INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Digital Communication System with Multi-carrier Modulation (OFDM) for Power Line Communication

Francisco Ramírez Jávega

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri

Presidente:	Prof. Mário S. dos Santos Nunes
Orientador:	Prof. José A. Beltran Gerald
Co-Orientador:	Prof. Gonçalo N. Gomes Tavares
Vogais:	Prof. Paulo A. Crisóstomo Lopes

October 2008

Resumen

Se propone el un diseño para un receptor OFDM. El esquema de sincronismo esta dividido en cuatro partes para estimar el sincronismo de los simboles, el offset frecuencial y el canal de transmision. Se implementa un esquema conjunto de ecualizacion y decodificacion basados en un algoritmo MMSE con informacion a priori y codigos LDPC, respectivamente. La asignacion de la carga de los subcanales sigue el criterio Water-filling.

I Abstract

A novel design for an OFDM receiver is proposed. The synchronization module follows Minn's synchronization scheme. Two novel timing estimators are proposed to implement the rough and fine timing estimation modules. The rough and fine frequency offset estimators are implemented using the Morelli and Mengali and Schmidt and Cox frequency offset estimators respectively. A basic Channel State Information estimation algorithm and a joint decoding and equalization method based on LDPC codes and MMSE algorithm with a priori information are introduced and implemented. The flat fading channel capacity is deduced from the basis of the mutual information between the transmitted and the received signals, being implemented the water-filling optimization algorithm on the basis of the obtained results. The performance of each module is simulated and compared to the state of the art. Finally, the overall performance of our design is evaluated by means of Monte Carlo simulation.

II Keywords.

OFDM, Water-filling, MMSE with a priori information, LDPC, SISO, Synchronization.

III Resumo

Nesta tese é proposta uma nova arquitectura para um receptor de OFDM. O módulo de sincronização baseia-se na técnica de Minn. Para realizar os módulos de ajuste grosso e ajuste fino de sincronismo do receptor são propostos dois novos estimadores de sincronismo. Os estimadores de ajuste grosso e ajuste fino do desvio de frequência são realizados com os estimadores de desvio de frequência de Morelli e Mengali, e Schmidt e Cox, respectivamente. São apresentados e implementados um algoritmo de estimação da resposta impulsional do canal e um método de realização conjunta da descodificação e equalização baseado nos códigos LDPC e algoritmo MMSE com informação *a priori*. É deduzida a capacidade do canal com desvanecimento não selectivo de frequência a partir da informação dos sinais emitidos e recebidos, e com base neste conhecimento é implementado o algoritmo de optimização de *water-filling*. O desempenho de cada módulo é comparado com os resultados correspondentes do estado da arte. Finalmente, o desempenho do sistema global é avaliado recorrendo a simulação de Monte Carlo.

IV Palavras-Chave

OFDM, Water-filling, MMSE with a priori information, LDPC, SISO, Synchronization.

V Table of Contents

VI List of Figures

VII List of Tables

VIII Table of Acronyms

Abbreviations and Acronyms	Meanings
A/D	Analogical to Digital Signal Converter
ACI	Adjacent Carrier Interference
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cockle, Jelinex and Ravin
BER	Bit Error Rate
CC	Convolution Code
CCAOS	Constant Complex Amplitude Optimal Sequences
CFO	Carrier Frequency Offset
CP	Circular Prefix
CSI	Channel State Information
dB	Decibel
ECC	Error Correcting Codes
FDM	Frequency-Division Multiplexing
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
ISI	Inter-Symbol Interferences
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratio
MAP	Maximum a Posteriori
Mbps	Megabits per Second
ML	Maximum Likelihood
MIMO	Multiple Input Multiple Output
MM	Morelli and Mengali
MMSE	Minimum Mean Square Error
OFDM	Orthogonal Frequency-Division Multiplexing
p.d.f	Probability Distribution Function
PC	Parity Check
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SC	Schmidl and Cox
SISO	Soft Input Soft Output
SISO	Single Input, Single Output
SNIR	Signal to Noise and Interference Ratio
SNR	Signal to Noise Ratio
SP	Set Partitioning
RV	Random Variable
UPA	Uniform Power Allocation
WER	Word Error Rate
WF	Water-Filling
ZF	Zero Forcer

1 Introduction

1.1 Motivation

One of the most important challenges in present day telecommunications is to design transmission systems which maximize transmitted data throughput with a limited bandwidth assigned, exploiting to the most the available communications infrastructure. Practical designs of such transmission systems need as well to be power efficient and simple enough to guarantee ease of maintenance and upgrade.

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier modulation scheme especially relevant for addressing the challenge presented above. This fact results from the high degree of flexibility that OFDM allows for adapting the transmitted data stream to the characteristics of the transmission channel.

