[k] + \sum_{n=2}^{N_u} R_{c'_n c'_1}[k] + R_{nv_1}[k]$$

Observing this last equation we can state that in order to estimate user 1's TOA successfully the term $R_{c'_1}[k]$ needs to be minimum $\forall k \neq 0$ and the term $\sum_{n=2}^{N_u} R_{c'_n c'_1}[k]$ needs the minimum $\forall k$. This, when estimating user 1 TOA all correlations of c'_1 with the rest of the codes are present. Therefore, after the estimation of all users, all possible auto- and cross-correlations of THC will have an impact to the final TOA estimation. This is why it is of special importance to design codes with minimum auto- and cross-correlation.

3.4 Conditions for minimum correlation THC

Let us define the auto-correlation $R_{c'_n}[k]$ and cross-correlation $R_{c'_n c'_m}[k]$ of codes as follows

$$R_{c'_n}[k] = \sum c'_n[l] c'_n[k+l]$$

$$R_{c'_n c'_m}[k] = \sum c'_n[l] c'_m[k+l]$$

The objective is to obtain time-hopping codes with minimum auto- and cross-correlation. Minimum cross-correlation ensures that the pair of codes n and m are absolutely different so they can be distinguished correctly and reduces probability of ambiguity while minimum auto-correlation reduces the probability of ambiguity.

Both $R_{c'_n}[k]$ and $R_{c'_n c'_m}[k]$ belong to the interval [0, R_{max}]. The maximum value of R_{max} for the auto-correlation is Nc, which is in fact its energy, and it belongs to $R_{c'_n}[0]$. It is not a value of interest since we need to know if there is correlation for other values of k, consequently we won't consider the case $k = 0$ for the auto-correlation. The maximum value for the cross-correlation is also N_c which would occur in the case that the pair of codes were equal. The minimum value of R_{max} for both auto- and cross-correlation is 1 since at least one pulse from c'_n will collide with one pulse from c'_m.

So, if we define $R^{max}_{c_{n'}}$ and $R^{max}_{c_{n'} c_{m'}}$ as

$$R^{max}_{c'_n} = \max_{k \neq 0} \left\{ R_{c'_n}[k] \right\}$$

$$R^{max}_{c'_n c'_m} = \max_k \left\{ R_{c'_n c'_m}[k] \right\}$$

we can define the conditions a time-hopping code must satisfy to obtain minimum auto- and cross-correlation

$$R^{max}_{c'_n} = 1 \ \forall n$$

$$R^{max}_{c'_n c'_m} = 1 \ \forall n, m$$

3.4.1 Maximum correlation matrix

An useful way to detect if a given time-hopping code has minimum auto- and cross-correlation is computing the maximum correlation matrix. Given a number of users N_u, the maximum correlation matrix M is a square matrix $N_u \times N_u$ and is as follows

$$\mathbf{M} = \begin{bmatrix} R^{max}_{c'_1} & R^{max}_{c'_1 c'_2} & \cdots & R^{max}_{c'_1 c'_{N_u}} \\ R^{max}_{c'_2 c'_1} & R^{max}_{c'_2} & \cdots & R^{max}_{c'_2 c'_{N_u}} \\ \vdots & \vdots & \ddots & \vdots \\ R^{max}_{c'_{N_u} c'_1} & R^{max}_{c'_n c'_m} & \cdots & R^{max}_{c'_{N_u}} \end{bmatrix}$$

The main diagonal is filled with the maximum auto-correlations and the rest of the matrix contains maximum cross-correlations. Also, given that cross-correlation satisfies

$$R_{c'_n c'_m}[k] = \sum c_{n'}[l] c_{m'}[k+l] = \{k + l = i\} = \sum c_{m'}[i] c_{m'}[-k+i] = R_{c'_m c'_n}[-k]$$

and

$$\max_k \left\{ R_{c'_n c'_m}[k] \right\} = \max_k \left\{ R_{c'_n c'_m}[-k] \right\}$$

we can conclude that the maximum correlation matrix is symmetric.

Following the previous example, we can now compute the maximum correlation matrix from time-hopping code $c = [4\,0\,0;\ 0\,2\,0;\ 0\,4\,1]$

$$M = \begin{bmatrix} \max_{k \neq 0} \left\{ R_{c'_1}[k] \right\} & \max_k \left\{ R_{c'_1 c'_2}[k] \right\} & \max_k \left\{ R_{c'_1 c'_3}[k] \right\} \\ \max_k \left\{ R_{c'_1 c'_2}[k] \right\} & \max_{k \neq 0} \left\{ R_{c'_2}[k] \right\} & \max_k \left\{ R_{c'_2 c'_3}[k] \right\} \\ \max_k \left\{ R_{c'_1 c'_3}[k] \right\} & \max_k \left\{ R_{c'_2 c'_3}[k] \right\} & \max_{k \neq 0} \left\{ R_{c'_3}[k] \right\} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

As we can see, this code satisfies both minimum auto- and cross-correlation.

3.5 Conclusion

In this chapter we have seen in detail the characteristics of a THC. We have determined its impact to the correlation output, and with this information we have been able to state conditions an optimum code must satisfy. Also, we have provided an useful way to determine if a given code satisfies these conditions.

Chapter 4

Time-hopping code generators

4.1 Introduction

In chapter 3 we have defined the conditions to obtain minimum auto- and cross-correlation THC. Now, we will explain two time-hopping code generators based on these conditions. The backtracking generator finds the maximum possible number of coexisting codes within a given configuration while the fast generator quickly finds one solution. We will explain the procedure of both and we will compare them in the conclusion.

4.2 Backtracking generator

Backtracking should not be confused with brute-force or exhaustive search. In brute-force all possible candidates for the solution are enumerated and then each candidate is checked whether satisfies the problem condition. In the other hand, in backtracking large sets of solutions can be discarded without being explicitly enumerated. In fact, in this generator, the backtracking algorithm is applied in the last step, when many solutions have already been discarded and the remaining solutions are ordered.

Let us now define the procedure.

4.2.1 Procedure

As we have seen previously, a time-hopping code is defined by three variables: the maximum range of values N_h, the number of active users or transmitters N_u and the period or length of the code N_c. So, if we fix two variables we can find, if exists, the value of the third variable that satisfies the conditions of minimum auto- and cross-correlation. For sake of simplicity we have decided to fix N_h and

N_c and consequently we have the objective to find the largest possible number of users N_u that can coexist within these conditions.

1. Choose a fixed value for the pair of variables N_h and N_c.

2. Generate all existing codes using the previous values of N_h and N_c. With these values fixed, there will be $N_h^{N_c}$ possible codes . We generate the matrix $\mathbf{C}$

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1^T \\ \mathbf{c}_2^T \\ \mathbf{c}_3^T \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{c}_{N_h^{N_c}}^T \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & 2 \\ \vdots & \dots & \vdots & \vdots \\ 0 & \dots & 0 & N_h-1 \\ 0 & \dots & 1 & 0 \\ \vdots & \dots & \vdots & \vdots \\ 0 & \dots & 1 & N_h-1 \\ 0 & \dots & 2 & 0 \\ \vdots & \dots & \vdots & \vdots \\ 0 & \dots & 2 & N_h-1 \\ \vdots & \dots & \vdots & \vdots \\ 0 & \dots & N_h-1 & N_h-1 \\ \vdots & \dots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \\ \vdots & \dots & \vdots & \vdots \\ N_h-1 & \cdots & N_h-1 & N_h-1 \end{bmatrix}$$

   The number of columns is determined by N_c and the number of rows by $N_h^{N_c}$

3. Convert the previous matrix to a new one that let us compute distance between codes in an easy way. We define the matrix $\widetilde{\mathbf{C}}$ and it can be generated using the following formula to each code or row

$$\widetilde{\mathbf{c}}_{\mathbf{n}}^T = \begin{bmatrix} c_n(0) & c_n(1)+N_h & c_n(2)+2N_h & \cdots & c_n(N_c)+N_cN_h \end{bmatrix}$$

   The dimensions of matrix $\widetilde{\mathbf{C}}$ are the same ones as $\mathbf{C}$, $N_h^{N_c} \times N_c$.

4. Calculate the distance between all values in each row from matrix $\widetilde{\mathbf{C}}$. Using the previous notation, computing the distances is as easy as calculating the difference between all possible combinations. The number of distances, N_d, we will need to compute corresponds to the number of combinations of N_c things taken 2 at a time, that is

$$N_d = \frac{N_c!}{2!(N_c-2)!}$$

We can define $d_n(i,j)$ as the distance between $\widetilde{c}_n(i)$ and $\widetilde{c}_n(j)$ with the following expression

$$d_n(i,j) = \widetilde{c}_n(i) - \widetilde{c}_n(j),\ \forall i > j$$

With this, we can compute all distances from the matrix we have been working using the distance vector expression $\mathbf{d}_n$ with length N_d

$$\mathbf{d_n} = \begin{bmatrix} d_n(1,2) \\ d_n(1,3) \\ \vdots \\ d_n(1,N_c) \\ \vdots \\ d_n(2,3) \\ d_n(2,N_c) \\ \vdots \\ d_n(N_c-1,N_c) \end{bmatrix}$$

We will obtain a matrix of distances defined $\mathbf{D}$ with $N_h^{N_c}$ rows and N_d columns

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \vdots \\ \mathbf{d}_{N_h^{N_c}}^T \end{bmatrix}$$

5. Remove codes with maximum auto-correlation higher than 1, this is

$$\max_{k \neq 0} \left\{ R_{c'_n}[k] \right\} > 1\ \forall n$$

We can state that if any value of a given distance vector is repeated, auto-correlation will be higher than 1 for a certain k. So, if we have a repeated distance, when computing auto-correlation there will be at some position at least two pulses colliding, which would lead us to a maximum auto-correlation of at least 2.
For example, the code [0 0 0] has a distance vector [N_h $2N_h$ N_h] and, consequently, it is not a minimum auto-correlation code and should be removed.