1.2 The challenge of High Speed Transmission Systems

OFDM-based systems design is an active area of research, with many new algorithms and transmission techniques being proposed. When designing a transmission system, we must keep in mind that although all the components are performing a different task, they work as a whole. Thus, we must design them all keeping in mind the requirements of all the components connected to them. This process is known as *Cross-Layer Design*. There are different optimal solutions depending of the available information, so we have multiple approaches to solve the same design problem.

We can distinguish two different trends in these systems. The formed includes those systems that face up to the synchronization challenges, obtaining the initial information about the data structure conveyed into the channel. The latter includes those systems that improve error-correction capabilities. Both trends aima at decreasing the packet and bit error rate which reduce the number of retransmission needed to send the data from the original user to the end user as well as enable optimized techniques to allocate the energy available for the transmitted signal, adapting the transmitted data stream to the current state of the channel.

As a last remark, the main goal of the procedure is to optimize three design variables: the available energy at the transmitter, the bandwidth assigned to the transmission and the final device cost.

1.3 Research Goals

In this thesis the goal is aimed to improve a previous design of an OFDM receiver. It will be implemented and evaluated by means of Monte Carlo simulation the performance of the different modules designed for this purpose. I designed, implemented and simulated the performance of a synchronization module, a bit allocation module an adaptive modulation module, a MMSE with a priori information equalizer and a decoding stage based on LDPC codes.

A synchronization scheme based on Minn's scheme is introduced [2] to perform the different synchronization tasks. This synchronization scheme is composed of four different stages:

- A first timing estimation stage (rough timing estimation module).

- A first frequency offset estimation stage (rough frequency estimation module).
- The joint channel and fine timing estimation stage.
- A second frequency offset estimation stage (fine frequency estimation)

In the first module, a novel timing synchronization algorithm is proposed and its performance compared with the State of the Art algorithms by means of Monte Carlo simulations.

In the second module, Moreli and Mengali frequency estimation method is introduced [5].

In the channel estimation stage, a channel estimation method is introduced and a fine timing estimation method is proposed.

Finally, the Schmidt and Cox frequency offset estimation method is introduced for the fine estimation stage.

After above mentioned modules, next module performs the optimal bit allocation and the adaptive modulation. Here, a bit allocation algorithm, the Water-Filling algorithm, and we propose an adaptive modulation module with three different constellations with rate ½ are proposed.

A joint equalization and decoding stage is introduced based on MMSE algorithm with a priori information, the message passing algorithm and LDPC codes.

Finally, a frame scheme is proposed aimed to test the performance of the design receiver module.

1.4 Novel Contributions

In this thesis a novel joint synchronization scheme based on the previously proposed by Minn in [2] is proposed and analyzed. A new training symbol composed of several sequences without pattern is proposed enabling a frequency offset estimation with an acquisition range based on the number of sequences of the training symbol without introducing any lack of precision in the frequency offset estimation. A novel fine timing estimator is proposed, enabling a symbol-timing estimation only limited by the noise level.

1.5 Thesis Organization

This thesis is organized as follows: in Chapter 2, a OFDM explanation and a basic transmission system model is presented. In Chapter 3, the synchronization scheme and its modules are introduced and its performance simulated and compared with the state of the art methods. In Chapter 4, the Error Correcting module for SISO channels is introduced and tested allowing to perform the symbol and bit estimation processes. In Chapter 5, the Water-filling algorithm is introduced and the adaptive modulation module is proposed. In Chapter 6 MMSE equalizer with a priori information is introduced and the ideal performance for UPA and WF bit allocations strategies are given. In chapter 7, the simulation result of the system with the remaining estimation errors is given and analyzed. Finally, in Chapter 8, we do a briefing of the results obtained in this thesis and discuss possible improvements of our system and future lines of work.

2 Transmission System Model

In this chapter we will present the theoretical foundations of the present thesis. First we will offer a brief description of general transmission systems elements, presenting *Orthogonal Frequency-Division Multiplexing* (OFDM), the class of systems that this thesis has as object of study. After introducing these concepts, we will discuss the effects of different kinds of perturbations, channel, noise and synchronization challenges have on transmitted signals. Next we will show the different sub-systems present on both the transmitter and the receiver, in the context of OFDM transmission systems. Finally, we will comment the approaches we will do when designing the system.

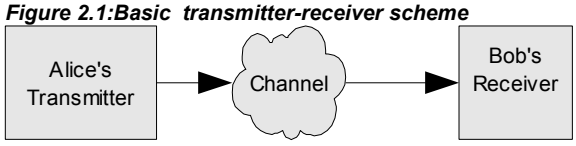
2.1 Overall View

2.1.1 The transmission system

In communications, a transmission system is a system that performs any transmission, emission or reception of any kind of signs, signals or information, through a wire, radio-electricity, by optical means or any electromagnetic wave.