6. Remove repeated codes. We understand repeated codes as those that have the same values in its distance vector.
For example, the pair of codes [0 0 1] and [1 1 2] have the same distance vector [N_h $2N_h+1$ N_h+1 N_h] and thus, there is one repeated code. This example is very clear since the second code is simply the first one increased by one, thus, the distances are the same. But not all

repeated codes are due to the same reason, for example codes [1 0 0] and [1 1 0] have distance vectors equal to [$N_h - 1$ $2N_h - 1$ N_h] and [N_h $2N_h - 1$ $N_h - 1$] respectively. Both vectors contain the same values, so in practice they are the same code. For sake of simplicity we will keep the code with lower values, in this case, we would keep [0, 0, 1] and [1 0 0].
Note that after removing codes in steps 5 and 6 there will be less than $N_h^{N_c}$ possible codes left.

7. Generate a list of compatible codes. We can say a pair of codes are compatible if they don't have any value in common in their distance vector. So, we can generate a matrix that contains compatible codes with first code in the first row, compatible codes with second code in the second row and so forth. For example, we can imagine a situation with 4 codes, where C_1 is compatible with all codes, C_2 is compatible with C_1 and C_4, C_3 is compatible only with C_1, and C_4 is compatible with C_2 and C_4. This would be the matrix of compatible codes

$$\begin{array}{c} c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} \begin{bmatrix} c_2 & c_3 & c_4 \\ c_1 & c_4 & \\ c_1 & & \\ c_1 & c_2 & \end{bmatrix}$$

8. Find the largest number of compatible codes. This is where we use the backtracking algorithm.
Backtracking is a general algorithm for finding all (or some) solutions to some computational problem, that incrementally builds candidates to the solutions, and abandons each partial candidate as soon as it determines the candidate cannot possibly be completed to a valid solution.The backtracking algorithm enumerates a set of partial candidates that, in principle, could be completed in various ways to give all the possible solutions to the given problem. The completion is done incrementally, by a sequence of candidate extension steps. Conceptually, the partial candidates are the nodes of a tree structure, the potential search tree. Each partial candidate is the parent of the candidates that differ from it by a single extension step; the leaves of the tree are the partial candidates that cannot be extended any further. The backtracking algorithm traverses this search tree recursively, from the root down, in depth-first order. We can see how the tree would look like after applying backtracking in figure 4.1. The approach for this example is explained now. We start with c_1 and we know that c_2, c_3 and c_4 are compatible with it. But now, we need to know if $c_{2,}$ c_3 and c_4 are compatible among them. We try first with c_2 , we check its compatible codes: c_1 and c_4 . So the first branch of the tree would be $[c_1c_2c_4]$. We will keep doing this until no more compatible code are found. At that point we step back and try with next son. We can observe this example that the largest number of compatible codes is three and belongs to the unique sequence $c_1c_2c_4$. Note that since we are

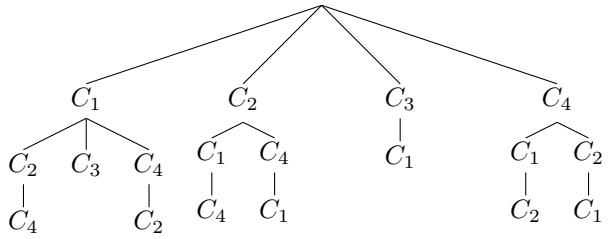


Figure 4.1: Example of a tree structure

allowing permutation with repetition there are $3! = 6$ branches that gives us the same solution.

9. (Optional) We can add some modifications in order to reduce computational time of the backtracking. For example, once the highest parent has tried all the possibilities with on son, they can be mutually eliminated from theirs lists of compatible codes, so no more combination between them will be done.

4.2.2 Example of a backtracking generated code

1. Choose a fixed value for the pair of variables: $N_h = 3$ and $N_c = 3$.
2. Generate all existing codes.
3. Convert the matrix.

4. Calculate matrix of distances.

$$
\begin{bmatrix} \\ \mathbf{c}_1^T \\ \mathbf{c}_2^T \\ \mathbf{c}_3^T \\ \mathbf{c}_4^T \\ \mathbf{c}_5^T \\ \mathbf{c}_6^T \\ \mathbf{c}_7^T \\ \mathbf{c}_8^T \\ \mathbf{c}_9^T \\ \mathbf{c}_{10}^T \\ \mathbf{c}_{11}^T \\ \mathbf{c}_{12}^T \\ \mathbf{c}_{13}^T \\ \mathbf{c}_{14}^T \\ \mathbf{c}_{15}^T \\ \mathbf{c}_{16}^T \\ \mathbf{c}_{17}^T \\ \mathbf{c}_{18}^T \\ \mathbf{c}_{19}^T \\ \mathbf{c}_{20}^T \\ \mathbf{c}_{21}^T \\ \mathbf{c}_{22}^T \\ \mathbf{c}_{23}^T \\ \mathbf{c}_{24}^T \\ \mathbf{c}_{25}^T \\ \mathbf{c}_{26}^T \\ \mathbf{c}_{27}^T \end{bmatrix}
=
\begin{bmatrix} & \mathbf{C} & (2) \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \\ 0 & 2 & 0 \\ 1 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \\ 0 & 2 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 1 \\ 0 & 0 & 2 \\ 1 & 0 & 2 \\ 2 & 0 & 2 \\ 0 & 1 & 2 \\ 1 & 1 & 2 \\ 2 & 1 & 2 \\ 0 & 2 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}
\Rightarrow
\begin{bmatrix} & \widetilde{\mathbf{C}} & (3) \\ 0 & 3 & 6 \\ 1 & 3 & 6 \\ 2 & 3 & 6 \\ 0 & 4 & 6 \\ 1 & 4 & 6 \\ 2 & 4 & 6 \\ 0 & 5 & 6 \\ 1 & 5 & 6 \\ 2 & 5 & 6 \\ 0 & 3 & 7 \\ 1 & 3 & 7 \\ 2 & 3 & 7 \\ 0 & 4 & 7 \\ 1 & 4 & 7 \\ 2 & 4 & 7 \\ 0 & 5 & 7 \\ 1 & 5 & 7 \\ 2 & 5 & 7 \\ 0 & 3 & 8 \\ 1 & 3 & 8 \\ 2 & 3 & 8 \\ 0 & 4 & 8 \\ 1 & 4 & 8 \\ 2 & 4 & 8 \\ 0 & 5 & 8 \\ 1 & 5 & 8 \\ 2 & 5 & 8 \end{bmatrix}
\Rightarrow
\begin{bmatrix} & \mathbf{D} & (4) \\ 3 & 6 & 3 \\ 2 & 5 & 3 \\ 1 & 4 & 3 \\ 4 & 6 & 2 \\ 3 & 5 & 2 \\ 2 & 4 & 2 \\ 5 & 6 & 1 \\ 4 & 5 & 1 \\ 3 & 4 & 1 \\ 3 & 7 & 4 \\ 2 & 6 & 4 \\ 1 & 5 & 4 \\ 4 & 7 & 3 \\ 3 & 6 & 3 \\ 2 & 5 & 3 \\ 5 & 7 & 2 \\ 4 & 6 & 2 \\ 3 & 5 & 2 \\ 3 & 8 & 5 \\ 2 & 7 & 5 \\ 1 & 6 & 5 \\ 4 & 8 & 4 \\ 3 & 7 & 4 \\ 2 & 6 & 4 \\ 5 & 8 & 3 \\ 4 & 7 & 3 \\ 3 & 6 & 3 \end{bmatrix}
$$