Currently, communication systems are a concatenation of multiple devices allowing the transmission of any kind of information from one point to another. They are composed of three parts,namely: the *transmitter,* at the beginning of the line, is the device where the information is introduced, processed and sent to the *receiver,* at the end of line, which passes the information to the user once it has been processed. The medium to convey the information between the transmitter and the receiver is called *transmission channel*. An example of communication system is given in figure 2.1: we identify two users: Alice and Bob. Alice will use the transmitter to encode some information into a message and send it to Bob. Messages are, in turn, formed by symbols, the elemental units of information. The transmission channel is the medium messages travel through. Finally, the receiver is the entity supposed to receive incoming messages sent by the *transmitter* from the medium*,* decode them, and present them to Bob.

Figure 2.1:Basic transmitter-receiver scheme



In many practical communication systems, information data is transmitted over a channel with inter-symbol interference (ISI). At the transmitter, the data is often protected by the addition of a controlled amount of redundancy, a code. It is the task of the receiver to exploit both the structure of the transmitted symbols (as viewed at the output of the channel) and the structure of the code, so it may detect and decode the transmitted data sequence in order to extract the transmitted information data.

2.1.2 Orthogonal Frequency-Division Multiplexing

OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation

method. A large number of closely-spaced orthogonal sub-carriers are used to carry data. The data are divided into several parallel data streams (sub-channels). Each sub-channel is modulated with a conventional modulation scheme, where the data bits are mapped.

The frequency band assigned to the communication system is filled with a given number of sub-channels, each one carrying a different data stream, forming an OFDM symbol in frequency domain. In the following step, the Inverse Fast Fourier Transform (IFFT) is applied to the frequency domain OFDM symbol obtaining its equivalent in the time domain (the actual OFDM symbol).

Figure 2.2: Frame structure

In order to be transmitted through the channel, these symbols are grouped together in data frames. As is shown in the figure above, tt the beginning of the frame, a training symbol is added in order to enable the frame detection process at the receiver.

2.2 About the Transmission Channel, the Added Noise and Synchronization Challenges

2.2.1 The transmission channel

A channel can be characterized by finding some suitable mathematical model which describes faithfully enough the physical processes acting on transmitted signals. For instance, one can model static wireless communication channels by accounting for the reflection of electromagnetic waves when interacting with matter present in the medium. Very often, no analytical model can be readily found, so it is a common practice to rely on tabulated statistical data, from which statistical models of the phenomena under controlled conditions are generated.

The most basic channel model found in literature is depicted by the following equation:

$$r_x(n)=\sum_{k=0}^{L-1} h(k)*x(n-k) \quad (2.1)$$

where $r_x(n)$ stands for the received sample, $h(k)$ is a complex number whose magnitude accounts for the power of the incoming signal, and its phase, accounts for the delay suffered by the signal while crossing the medium, $x(n)$ is the transmitted sample and L the amount of received taps. This is the transference function of a typical model of the so called *fading* channel. The taps are generated according to a exponential power delay (the probability of high energy taps exponentially decreases), and the coefficients with the following equation:

$$h(k)=\exp(-k/\tau)\cdot\sqrt{(N(0,\sigma)+i\cdot N(0,\sigma))} \quad (2.2)$$

where the exponential represents the power delay, and N(0,sigma) a Gaussian probability distribution with zero mean and standard deviation σ. The tap generation process model can be further extended to account for *shadowing,* the signal's fading produced when a object, like a building, obstacles the LOS

between the receiver and the transmitter, the *Doppler Effect,* a frequency offset introduced because of the relative velocity between the receiver and the transmitter, or the Earth's curvature.

In this thesis we will be using a simple, but meaningful, channel model featuring the effect known as *flat fading* with no Doppler effect. The flat fading channel model consists in assuming that perturbations can be explained by a random process which remains *stationary* during the frame transmission time. This process can be thought of as selecting randomly a certain, pre-established, subset of possible sub-channels in the OFDM symbol and filtering them. For terrestrial multi-path channels, white noise is commonly used to simulate the background noise of the channel under study.

2.2.2 The additive white Gaussian noise. (AWGN)

Very much like channels, noise sources can be also modeled. The AWGN model accounts for perturbations caused by the thermal (black body) radiation of the surface of the Earth, objects on its surface and that of celestial objects. Besides that, it also encompasses the vibrations of the antennae atoms. This noise model is especially useful for modeling terrestrial channels.

The model itself is remarkably simple, given the plethora of phenomena it can account for, consisting in the linear addition of the signal and a constant spectrum signal, as depicted in the following equation:

$$r_x(n) = x(n) + w(n) \quad (2.3)$$

where the noise, $w(n)$, is Gaussian distributed with p.d.f.: $N(0,\sigma_n)$.

2.2.3 Synchronization challenges

The basic synchronization challenges occurs whenever the transmitter and receiver oscillators are not oscillating in the same frequency and during the frame start detection while detecting the first sample of OFDM symbol.

One should note that the receiver cannot know *a priori* when the transmitter will start sending a message. *Training symbols* with specified properties are added to the frame to help the receiver to overcome synchronization challenge problems by marking events like the start of the transmission and to estimate which is the first OFDM symbol's sample, the channel impulse response or the frequency offset.

2.3 The Transmitter

The physical layer of any transmission protocol is called transmitter. The transmitter is composed of several sub-systems with the following purpose: to add redundancy to transmitted signal, to structure data into frames, to allocate optimally the power available and to transmit the processed signal though the channel at the assigned band. A basic transmission system is depicted in figure 2.3.

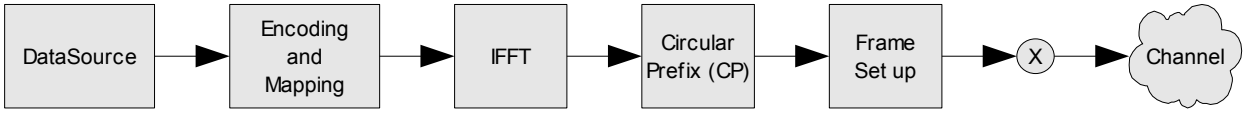


Figure 2.3: Transmitter scheme

The tasks assigned by each sub-system are described below.

2.3.1 Data Source

A memory buffer where input data is stored in binary form while waiting for being processed by the transmitter.

2.3.2 Encoding and mapping.

As shown in figure 2.3, the encoding and mapping process is performed in the first stage of the transmitter. Bit Error Rate (BER) improvements can be easily obtained protecting data with codes, with a negligible increase in computational complexity. Most practical communication systems also feature an interleaver after the encoder. Interleaving amounts to permute the symbols within a given block of data and tends to de-correlate error events introduced between neighboring symbols.

After adding redundancy, the obtained bits are grouped and mapped to a given constellation to be later transmitted through the transmission channel. The constellation mapping adds protection against additive white Gaussian noise effects (AWGN). Figure 2.4 shows a QPSK modulation, mapping 2 bits, $b_{0,}b_1 \in GF(2)$, following equation 2.4 criteria:

$$s_{(n)} = (-1)^{(b_0)} + (-1)^{(b_1)} i \quad (2.4)$$

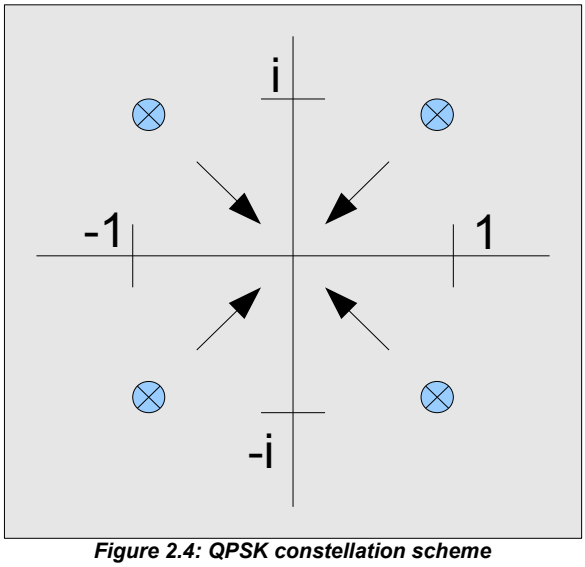


Figure 2.4: QPSK constellation scheme

AWGN is added to the phase and quadrature components of the mapped symbols, moving the received sampled symbol from the original position. The axis, in this case, are the decision frontiers; the symbol will be properly received at the receiver if the received sample doesn't change of quadrant. As the distance between the mapping position and the decision frontier between the symbols increases, symbol error rate decrease because the noise is Gaussian distributed. This distance is controlled by the amount of power assigned to the symbol and increases as the power assigned is increased.

2.3.3 Inverse Fast Fourier Transform

IFFT performs the symbol conversion from baseband frequency domain to baseband time domain in order to allow them to be transmitted though the channel. This module performs equation 2.5 mathematical

operation, being N the number of sub-channels in the OFDM symbol:

$$x(n)=1/\sqrt{(N)}\sum_{k=0}^{N-1}s(k)e^{(-j2\pi kn/N)} \quad (2.5)$$

2.3.4 Circular prefix module

Due to the impulse response of the transmission channel, *Inter symbolic Interference* (ISI) gets mixed with the received data. Generally, at the receiver, there is a module correcting this effect but avoiding ISI between different OFDM symbols is required in order to increase the performance of this module. A simple way of solving this is to append to the beginning of the symbol a copy of the last L samples of every OFDM symbol, the *Circular Prefix* (CP), in order to ensure that there is present only ISI coming from the current OFDM symbol.

2.3.5 Frame set up

In this section, the frame structure is set up by adding the training symbol with its own Circular Prefix to the previously generated OFDM symbols, buffering the signal samples before continuing with the transmission process.