5. Remove codes with maximum auto-correlation higher than 1.

6. Remove repeated codes.

$$\begin{bmatrix} \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \mathbf{d}_3^T \\ \mathbf{d}_4^T \\ \mathbf{d}_5^T \\ \mathbf{d}_6^T \\ \mathbf{d}_7^T \\ \mathbf{d}_8^T \\ \mathbf{d}_9^T \\ \mathbf{d}_{10}^T \\ \mathbf{d}_{11}^T \\ \mathbf{d}_{12}^T \\ \mathbf{d}_{13}^T \\ \mathbf{d}_{14}^T \\ \mathbf{d}_{15}^T \\ \mathbf{d}_{16}^T \\ \mathbf{d}_{17}^T \\ \mathbf{d}_{18}^T \\ \mathbf{d}_{19}^T \\ \mathbf{d}_{20}^T \\ \mathbf{d}_{21}^T \\ \mathbf{d}_{22}^T \\ \boldsymbol{d}_{23}^T \\ \mathbf{d}_{24}^T \\ \mathbf{d}_{25}^T \\ \mathbf{d}_{26}^T \\ \mathbf{d}_{27}^T \end{bmatrix} = \begin{bmatrix} \mathbf{D} & & \\ 3 & 6 & 3 \\ 2 & 5 & 3 \\ 1 & 4 & 3 \\ 4 & 6 & 2 \\ 3 & 5 & 2 \\ 2 & 4 & 2 \\ 5 & 6 & 1 \\ 4 & 5 & 1 \\ 3 & 4 & 1 \\ 3 & 7 & 4 \\ 2 & 6 & 4 \\ 1 & 5 & 4 \\ 4 & 7 & 3 \\ 3 & 6 & 3 \\ 2 & 5 & 3 \\ 5 & 7 & 2 \\ 4 & 6 & 2 \\ 3 & 5 & 2 \\ 3 & 8 & 5 \\ 2 & 7 & 5 \\ 1 & 6 & 5 \\ 4 & 8 & 4 \\ 3 & 7 & 4 \\ 2 & 6 & 4 \\ 5 & 8 & 3 \\ 4 & 7 & 3 \\ 3 & 6 & 3 \end{bmatrix} \Rightarrow \begin{bmatrix} (5) & & \\ - & - & - \\ 2 & 5 & 3 \\ 1 & 4 & 3 \\ 4 & 6 & 2 \\ 3 & 5 & 2 \\ - & - & - \\ 5 & 6 & 1 \\ 4 & 5 & 1 \\ 3 & 4 & 1 \\ 3 & 7 & 4 \\ 2 & 6 & 4 \\ 1 & 5 & 4 \\ 4 & 7 & 3 \\ - & - & - \\ 2 & 5 & 3 \\ 5 & 7 & 2 \\ 4 & 6 & 2 \\ 3 & 5 & 2 \\ 3 & 8 & 5 \\ 2 & 7 & 5 \\ 1 & 6 & 5 \\ - & - & - \\ 3 & 7 & 4 \\ 2 & 6 & 4 \\ 5 & 8 & 3 \\ 4 & 7 & 3 \\ - & - & - \end{bmatrix} \Rightarrow \begin{bmatrix} (6) & & \\ - & - & - \\ 2 & 5 & 3 \\ 1 & 4 & 3 \\ 4 & 6 & 2 \\ - & - & - \\ - & - & - \\ 5 & 6 & 1 \\ 4 & 5 & 1 \\ - & - & - \\ 3 & 7 & 4 \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \\ 5 & 7 & 2 \\ - & - & - \\ - & - & - \\ 3 & 8 & 5 \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \end{bmatrix}$$

Note that only 8 out of 27 codes are left.

7. List of compatible codes. Before, we can restructure the matrix by removing the empty spaces left by removed codes.

$$\begin{bmatrix} \\ c_1^T \\ c_2^T \\ c_3^T \\ c_4^T \\ c_5^T \\ c_6^T \\ c_7^T \\ c_8^T \end{bmatrix} = \begin{bmatrix} \mathbf{C} & & \\ 1 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{D} & & \\ 2 & 5 & 3 \\ 1 & 4 & 3 \\ 4 & 6 & 2 \\ 5 & 6 & 1 \\ 4 & 5 & 1 \\ 3 & 7 & 4 \\ 5 & 7 & 2 \\ 3 & 8 & 5 \end{bmatrix} \Longrightarrow \begin{matrix} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{matrix} \begin{bmatrix} (7) \\ - \\ c_7 \\ c_8 \\ c_6 \\ - \\ c_4 \\ c_2 \\ c_3 \end{bmatrix}$$

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
	|	|	|		|	|	|
	c_7	c_8	c_6		c_4	c_2	c_3

Figure 4.2: Tree structure for $N_h = 3$ and $N_c = 3$

8. Find the largest number of compatible codes. We generate the tree structure as the one in figure 4.2. As we can, the largest number of compatible codes for this case is two and there exist several combinations.

4.3 Fast generator

The fast generator does not find the maximum number of users, instead, given the number of users N_u and the code period N_c finds a certain value N_h that satisfies the conditions of minimum auto- and cross-correlation. This N_h value it's not the minimum achievable but it is the first that the generator finds.

4.3.1 Procedure

In this generator we will use the *developed time-hopping code* seen in 3.2. So we will mention $\mathbf{c}'_n$ as the developed time-hopping code of $\mathbf{c}_n$. Also, we will start with a certain N_h and we will increase it if is not high enough. Since we want N_u codes we need a minimum N_h such that $N_u = N_h^{N_c}$.

Initial parameters

$$N_u$$

$$N_c$$

$$N_h = \left\lceil e^{\left(\frac{\ln N_u}{N_c}\right)} \right\rceil$$

1. Generate $\mathbf{c}_1$
   (a) Set $c_1(0) = 0$
   (b) Set $c_1(1) = 0$
   (c) Set $c_1(2) = 0$. Check the condition $R^{max}_{c'_1} = 1$. If not meet, increment one by one the value of $c_1(2)$: $c_1(2) = 1$, $c_1(2) = 2$, $\cdots$, $c_1(2) = N_h - 1$, until the condition is satisfied. If no value satisfies the condition increase N_h by one and repeat the whole process.
   (d) Repeat (c) with $c_1(k)$ for $k = \{3, \cdots, N_c - 1\}$
2. Generate $\mathbf{c}_2$
   (a) Set $c_2(0) = 0$

(b) Set $c_2(1) = 0$. Check the condition $R^{max}_{c'_2} = 1$ and $R^{max}_{c'_2 c'_m} = 1$ $\forall m \in$ {Users whose code has already been generated}. If not meet, increment one by one the value of $c_2(2)$: $c_2(2) = 1$, $c_2(2) = 2$, $\cdots$, $c_2(2) = N_h - 1$, until both conditions are satisfied. If no value satisfies the condition increase N_h by one and repeat the whole process.

(c) Repeat (b) with $c_2(k)$ for $k = \{2, \cdots, N_c - 1\}$

3. Repeat 2 with $\mathbf{c}_n$ for $n = \{3, \cdots, N_u\}$

At the end of the process, we will obtain a value N_h that allows a THC matrix $\mathbf{C}$ with minimum auto- and cross-correlation THC.

4.4 Conclusion

We have shown two ways to generate a THC. On the one hand, the backtracking generator has the advantage that always a minimum dimensions code is designed. This means that the obtained THC for N_u users has the minimum period $T_p = N_c T_f = N_c N_h T_c$ given any T_c. This is important in TOA estimation because lower periods will lead us to higher refreshing rates. However, backtracking weakness is its computation time that grows rapidly when higher values of N_h and N_c are used. On the other hand, the fast generator has the inverse qualities of backtracking generator. Its strong point is the speed in generating codes but has the disadvantage of obtaining lower refreshing rates since the obtained period T_p is not minimum. Also, note that even that codes generated with backtracking take more time, you only need to generate it once. In fact, a large database of codes could be created so there would be no need to generate it each time. In the annexe several codes generated with the backtracking algorithm are shown. Another point to note is that using larger periods is not necessarily negative since the probability of collision will decrease.

Chapter 5

Proposed estimator

5.1 Introduction

In this chapter we propose an improved estimator. First, but, we will review the basic estimator since its output is utilized by our proposed estimator. Also, since the proposed estimator is in fact a concatenation of two estimators, we will explain them separately. Note that in all estimators we are assuming the receiver correctly detects the amplitude of the received signal.

5.2 Basic estimator

The basic estimator consists in computing the TOA estimation for each user. We use the equation seen in 2.7 as

$$\hat{\tau}_{TOA_n} = \arg\max_{\tau} \int_{t_i} r(t)\upsilon_n(t+\tau)dt, \text{ for } n \in \{1, 2, \cdots, N_u\}$$

At the end of the process we will have computed

$$\hat{\boldsymbol{\tau}}_{\mathbf{TOA}} = \begin{bmatrix} \hat{\tau}_{TOA_1} & \hat{\tau}_{TOA_2} & \cdots & \hat{\tau}_{TOA_{N_u}} \end{bmatrix}^T \tag{5.1}$$

And we can compute the estimation error vector $\mathbf{e}$ with

$$\mathbf{e}_{\mathbf{TOA}} = \begin{bmatrix} \hat{\tau}_{TOA_1} - \tau_1 & \hat{\tau}_{TOA_2} - \tau_2 & \cdots & \hat{\tau}_{TOA_{N_u}} - \tau_{N_u} \end{bmatrix}^T \tag{5.2}$$

5.3 Maximum cancellation estimator

The estimator proposed exploits the information once the TOA for all N_u users has been determined. At this point, we can know which user has been estimated with the higher value of correlation at its TOA, and thus, we can assume that this user is the most likely to have been more precisely estimated. So, we consider the TOA for that user as the definitive. Then, we can shift the receiver's

template for that user using its TOA estimation, and subtract it to the received signal. Now, we can estimate again the TOA from the rest of the users and, if the removed signal has been removed in a precise enough position, we can expect an improvement since there will be less MAI. After this, we determine which user has de maximum value at its TOA, shift the template and subtract it from the already reduced received signal. We repeat the process until only one user is left.

The procedure of this estimator is detailed below.

5.3.1 Procedure

First we need to define some initial parameters. Given that the procedure ends after a number of loops equal to the number of users N_u, we define the parameter k that represents the loop in which the process is. So, we can define $n^{(k)}$ as the set of users that belong to n in the loop k and $r^{(k)}(t)$ as the remaining of the received signal $r(t)$ in the loop k.

Initial parameters

$$r^{(1)}(t) = r(t)$$
$$n^{(1)} = \{1, 2, \cdots, N_u\}$$

$$k = 1$$

1. Estimate TOA for users that belong to $n^{(k)}$. As seen in equation 2.7 we will compute

$$\hat{\tau}_{TOA_m} = \arg\max_{\tau} \int_{t_i} r^{(k)}(t) \upsilon_m(t+\tau) dt, \text{ , for } m \in n^{(k)}$$

2. Store the values of the integral at the point where they are maximum. That is

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_m \\ \vdots \\ \sigma_{Nu} \end{bmatrix} = \begin{bmatrix} \int_{t_i} r(t)\upsilon_1(t+\hat{\tau}_{TOA_1})dt \\ \vdots \\ \int_{t_i} r(t)\upsilon_m(t+\hat{\tau}_{TOA_m})dt \\ \vdots \\ \int_{t_i} r(t)\upsilon_{N_u}(t+\hat{\tau}_{TOA_{N_u}})dt \end{bmatrix} \text{ for } m \in n^{(k)}$$

3. Determine the user n_{max} with the maximum correlation value at its $\hat{\tau}_{TOA}$. This is

$$n_{max} = \arg\max_{j} \sigma_j$$

4. Store the n_{max} user TOA estimation

$$\hat{\tau}_{TOA_{MC_{n_{max}}}} = \hat{\tau}_{TOA_{n_{max}}}$$

5. Add a time shift to the template signal $\upsilon_{n_{max}}(t)$ equal to the correspondent TOA as follows
$$\hat{\upsilon}_{n_{max}}(t) = \upsilon_{n_{max}}(t + \hat{\tau}_{TOA_{n_{max}}})$$

6. Subtract $\hat{\upsilon}_{n_{max}}(t)$ from $r(t)$
$$r^{(k+1)}(t) = r^{(k)}(t) - \hat{\upsilon}_{n_{max}}(t)$$
and remove user n_{max} from the set of users
$$n^{(k+1)} = n^{(k)} - \{n_{max}\}$$

7. Increase the loop indicator k by one
$$k \equiv k + 1$$

8. Repeat the process 1-5 until only one user is left and its TOA has been estimated

After the procedure we will obtain the maximum cancellation TOA estimation vector
$$\hat{\boldsymbol{\tau}}_{\mathbf{TOA_{MC}}} = \left[\begin{array}{cccc} \hat{\tau}_{TOA_{MC_1}} & \hat{\tau}_{TOA_{MC_2}} & \cdots & \hat{\tau}_{TOA_{MC_{N_u}}} \end{array} \right]^T \tag{5.3}$$
Its estimation error is

$$\mathbf{e}_{\mathbf{TOA_{MC}}} = \left[\begin{array}{cccc} \hat{\tau}_{TOA_{MC_1}} - \tau_1 & \hat{\tau}_{TOA_{MC_2}} - \tau_2 & \cdots & \hat{\tau}_{TOA_{MC_{N_u}}} - \tau_{N_u} \end{array} \right]^T \tag{5.4}$$

5.4 Iterative cancellation estimator

This iterative cancellation estimator is applied just after the maximum cancellation estimator. The idea is that given the improved TOA estimation vector $\hat{\boldsymbol{\tau}}_{\mathbf{TOA_{mc}}}$, we can estimate the TOA of one user by removing the rest of user's template signals shifted at their respective $\hat{\tau}_{TOA_{MC}}$. After this, we update the new TOA in the vector and then we do the same process for the next user. Once we have done the process for all users we have finalized one loop. We will keep realizing the same loop until no change in TOA estimation vector is observed.

5.4.1 Procedure

In this case we need two indicators. k represents again the loop indicator and n represents the user indicator. So, let us define $r_n^{(k)}(t)$ as the canceled received signal for estimation of user n in loop k and $\hat{\tau}^{(k)}_{TOA_{IC_n}}$ as the iterative cancellation TOA estimation of user n in loop k. Also, we need to retake the vector obtained after applying maximum cancellation TOA seen in 5.3.

Initial parameters

$$\hat{\boldsymbol{\tau}}^{(1)}_{\mathbf{TOA_{IC}}} = \hat{\boldsymbol{\tau}}_{\mathbf{TOA_{mc}}}$$

$$k = 1$$
$$n = 1$$

1. Remove all template signals from $r(t)$ except the one from user n
$$r_n^{(k)}(t) = r(t) - \sum_{m \neq n} \upsilon_m(t + \hat{\tau}^{(k)}_{TOA_{IC_n}})$$
2. Estimate the TOA for user n using $r_n^{(k)}(t)$
$$\hat{\tau}^{(k)}_{TOA_{IC_n}} = \arg\max_{\tau} \int_{t_i} r_n^{(k)}(t)\upsilon_n(t+\tau)dt$$
3. Increase the user indicator n by one
$$n \equiv n + 1$$
4. Repeat 1-3 until estimation is done for all N_u users.
5. Check if the current loop TOA_{IC} estimation is equal to the previous one
$$\hat{\boldsymbol{\tau}}^{(k)}_{\mathbf{TOA_{IC}}} = \hat{\boldsymbol{\tau}}^{(k-1)}_{\mathbf{TOA}}$$
If so, the process is done. Otherwise, follow the next step.
6. Reset the user indicator and increase the loop indicator
$$n = 1$$
$$k \equiv k + 1$$
7. Repeat 1-6 until the condition in 5 is met.

At the end of the process we will obtain the iterative cancellation TOA estimation vector

$$\hat{\boldsymbol{\tau}}_{\mathbf{TOA_{IC}}} = \begin{bmatrix} \hat{\tau}_{TOA_{IC_1}} & \hat{\tau}_{TOA_{IC_2}} & \cdots & \hat{\tau}_{TOA_{IC_{N_u}}} \end{bmatrix}^T \tag{5.5}$$

with an estimation error equal to

$$\mathbf{e}_{\mathbf{TOA_{IC}}} = \begin{bmatrix} \hat{\tau}_{TOA_{IC_1}} - \tau_1 & \hat{\tau}_{TOA_{IC_2}} - \tau_2 & \cdots & \hat{\tau}_{TOA_{IC_{N_u}}} - \tau_{N_u} \end{bmatrix}^T \tag{5.6}$$

5.5 Conclusion

This chapter has shown the procedure of two estimators that are used together to improve the variance. Also, we have given the notation to compute the estimation error of each estimator.

Chapter 6

Simulation results

6.1 Introduction

This chapter shows the practical part of the thesis. Several simulations have been carried out in order to

- Determine whether the use of minimum auto- and cross-correlation THC improves the TOA estimation variance.
- Determine whether the use maximum and iterative cancellation estimator improves the TOA estimation variance.
- Determine whether the TOA estimation variance for multiple sources is close to the CRB.
- Determine how the obtained TOA variance will affect the variances of the TDOA based positioning.

In the next sections we explain the method used in the simulations, we show the results and finally we give the conclusions.

6.2 Method

6.2.1 Introduction

The objective is to present the variance of a given system with respect to the SNR. The parameters in table 6.1 have been used for the simulations unless otherwise indicated.

As we can see, the number of periods generated is one. This is not particularly realistic because in practice we would be transmitting infinitely and we should estimate the TOA for each received period of the signal. But, due to software limitations in the maximum array size, we have decided to launch N experiments of 1 period instead of one experiment of N periods.

Parameter	Value
Pulse width, T_w	$1ns$
Chip time, T_c	$2.5ns$
Frame time, T_f	$T_c N_h$
Period,T_p	$T_f N_c$
Number of periods T_p generated	1
Time asynchronism,θ_n	[0, T_p]
Path attenuation, A_n	1

Table 6.1: Used UWB parameters

So, one simulation consists in N experiments per SNR value, and we take SNR from 0 to 30 dB.

Moreover, we have set the same path attenuation for all users. If we assume they are all transmitting at the same power, this would mean they are all at the same position in a 2D scenario. Again, this is not realistic, but in fact it represents the worst condition. For example, if a transmitter is very close to the receiver, its signal is going to be detected as the one with maximum power and removed more successfully than if two transmitters are at the same distance to the receiver.

We will use generated time-hopping codes but also randomly generated time-hopping codes in order to compare their performance.

For each experiment we are using different

- Time asynchronism θ_n for each user
- Noise $n(t)$ realization
- THC (Only in the case of randomly generated THC)

So, in a simulation we are averaging over the time delay, the noise and the THC if we want it randomly generated.

6.2.2 Error calculation

As we have seen, each simulation consists in N experiments. For each experiment, we proceed to apply the estimators seen in 5.2, 5.4 and 5.6 for the basic, maximum cancellation and iterative cancellation estimators respectively. We obtain N error vectors such as $\mathbf{e}^{(n)}$ where n represents the n^{th} experiment. So, if we use the basic estimator we obtain the matrix

$$\mathbf{E_{TOA}} = \begin{bmatrix} \mathbf{e}_{TOA}^{(1)T} \\ \mathbf{e}_{TOA}^{(2)T} \\ \vdots \\ \mathbf{e}_{TOA}^{(N)T} \end{bmatrix} = \begin{bmatrix} \hat{\tau}_{TOA_1}^{(1)} - \tau_1^{(1)} & \hat{\tau}_{TOA_2}^{(1)} - \tau_2^{(1)} & \cdots & \hat{\tau}_{TOA_{N_u}}^{(1)} - \tau_{N_u}^{(1)} \\ \hat{\tau}_{TOA_1}^{(2)} - \tau_1^{(2)} & \hat{\tau}_{TOA_2}^{(2)} - \tau_2^{(2)} & \cdots & \hat{\tau}_{TOA_{N_u}}^{(2)} - \tau_{N_u}^{(2)} \\ \vdots & \vdots & & \vdots \\ \hat{\tau}_{TOA_1}^{(N)} - \tau_1^{(N)} & \hat{\tau}_{TOA_2}^{(N)} - \tau_2^{(N)} & \cdots & \hat{\tau}_{TOA_{N_u}}^{(N)} - \tau_{N_u}^{(N)} \end{bmatrix} \tag{6.1}$$

In this matrix users are sorted into columns and experiments into rows. In the same way we obtain matrix $\mathbf{E}_{TOA_{MC}}$ when we using the maximum cancellation estimator and matrix $\mathbf{E}_{\mathbf{TOA_{IC}}}$ when we use iterative cancellation estimator.