2.3.6 Modulator

The function of this module is to convey the sampled complex data into the channel at the assigned frequency bandwidth. A signal conversion process is performed, from baseband equivalent to analytical signal, being up-sampled and sent to the transmission array of antennas. Generally, this module performs the following operation, being $b_s(n)=s(n)$ and f_{osc} the output oscillator frequency:

$$x_{(tx)}(t)=\Re\, b_x(n)\cdot e^{(j2\pi f_{(osc)}t)} \quad (2.6)$$

2.4 The Receiver

In this section, the structure of a receiver will be introduced. A basic scheme of a receiver is depicted in figure 3.1. These modules which are added to achieve improvements in throughput, will be deeply explained in the next chapters.

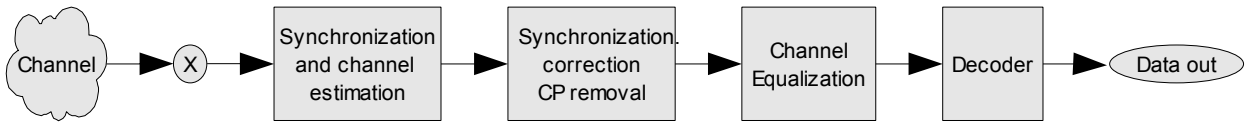


Figure 2.5: Receiver Scheme

2.4.1 Sampling process.

The receiver is continuously sampling the input signal received by its array of antennas. A thorough explanation of such modules will be offered in the following chapters. The signal is uniformly sampled at every T_s seconds,2.7. The signal samples are obtained and passed to the following system for further processing. This module performs 2.8 mathematical operation. After obtaining equivalent baseband of the

signal, $x(n)$ is filtered, removing the frequency aliasing arising from the previous sampling procedures

$$r_x(n)=r_x(n\cdot T_s) \quad (2.7)$$

$$x(n)=r_x(n)\cdot e^{(-j2\pi kn/N)} \quad (2.8)$$

2.4.2 Synchronization and channel estimation

In this stage, a search for the training symbol pattern is performed as soon as a pre-established level of energy is measured in the input signal. Detection methods are applied in order estimate the moment when information data starts to arrive. After that, the frequency offset is estimated applying frequency estimation algorithms over the training symbol.

After this process, the received samples of the training symbol are corrected with the estimations and then, Channel State Information, (CSI) is obtained. The CSI is an estimation of the impulse response of the transmission channel. Their phases and amplitudes might be estimated enabling further correcting processes.

2.4.3 Synchronization correction, CP removal

Once the synchronization errors are estimated, corrections are applied on the OFDM symbols, obtaining synchronized data with ISI, distortion and noise effects added. Further processing will try to estimate the original data bits from the received data streams

2.4.4 Channel equalizer and decoder

Methods that exploit the structure of the transmitted symbol constellation are referred to as equalization, whereas those that exploit the structure of the code are termed decoding. A number of important advances [8,9,10] have been made in the area of joint equalization and decoding, in which equalization and decoding methods exchange information in an iterative fashion until convergence is achieved.

2.4.5 Data out

After transmitting and receiving the signal, the transmitted information is received by the end user. The protocol stack tests data and once it is proven that it is not corrupted, it is passed to the services. If, after all the process, a data block is proven corrupted, packet retransmission is needed because the information cannot be extracted from the received signal.

2.5 Conclusions

In this chapter, we have introduced some generic concepts about OFDM based transmission systems and their components were introduced. Now we know a bit more about Alice's transmitter and Bob's receiver. The proposed transmitter will have all the modules referred in section 2.3 and the correspondent receiver will have those introduced in section 2.4. Despite of this, only the synchronization and the joint equalization and decoding stages, in the receiver, and the adaptive modulation stage, in the transmitter, will be deeply analyzed.

During the design process the following approaches are assumed:

- As a first approach, it was assumed that there aren't any kind of interferences produced by other devices transmitting in the assigned band.
- We will consider that the transmitter will know the channel impulse response as soon as the synchronization stage finish in the receiver and both, receiver and transmitter, will calculate the bit allocation algorithm at the same time. Thus, they both will know what kind modulations are carrying the data streams received through each sub-channel.
- The impulse response of our transmission channel is defined by (2.2).
- The only noise present at the receiver is additive white Gaussian noise.
- It was assumed that there isn't any ISI in the system, as well. Thus, the *Single In Single Out* (SISO) channel approach in the Chapter 5 matrices is assumed, so they will be diagonal matrices, generally. This assumption will hold for a perfect timing synchronization but, if not, it will still hold because usually the optimal sampling position is shifted to the right and, as the last taps of the channel usually have a low power level, the interference created by them will be, approximately, Gaussian distributed. So whenever there is an error event in the timing synchronization module, the noise variance will increase, hopefully, slightly. Thus, the performance of the receiver will decrease, hopefully again, a bit.

It will be shown, in the last chapter, that in the given scenario all these approaches hold in the proposed system.