6.2.3 TOA Variance calculation

To compute the variance of the estimation $e = \hat{\tau} - \tau$, we use the unbiased estimator

$$\mathrm{Var}\{\mathbf{e}\} = \frac{1}{N-1}\sum_{i=1}^{N}(e_i - \mathrm{E}\{\mathbf{e}\})^2$$

where $\mathrm{E}\{\mathbf{e}\}$ is the expected value of $\mathbf{e}$ and is computed as

$$\mathrm{E}\{\mathbf{e}\} = \frac{1}{N}\sum_{i=1}^{N} e_i$$

Also, given that we have N_u users we could calculate the variance for each user by taking $\mathbf{e}$ as the columns from matrix $\mathbf{E}$. But given that we will be using minimum auto- and cross- correlation codes or either randomly generated codes in a time delay and noise averaging scenario, each user error should be independent from one experiment to another. Moreover, we are more interested in an overall variance from all users that in the variance from only one user. Being this so, we will take $\mathbf{e}$ as a vector containing all elements from matrix $\mathbf{E}$. Thus, if we realize a simulation with N experiments and N_u users we will have $N \cdot N_u$samples to compute the variance.

6.2.4 TDOA positioning variance

Since we have only used one receiver in the simulations and, in order to localize via TDOA at least three receivers are required, we will do the following. Using the theory explained in 1.1.3.2, let us suppose that the scenario is a room with dimensions 6 m $\times$ 6 m and the target to localize is positioned at $T = (x, y) = (3, 3)$. Also, we put an antenna on each corner

$$R_1 = (0,\ 0)$$
$$R_2 = (0, 6)$$
$$R_3 = (6,\ 0)$$
$$R_4 = (6,\ 6)$$

In this situation, given that the transmitter is in the middle of the room each antenna is at the same distance from the target and thus, the received signal has the same SNR at each antenna. Now, we can use the error matrix $\mathbf{E}$ for a given SNR (seen in 6.1) obtained from the simulations. We can take 4 samples randomly from $\mathbf{E}$ and use them as the estimation error at each receiver. Let us suppose we take the values e_1, e_2, e_3 and e_4. We can now define three hyperbolas as

$$d_{1,2} = (e_2 - e_1)c$$
$$d_{1,3} = (e_3 - e_1)c$$
$$d_{1,4} = (e_4 - e_1)c$$

where c is the speed of light. The estimated position $\hat{T} = (\hat{x}, \hat{y})$ is obtained by non-linear least-square methodology. We will finish the process when all samples from $\mathbf{E}$ have been used. Let us define M as the number of positions we can estimate from a given matrix of error. We can now compute the distance error of each estimated position as

$$e_{TDOA} = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2}$$

At the end of the process we will obtain the TDOA distance error vector for a given SNR

$$\mathbf{e_{TDOA}} = [\ e_{TDOA_1} \quad e_{TDOA_2} \qquad e_{TDOA_M}\ ]^{\ T}$$

6.3 Results presentation

In each figure we show the variance with respect to the SNR. We represent the $\log_{10}$(logarithm base 10) of the variance in order to distinguish more precisely for high values of SNR. However, instead of the variance we plot the square of the variance so we can plot a new axe in the right with the variance of the distance just by multiplying by the speed of light.

$$\mathrm{Var}(m) = \sqrt{\mathrm{Var}(s^2)}c$$

where c is speed of light.

Finally, we plot another axe above the figure. This axe represents the distance the source transmits from. It is obtained from the propagation model seen in 2.9 and 2.10. Note that this information is for guidance only since it depends on several variables such as the bandwidth B, the central frequency f_c and the repetition period T.

6.4 Results

6.4.1 One user

TOA estimation variance for one user, $N_h = 6$ and $N_c = 3$

First, we will see how the variance for one user ($N_u = 1$) is. Note that since there is only one user the maximum cancellation and iteration estimator do not apply.

Two simulations have been carried out, one using a generated code and the other one with random codes. The parameters are $N_h = 6$ and $N_c = 3$. The number of experiments is $N = 10000$, so there are 10000 samples per SNR value. The generated code used is [3 4 0]. Results can be seen in figure 6.1.

The first we observe is that both simulations reach the CRB around 17 dB and they stay quite far from it for low SNR. Given this, the Cramér–Rao bound seems to be too optimistic for low SNR, especially in this case since there is only

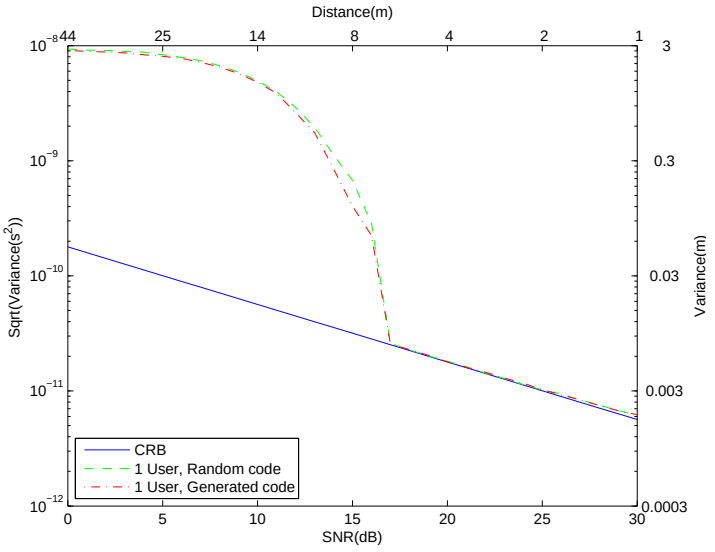


Figure 6.1: Variance for one user

one user. We can also appreciate that the variance when using the generated code is always a little bit lower than the one from the random code, not in a decisive way but it indicates a tendency. Moreover, we observe that beyond 14 dB the variance of the distance reaches acceptable values below 30 cm and around 30 dB the variance of the distance is in the order of mm. This figures gives us an idea of how the variance for multiple sources could be in the best case, which would happen when each user separately had the same performances as only one user transmitting.

6.4.2 Multiple users

TOA estimation variance for 4 users, $N_h = 6$ and $N_c = 3$

Now we will proceed with multiple sources TOA estimation. We simulate with 4 users, $N_u = 4$, and the rest of parameters the same as the previous simulation: $N_h = 6$ and $N_c = 3$. The number of experiments is again $N = 10000$ but since the number of users is 4 we work with 40000 samples per SNR value. The code has been backtracking generated and is the following

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 3 & 4 & 0 \\ 0 & 4 & 1 \end{bmatrix}$$

Results can be observed in figure 6.2. We see that the variance is clearly far from the CRB for both cases. The variance of the distance stays around

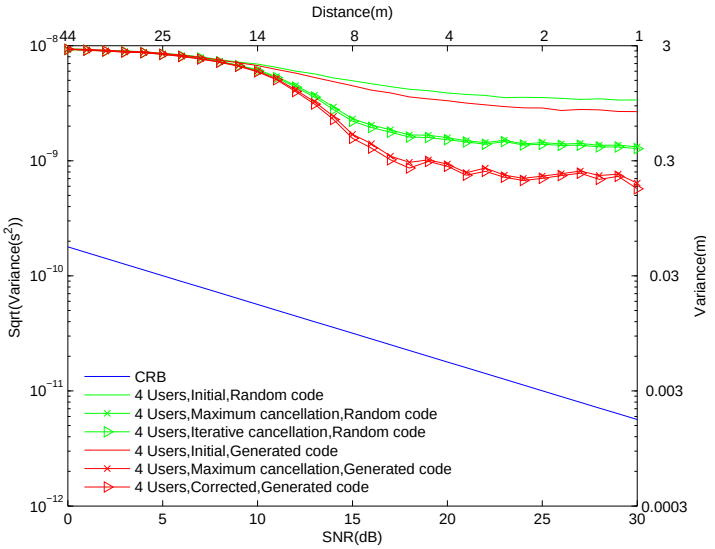


Figure 6.2: Variance of $N_u = 4$, $N_h = 6$, $N_c = 3$

3 m for low SNR and around 30 cm for high SNR. Although these values are outside an acceptable range, the simulation helps us to appreciate differences between the codes and the estimator. On the one hand, wee see a clear difference between random codes variance and generated codes beyond 14 dB. There is an improvement of around 30 cm for each estimator variance. On the other hand, the improvement of the estimators is notorious. The maximum cancellation estimator greatly improves the variance of both simulations, around 60 cm. The iterative cancellation estimator, however, slightly improves the estimation, but it is bellow the maximum cancellation curve for all points. In any case, it is interesting and positive that generated codes curves are always below random codes curves, that maximum cancellation curve is always below basic estimation curve and that iterative cancellation curve is also below maximum cancellation curve.

Concerning the high variances obtained, it seems that the problem lies in the system configuration since there is little improvement for higher SNR values. This is most likely because the signals are not sufficiently spread and there is a high chance of ambiguity. This can be solved by increasing N_h, N_c or both. Increasing N_h would simply increase the distance between pulses, which would happen in a similar way if the chip time T_c was increased. So, increasing T_c is a substitute solution to increasing N_h. Also, note that if we increase $N_c$we will most likely need to increase N_h as well, so the code satisfies the conditions seen in chapter 3.