3 Synchronization Errors and Channel State Information Estimation.

In this chapter Minn's synchronization scheme [2] is introduced. A novel synchronization scheme is proposed and its simulated performance compared with the present state of art estimators' performances. A symbol-timing estimator is proposed and its simulated performance with the present state of art estimator's performances. Three different frequency offset estimation methods are introduced and their performance evaluated by means of Monte Carlo simulation in AWGN and fading channels. Every section is divided in three parts: first, a brief overview of the methods, next a formal mathematical presentation and finally, a performance analysis based on the data obtained by the simulation.

3.1 Timing Estimation

Here, the results obtained from an extensive performance comparison between the proposed method, the classical Schmidl-Cox [3] and the newer Minn [2] methods are presented. In order to compare them, their performance on a given period, on two different dimensions, is measured . First, each method with a number of training Golay [16], Pseudo-Noise and Constant Complex Amplitude Optimal Sequences [6], (CCAOS), symbol sequences, in each case, with or without patterns is performed. Second, the methods behavior on both AWGN channels and fading channels with cyclic prefix is observed. Finally, their performance will be compared with the one of the proposed method.

3.1.1 Introduction

About timing estimation methods:

Schmidl and Cox (SC) method uses a training symbol with two identical halves in the time domain to find the start point of the symbol. However, due to the circular prefix (CP), there is an inherent uncertainty at the frame start point estimation, as shown later in figure 3.3. In order to avoid this effect, the timing estimation will be usually taken as the running average of the last N samples where N is the length of the CP for every received sample.

Minn's method avoids some of the inherent lack of precision of SC methods using training symbols with more than two halves and with different patterns.

The performances of both estimators will be studied.

About training symbols:

Pseudo-noise sequences are generated from the IFFT of an OFDM symbol filled with QPSK symbols at the odd frequencies and zero at the even frequencies.

For CCAOS sequences, the results of all the cross-correlations with all the possible shifts are zero and the only at zero shift the autocorrelation is non-zero, with its value being the length of the sequence, N. All

the sequence coefficients are complex numbers with the same magnitude. [6] [7]

Golay sequences are derived from Golay codes which are one of the three codes known to be perfect (Hamming codes, Golay codes and Shannon's repetition codes). Basically they are like CCAOS codes but they are formed by real numbers, which were chosen to be 1 and -1. [16]

3.1.2 Timing estimation algorithms

OFDM system model:

Using the IFFT to implement an OFDM symbol in time domain, the N symbol samples in baseband are given by:

$$x_n = 1/\sqrt{(N)} * \sum_{i=0}^{N-1} s_i \cdot e^{(-j \cdot 2 \cdot pi \cdot n \cdot i)}, n = 0,1,\ldots, N-1 \quad (3.1)$$

where N is the IFFT size, s_i is the complex modulated data on the $i^{(th)}$ sub-channel and $1/\sqrt{(N)}$ a normalization factor. In a CP OFDM system, a CP, that is the copy of the last part of the current symbol, is appended to each OFDM symbol. The signal is then passed though a D/A converter, up-converted to RF and transmitted, as explained in section 2.3.4. When the CP length is longer than the channel delay spread, the ISI caused by the multi-path fading is eliminated. At the receiver the signal is down-converted into a baseband signal $r(t)$, passed though an A/D converter and sampled at a rate $T_s = T/N$, where T is the duration of the OFDM symbol excluding the CP. The resulting samples may be expressed as follows

$$r_n = s_{(n-\varepsilon)} \cdot e^{(2 \cdot pi \cdot \Delta f_n / N + O_0)} + w_n \quad (3.2)$$

where **ε** is an integer representing the unknown arrival time of a symbol, Δf is the Carrier Frequency Offset (CFO), O_0 is the initial phase and w_n is the n^{th} sample of complex Gaussian noise with zero mean and variance σ_n^2 , $N(0,\sigma_n^2)$. For an AWGN channel $s_n = x_n$. For ISI or fading channels modeled as a tapped delay line s_n is given by

$$s_n = \sum_{k=0}^{l-1} h_k \cdot x_{(n-k)} \quad (3.3)$$

where l is the channel impulse response length and h_k the complex coefficient of each tap.

Schmidl and Cox method:

The first training symbol in SC method has two identical halves in time domain (see figure 3.1) and the timing metric is given by (3.4), (3.5) and (3.6), where $M = N/2$ samples,N is the IFFT size and $conj(r_{(d+m)})$ denotes the conjugate of r.

$$M_{SC}(d) = \left|(P_{SC}(d))^2\right| / R_{sc}^2(d) \quad (3.4)$$

$$P_{SC}(d)=\sum_{m=0}^{M-1} r^{*}_{(d+m)} * r_{(d+m+M)} \quad (3.5)$$

$$R_{SC}(d)=\sum_{m=0}^{N-1} r^{*}_{(d+m)} * r_{(d+m)} \quad (3.6)$$

$$M_{SC}(d_{opt})=\arg_{d}(max(M_{SC}(d))) \quad (3.7)$$

Evaluating P_S and R_{SC} the correlated transmitted power and the total received power, respectively, are obtained. R_{SC} in a continuous transmission, is usually constant and P_{SC} is, generally, a very small number, increasing only when the training sequence is present. Thus the receiver can know when the data transmission starts and estimate which the first OFDM sample is. The algorithm seeks to find d_{opt} evaluating $M_{SC}(d_{opt})$ according to equation (3.7). Due to the inherent plateau of this method, there is an uncertainty during the starting point decision process, being avoided to calculate the mean of the last N_g samples. Later, in figures (3.3.a) and (3.3.b) are shown the M_{SC} and the averaged M_{SC} metric profiles.

Ng	M	M

Figure 3.1: SC training symbol structure

Minn's method:

Minn's method uses a new training pattern [-a a -a -a] instead of the two identical halves with corresponding new timing metric given by (3.8), (3.9) and (3.10), where N is the IFFT size, L is the number of concatenated sequences, $M=N/L$ the number of samples of the sequence *a* and *pattern* the sign of each training segment (see figure 3.2). The new metric equation is given in (3.8), and is similar to the given for M_{SC} . The differences appear when calculating P_{Mn} and R_{Mn} . Equations (3.9) and (3.10) shown how these values are obtained. Schmidl 's method is a particular case of Minn's method, for $L=2$ .The generalized equation of P_{Mn} and R_{Mn} , the pattern between two consecutive segments is introduced through the parameter $b(k)$, being positive if the sign of the consecutive segments, is the same and negative otherwise.(see equation 3.11). The first OFDM symbol sample is selected using the same criteria that in SC method as shown in (3.12).

$$M_{Mn}(d)=\left|(P_{Mn}(d))^{2}\right| / R^{2}_{Mn}(d) \quad (3.8)$$

$$P_{Mn}(d)=\sum_{k=0}^{L-2}\sum_{m=0}^{M-1} b(k)\cdot r^{*}_{(d+m+k*M)}\cdot r_{(d+m+M*(k+1))} \quad (3.9)$$

$$R_{Mn}(d)=\sum_{k=0}^{L-1}\sum_{m=0}^{M-1} r^{*}_{(d+m+k*M)}\cdot r_{(d+m+k*m)} \quad (3.10)$$

$$b(k) = pattern(k) \cdot pattern(k+1) \quad (3.11)$$

$$M_{Mn}(d_{opt}) = \arg_{d}(max(M_{Mn}(d))) \quad (3.12)$$

Ng	[-] M	[+] M	[-] M	[-] M

Figure 3.2: Minn's training symbol structure

Due to the introduction of sub-sequences, sub-peaks appear into the M_{Mn} profile, as shown in figure (3.3.a). These sub-peaks could cause false detections and may be reduced by the introduction of patterns.

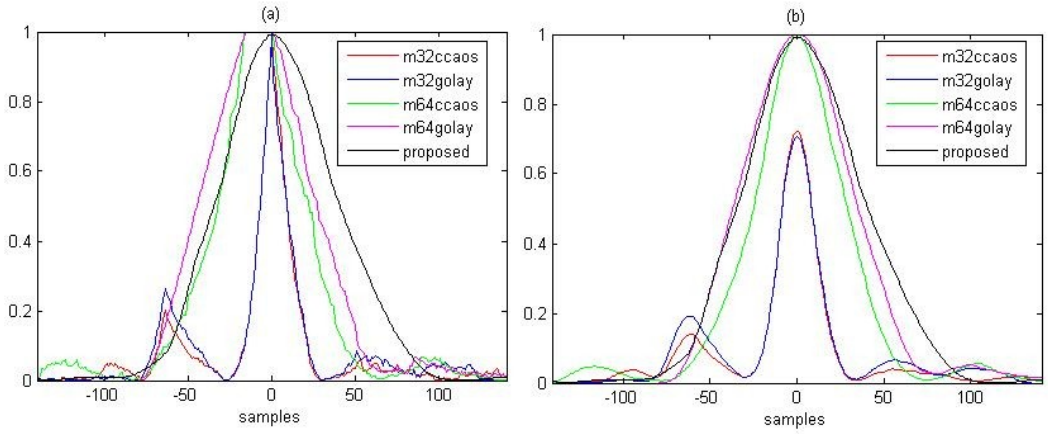


Figure 3.3: Perfect metrics. (a) Minn, SC and the prosed metrics. (b) Averaged Minn's and SC and proposed metrics

Figures 3.3.a and 3.3.b show the timing metrics of different methods under noise free and no distortion condition in a continuous transmission with QPSK modulation. All the curves use CP of 16 samples. We can see as Minn's method doesn't have the plateau of SC's method and SC method doesn't have any sub-peaks. On the other hand, Minn's method can be seen to feature the sharper peak. Figure 3.3.b shows the averaged methods. We can see as the plateau disappears in SC method and neither Minn's peak nor the sub-peaks are as sharp as in figure3.3.a.