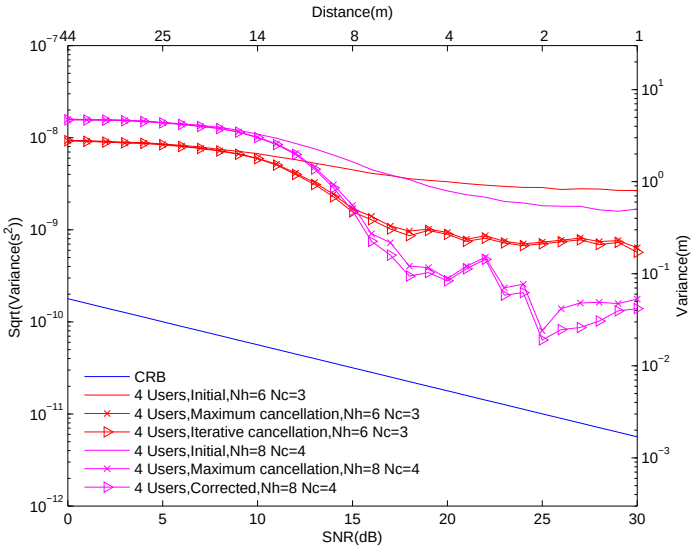


Figure 6.3: Variance of $N_u = 4$, $N_h = 8$, $N_c = 4$ vs $N_u = 4$, $N_h = 3$, $N_c = 3$

TOA estimation variance for 4 users, $N_h = 8$ and $N_c = 4$

Now, for the same number of users, we have increased N_c to 4. We have also had to increase N_h to 8 in order to achieve minimum auto- and cross-correlation THC. Again, the number of experiments N is 10000 and the number of samples per SNR value is 40000. In this case, we have only done one simulation for a backtracking generated code, which is

$$C = \begin{bmatrix} 4 & 3 & 0 & 0 \\ 0 & 1 & 7 & 1 \\ 4 & 0 & 7 & 2 \\ 0 & 2 & 5 & 3 \end{bmatrix}$$

The results are shown in figure 6.3 In the figure we compare the variance obtained to the backtracking generated code seen in previous section, that had $N_c = 3$. We see that for all estimators variance is now higher for low SNR values and lower. The reason why variance is higher for low SNR is that since we have increase N_c and N_h the period of the signal $T_p = N_h N_c T_c$ is now a $100\frac{8 \cdot 4}{6 \cdot 3} = 177.8\%$ longer, almost the double. Given this, for low SNR values, even if the error rate is lower now, errors are distributed in a longer space, and consequently the variance is higher. Anyway, for both cases the variance of the distance is higher than 1 m and thus, we are out of an acceptable range.

What interests us is that beyond 15 dB the new curves improve the variance substantially since the distance variance is now around 20 cm lower for high SNR values. Also, we can observe that curves are now more unstable, in particular the ones from the cancellation estimators. The explanation for this lies

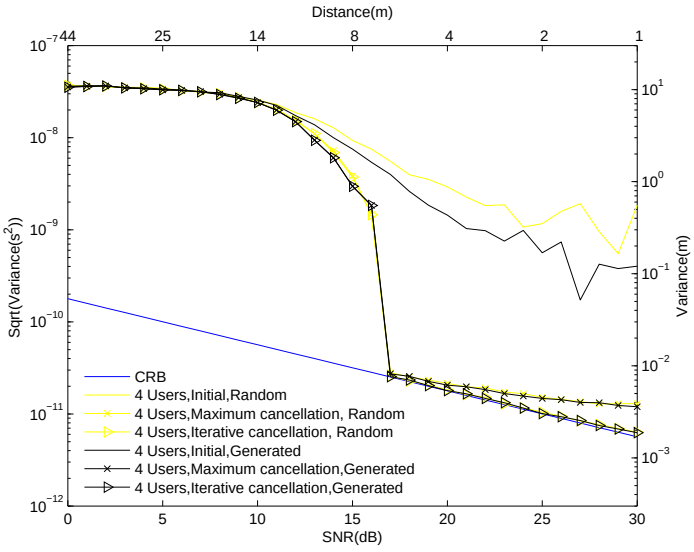


Figure 6.4: Variance of $N_u = 4$, $N_h = 15$, $N_c = 5$

again in the longer period. Since we have a longer signal period we have many more situations possibles given by the time asynchronism θ_n. Although this instability, we can appreciate the tendency of the curve and the improvement is evident.

TOA estimation variance for 4 users, $N_h = 15$ and $N_c = 5$

Since the last simulation was not achieving variances too far from the CRB we have incremented again N_c to 5. Again, we need to increment N_h, but this time we have used the fast generator since the backtracking takes too long when $N_c = 5$. An N_h equal to 15 has been obtained. In this case, besides the simulation with the fast generated code, another simulation has been carried out with randomly generated codes. Now, the period is a $100\frac{15 \cdot 5}{6 \cdot 3} = 416.6667\%$ longer than the first one. The generated code is

$$\mathbf{C} = \begin{bmatrix} 3 & 0 & 3 & 1 & 10 \\ 4 & 0 & 2 & 6 & 12 \\ 5 & 0 & 1 & 8 & 0 \\ 6 & 0 & 0 & 5 & 3 \end{bmatrix}$$

Results are in figure 6.4 The first thing that draws our attention is that variances of maximum cancellation and iterative cancellation estimators reach very close values to CRB beyond 17 dB. In fact, there is a transition zone between approximately 15 and 18 dB. Apparently, the variance of the basic estimator reaches a threshold, which if exceeded allows maximum cancellation

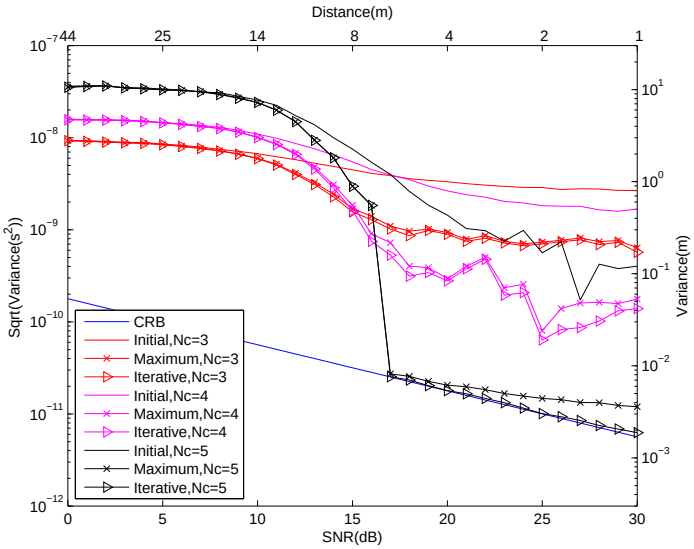


Figure 6.5: Variance for $N_c = 3$, $N_c = 4$ and $N_c = 5$

estimator to obtain variances similar to the CRB. Also, regarding the time-hopping code, we can appreciate differences in the initial estimation between the generated code and the random code, but then disappear when applying the cancellation. Furthermore, we can observe that the iterative cancellation estimation has a greater impact than in other simulations. Thus, we see that the maximum cancellation curve is distancing itself from the CRB curve as SNR increases, and how the iterative cancellation curve corrects it back to the CRB. We also appreciate some instability but now in the initial variance. This is due to the same reason from previous simulation, and now, since variance is successfully corrected, the figure does not show instability in the cancellation curves.

In figure 6.5 we plot the generated code variances together with the picture seen in 6.3, so we can compare the variances for $N_c = 3$, $N_c = 4$ and $N_c = 5$. We observe that when the initial variance of $N_c = 5$ significantly exceeds the other two initial variances around 17 dB, the maximum cancellation estimation suddenly reaches the CRB. So, when initial variances exceed below a certain threshold they can be corrected to the CRB using the maximum and iterative cancellation estimators. Also, as we have seen before, the variance for low SNR is higher for $N_c = 5$ due to its longer period T_p. As before, this does not affect us since we are outside the range of acceptable variance values.

At this point, we can figure out the sufficient condition the parameters of a THC must satisfy in order to reach the CRB. If the code length, N_c, is larger than the number of users, N_u, then the CRB is reachable for high SNR values.

To determine how good the results are, we can compare the variance of the

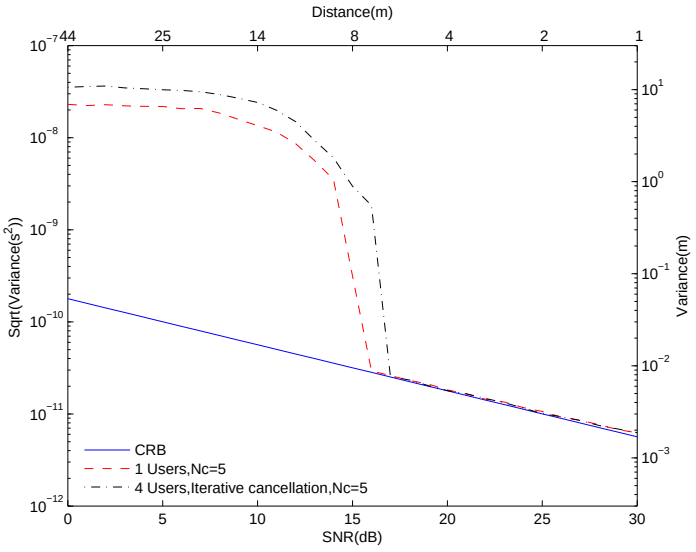


Figure 6.6: Variance of 1 user and 4 users, Nc=5

iterative cancellation estimation to the variance when only one user is transmitting. So, we have launched another simulation with $N_u = 1$, $N_h = 15$ and $N_c = 5$. We have used the code $\mathbf{c}_1 = \begin{bmatrix} 5 & 0 & 1 & 8 & 0 \end{bmatrix}$. The comparison can be seen in figure 6.6 We can see that until both curves reach the CRB, the estimation for 4 users has always higher variance than when only one user is transmitting. This does not particularly affect us for low SNR, but it does effect in the transition zone, where the estimation for one transmitting user reaches the CRB 1 or 2 dB before the multiple users estimation. Once both curves are in CRB zone, they are almost exact

6.4.3 TDOA simulation results

In order to realize the localization simulations we have taken the error matrix $\mathbf{E}$ from the last simulation with $N_c = 5$, since it is the one that has shown the best results. Specifically, we take the one from the iterative cancellation estimator with a minimum correlation code. Also, we have decided to take SNR values from 15 to 30 since its a range that let us observe the transition period and low variances zone. In the following sections we will show figures with a 2D scenario representing a room 6 m $\times$ 6 m, with the target to localize in the middle and green points representing the estimated position. Lastly, we will see a comparison graphic between the variance of the TOA estimations and the variance of the TDOA estimated position.

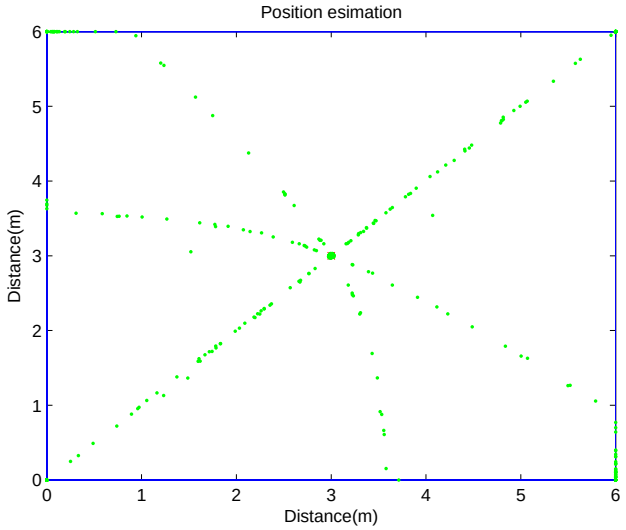


Figure 6.7: Position estimation for SNR=15

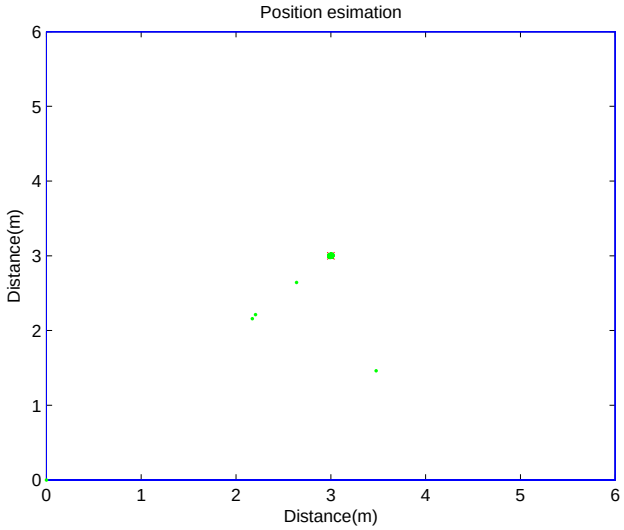
Position estimation for an SNR of 15

In figure 6.7 we see observe a 2D scheme with the estimated positions scenario corresponding to an SNR of 15 dB. With this SNR the TOA estimation variance is still high so we expect spread estimated positions. Since the original simulation had $N \cdot N_u$ samples now we have $M = {}^{N \cdot N_u}/_{4}{=}N$ estimated positions. This value is the same for the rest of simulations. We can see that there are three line-shaped zones, which divide the room in six zones, where the points usually are. This is due to the used position derivation technique. Since our technique is based in the use of three hyperbolas $d_{1,2}$, $d_{1,3}$ and $d_{1,4}$, the combination of errors produced in each one of these generates position estimations that fall in these zones. We have seven possibilities of error: error in $d_{1,2}$, in $d_{1,3}$, in $d_{1,4}$, in $d_{1,2}$and $d_{1,3}$, in $d_{1,3}$ and $d_{1,4}$, in $d_{1,2}$ and $d_{1,4}$, and finally in $d_{1,2}$, $d_{1,3}$ and $d_{1,4}$ The first three are the most common and lead to one of these zones. Also, note that most of the points attached to the wall are values that have been estimated outside the room. Since we assume the dimensions of the room are known, we put the point in the wall.

Also, we can observe that in the target position there is a small concentration of points that indicate us that at least some positions have been correctly estimated. The variance of this simulation is 0.3028 m^2 and its standard deviation is 0.5503 m.

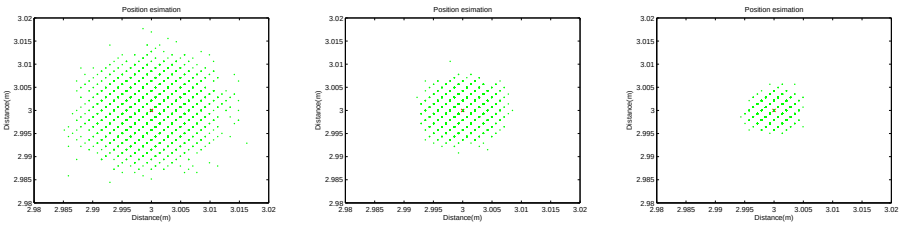
Position estimation for an SNR of 17 dB

We can see it in figure 6.8. It is clear the difference from the previous one since almost no spread is present and most of positions are correctly estimated.



o

Figure 6.8: Position estimation for SNR=15



(a) (b) (c)

Figure 6.9: Position estimation for (a) SNR=20 (b) SNR=25 (c) SNR=30

This is because at 17 dB the variance for the estimator has already reached the CRB and consequently the variance is significantly lower. The variance of this simulation is 0.0024 m^2 and its standard deviation is 0.0490 m$\simeq$ 5 cm.

SNR of 20, 25 and 25 dB

For this SNR values we have decided to plot a zoom of the zone, instead of the whole room, otherwise no difference would be noticed. We have plotted a square with a side of 4 cm around the target position (3, 3). We can see in figure 6.9 that, as expected, the zone where estimated positions belong decreases in size when SNR increases. Also we can see that the points are distributed on a grid form. This is due to the time resolution used in the simulations.

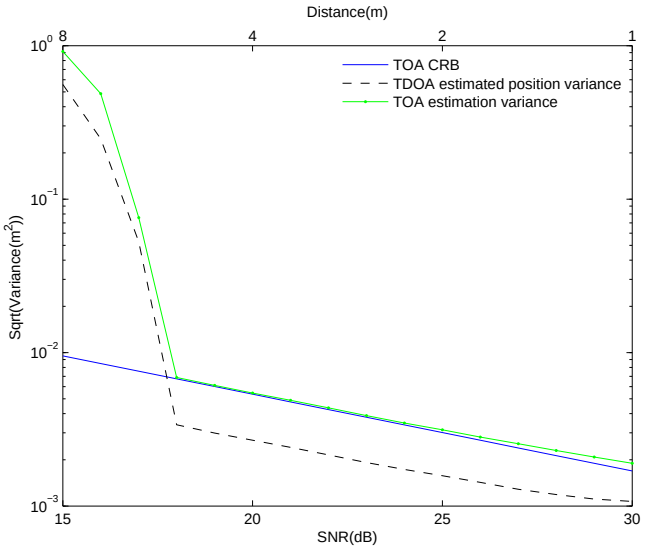


Figure 6.10: Variance of the TOA estimation and TDOA estimated position

Comparison with TOA estimation variance

In figure 6.10 we can see the comparison. It is clear that the variance of the distance when the object has been localized is lower than the simple TOA distance estimation error variance. In fact, we are comparing the estimation when we have 4 receivers to the estimation of 1 receiver. Various sources confirm that when the number of receives increases the variance of the estimation decreases [4]. Also, we can compute the ratio between the variance of the distance and the TOA CRB.

$$Ratio = \frac{\text{TDOA distance variance}}{\text{TOA CRB}}$$

SNR	15	16	17	18	19	20	21	22
Ratio	58.67	29.13	6.96	0.50	0.50	0.50	0.51	0.51

SNR	23	24	25	26	27	28	29	30
Ratio	0.51	0.51	0.52	0.53	0.54	0.56	0.59	0.63

As we can see, when TOA variance has reached CRB we obtain ratio values around 0.5. For higher SNR values the ratio increases but is still bellow one.

6.5 Conclusion

These simulations allow us to draw some conclusions.

We will start with the minimum auto- and cross-correlation codes. We have seen that in all simulations the use of optimum codes always improves the variance. However, we have only seen its impact in zones where the variance is far from the CRB. So, we haven't seen any situation where the use of optimum THC leads the variance to the CRB and randomly generated codes don't. Anyway, given that the variance is always lower for optimum, it is very likely that in certain situations the use of minimum correlation THC will let us reach the CRB while random won't. All this been said, the use of optimal THC is highly recommended.

Regarding the maximum and iterative cancellation estimators, they effectiveness is clear. Their use has shown improvement in the variance in all simulations. In fact, their use has been the only way to reach the CRB. Moreover, the iterative cancellation is particularly effective when the CRB has been reached since it corrects the maximum cancellation curve to the CRB curve.

Moreover, we have proved the feasibility of multiple sources positioning in UWB since the CRB has been reached. As a summary, we set the number of users at 4 and we started simulations with $Nc = 3$. Results were unsatisfactory so we decided to increase N_c to 4. Results were better but still out of acceptable ranges. Finally, with $N_c = 5$ the CRB was reached. This could make us think the best option is to always use a very high value of N_c, but this has an impact on positioning refreshing rates since larger periods are obtained. Its a compromise between variance and refreshing rate and if the lowest possible refreshing rate is required, the lowest N_c that reaches CRB is recommended. At this configuration, the performance of each user when 4 users are present is very similar to the performance when only one user is transmitting. Although we have only proved multiple sources ranging feasibility for 4 users, increasing N_c and N_h it is likely to allow more users to coexist.

About the TDOA based positioning, we have seen that its implementation is possible and satisfactory. We have observed that it improves the positioning variance for all SNR respect to the TOA distance error estimation. Also, for high SNR the standard deviation is in the order of mm.

Chapter 7

Conclusions and future work

7.1 Conclusions

This last chapter contains the final thoughts of this project. If we retake the objectives set at the beginning of the document, we can evaluate their degree of accomplishment.

The first objective of this thesis was to design optimum THC which improve the variance of the estimation. We have studied the impact of THC in the received signal and then found the necessary conditions a THC must have to allow minimum multiple-user interference. With this conditions, we have been able to design two different code generators. Finally, in the simulations we have seen that the variance obtained when using optimal THC is always lower than when we use randomly generated codes. However, we have also seen that using generated optimal codes doesn't mean that good results are going to be obtained. The code needs to be appropriate to the number of users. Furthermore, we have seen that backtracking generator is not always useful since for high values of the THC parameters too much computational time is required. In these cases, we can use the fast generator even though it does not provide minimum length periods. Being this said, we can conclude this first objective has been reached.

The second objective was to propose an estimator that could improve the variance. We have proposed two estimators, maximum cancellation and iterative cancellation, that we use concatenated. In simulations, these estimators have always shifted the initial estimation to a lower value and they have allowed us to reach the CRB. Therefore, this objective has been accomplished too.

Finally, the last objective that involved the previous objectives was to determine the feasibility of obtaining good variance values in UWB based positioning of multiple sources. In simulations we have proved it for four simultaneous users. We have seen that with a certain configuration, we can reach the CRB. Also, we have compared the variance of one user in a scenario of four transmitting sources

to an scenario where only one user transmits. The four user scenario reaches the same variance as one user scenario 1 or 2 dB higher due to the inevitable effect of multi-user interference. Also, simulations of TDOA positioning have been carried out. We have shown that for the used configuration we can achieve a standard deviation of the position error in the order of *mm* for high SNR values. So, we have accomplished the objective for at least four users. Although it is not possible to generalize, the possibility of using longer codes allows us to deduce that more users can coexist.

7.2 Future work

Given that this thesis represents only a first step in IR-UWB multiple sources positioning, several matters are left to introduce.

- Take into account multi-path propagation.
  In this thesis we have only considered single path propagation. It would be revealing that simulations were carried out with the impact of multi-path. Also, considering multi-path may require modifications in THC, such as the use of guard times.

- Take into account different attenuations
  In the simulations we have considered that all users are transmitting from the same position and power. We have stated that this represents the worse situation, but it would be interesting that some simulations were conducted with different attenuation so we could appreciate changes.

- Consider more than one period of the received signal
  In simulations we have used only one period due to software limitation, but still, it is possible to realize them in groups of periods. This would lead to changes in the variance, and, quite possibly, would lead to worse variance since the adjacent periods of the estimated period would affect the whole correlation. Also, this would affect the generation of optimal THC. Since we would have concatenated periods new distances between the pulses would appear and so, certain codes would not be optimum anymore. Also, the use guard times can be interesting here. Another aspect involved is that since we aren't modulating any data, we are using always the same repeated pulses and thus, it could be interesting to use a template signal of 2 or more periods in the receiver and check if there is any improvement.

- Improve the backtracking generator
  A computer technician could easily improve the backtracking algorithm so lower computation time would be required. This would be interesting since if minimum length codes are used lower refreshing rates can be achieved.

- Carry out a testbed for IR-UWB based multiple source positioning.
  Experimental measurements would provide a great contribution.

Bibliography

[1] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z. Win. Ranging with ultrawide bandwidth signals in multipath environments. 2009.

[2] J. George, Sh. Shaaban, and Kh. El Shennawy. Optimum time shift autocorrelation for time hopping uwb system using ppm technique. *Faculty of Engineering, Ain shams Univ., Egypt*, 2007).

[3] M. Ghavami, L. B. Michael, and R. Kohno. *Ultra Wideband Signals and Systems in Communication Engineering.* John Wiley & Sons, 2004.

[4] Achraf Mallat, Pierre Gérard, Maxime Drouguet, Farshad Keshmiri, Claude Oestges, Christophe Craeye, Denis Flandre, and Luc Vandendorpe. Testbed for ir-uwb based ranging and positioning: experimental performance and comparison to crlbs. *Communications and Remote Sensing Laboratory, Devices Integrated and electronic circuits Laboratory, and Electromagnetics Microwave Communication Laboratory. Ecole polytechnique de Louvain.*

[5] Achraf Mallat, J. Louveaux, and L. Vandendorpe. Uwb based positioning: Cramer rao bound for angle of arrival and comparison with time of arrival. *Communications and Remote Sensing Laboratory. Université catholique de Louvain.*

[6] Achraf Mallat, Claude Oestges, and Luc Vandendorpe. Crbs for uwb multipath channel estimation: Impact of the overlapping between the mpcs on mpc gain and toa estimation. *Communications and Remote Sensing Laboratory. Ecole polytechnique de Louvain.*

[7] Christophe J. Le Martret, Anne-Laure Deleuze, and Philippe Ciblat. Optimal time-hopping codes for multi-user interference mitigation in ultra-wide bandwidth impulse radio. *IEEE Transactions on wirelles communications, Vol. 5, No. 6*, JUNE 2006.

[8] Zafer Sahinoglu, Sinan Gezici, and Ismail Guvenc. *Ultra-wideband Positioning Systems.* Cambridge University Press, 2008.

[9] R. A. Scholtz. Multiple access with time-hopping impulse modulation. *Communication Sciences Institute, University of Southern California, Los Angeles, CA 90089-2565*, 1993.

[10] Xuemin (Sherman) Shen, Mohsen Guizani, Robert Caiming Qiu, and Tho Le-Ngoc. *Ultra-Wideband wireless communications and networks.* John Wiley & Sons, 2006.

[11] Kazimierz Siwiak and Debra McKeown. *Ultra-Wideband Radio Technology.* John Wiley & Sons, 2004.

[12] Jiangzhou Wang. *High-Speed Wireless Communications: Ultra-wideband, 3G Long-Term Evolution, and 4G Mobile Systems.* Cambridge University Press, 2008.

Chapter 8

Annexe

Backtracking generated codes

Nh=3, Nc=2

$$\mathbf{C} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ 0 & 1 \\ 0 & 2 \end{bmatrix}$$

Nh=4, Nc=2

$$\mathbf{C} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ 3 & 0 \\ 0 & 1 \\ 0 & 2 \\ 0 & 3 \end{bmatrix}$$

Nh=3, Nc=3

$$\mathbf{C} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Nh=4, Nc=3

$$\mathbf{C} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 3 & 2 \end{bmatrix}$$

Nh=5, Nc=3

$$\mathbf{C} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 4 & 1 \\ 0 & 3 & 2 \end{bmatrix}$$

Nh=6, Nc=3

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 3 & 4 & 0 \\ 0 & 4 & 1 \end{bmatrix}$$

Nh=7, Nc=3

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 6 & 2 & 0 \\ 0 & 5 & 0 \\ 4 & 6 & 0 \\ 0 & 4 & 1 \end{bmatrix}$$

Nh=8, Nc=3

$$\mathbf{C} = \begin{bmatrix} 2 & 0 & 0 \\ 6 & 1 & 0 \\ 0 & 4 & 0 \\ 0 & 7 & 1 \\ 0 & 5 & 2 \\ 0 & 1 & 4 \end{bmatrix}$$

Nh=5, Nc=4

$$\mathbf{C} = \begin{bmatrix} 3 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix}$$

Nh=6, Nc=4

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 4 & 2 & 5 & 0 \end{bmatrix}$$

Nh=7, Nc=4

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 2 & 0 & 4 & 1 \\ 0 & 3 & 5 & 1 \end{bmatrix}$$

Nh=8, Nc=4

$$\mathbf{C} = \begin{bmatrix} 4 & 3 & 0 & 0 \\ 0 & 1 & 7 & 1 \\ 4 & 0 & 7 & 2 \\ 0 & 2 & 5 & 3 \end{bmatrix}$$

Nh=6, Nc=5

$$\mathbf{C} = \begin{bmatrix} 3 & 0 & 5 & 1 & 0 \\ 0 & 2 & 5 & 5 & 3 \end{bmatrix}$$