The proposed method:

The proposed method uses a new training pattern [a a a a] with CCAOS sequences. It uses the Minn's metric given by (3.8), (3.9) and (3.10) but averaging the last N_g samples (see equation 3.13) obtaining a result similar to the SC metric. We see in figure 3.3b that there is not any difference a priori between the proposed and Minn's methods, but we will be able to exploit, the fact that the training symbol has four identical CCAOS sequences to improve the estimation accuracy.

$$M_{\text{prop}}(d) = \sum_{k=0}^{N_g-1} \left|(P_{Mn}(d-k))^2\right| / R_{Mn}^2(d-k) \quad (3.13)$$

The estimated start point is the maximum point of the metric.

Analysis of the effect of increasing the number of sequences in the training symbol:

In this section, we will perform a theoretical analysis of the methods to proof that the implemented metrics are correct and to analyze the effect of using different sequences to set up the training symbols. We consider an AWGN channel, SC and Minn metrics with CCAOS, GOLAY and pseudo-noise sequences.

First, let us define the Random Variable (RV):

$$M_d = M_{SC}(d_{opt}) - M_{SC}(d_{opt}+1) \quad (3.14)$$

Due to the central limit theorem Md may be considered Gaussian Distributed. Since $|(P_{SC}(d))|$, $R_{SC}(d)$ and $M_{SC}(d)$ have much larger means than their variances, $|(P_{SC}(d))^2|$ and $R_{SC}^2(d)$ can also be estimated by Gaussian RV with mean $L^2 \cdot (\sigma_s)^4$ and $L^2 \cdot (\sigma_s^2 + \sigma_s^2)^2$ respectively, also quite larger than their variances.

The mean of M_d are given by equations 3.16 and 3.17 for $M_{SC}(d)$ and $M_{Mn}(d)$, respectively:

$$E(M_d) = 4*(L-1)*(\sigma_s)^2/(L^2*(\sigma_s^2+\sigma_s^2)^2) \quad (3.15)$$

$$E(M_d) = 4*(L/2-1)*(\sigma_s)^2/((L/2)^2*(\sigma_s^2+\sigma_s^2)^2) \quad (3.16)$$

The variance are given by the same equation as in (3.15) and (3.16) swapping the factor L by L/2 for $M_{SC}(d)$ and L by L/4 for $M_{Mn}(d)$. Figures 3.4 to 3.9 show the comparison of the simulation and theoretic $M_d's$ mean and standard deviation using SC and Minn methods with pseudo-noise, CCAOS and Golay training sequences and CP in AWGN channels, where L=128, $\sigma_s^2 = 1/N$.

100,000 simulation runs were done for each SNR. Figures 3.6 and 3.9 show differences to figures 3.4, 3.5, 3.7 and 3.8 because the training symbols composed by CCAOS and Golay sequences have the same instant power but training symbols composed by Pseudo-noise sequences have the same average power.

Because of this, the power difference between the optimal sample and the next one can sometimes be higher than average power per sample.

As can be seen in the figures in the next page, the means observed for the Md RV in our simulations, when the training symbols are constant amplitude sequences, are almost the same than the theoretical result. The observed standard deviations are very close to the theoretic results; Standard deviations on observed Rvs, obtained with CCAOS sequences, match exactly the theoretical results, while the standard deviation for Golay sequences is slightly lower.

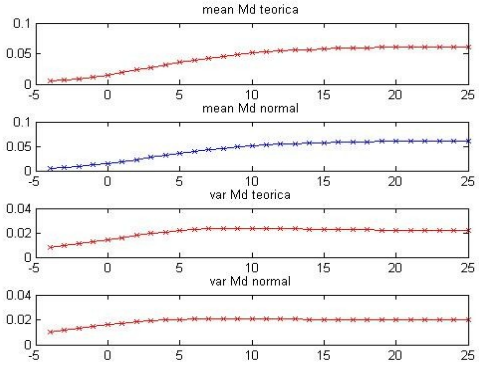


Figure 3.4 Mean and Standard deviation of Minn's Md. Golay training sequence

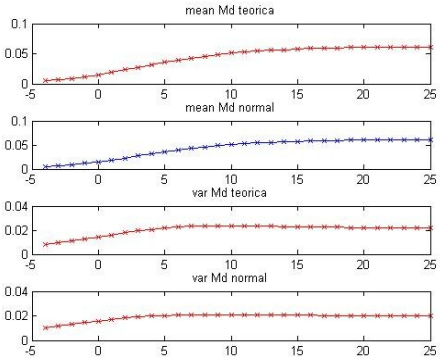


Figure 3.5 Mean and Standard deviation of Minn's Md. CCAOS training sequence

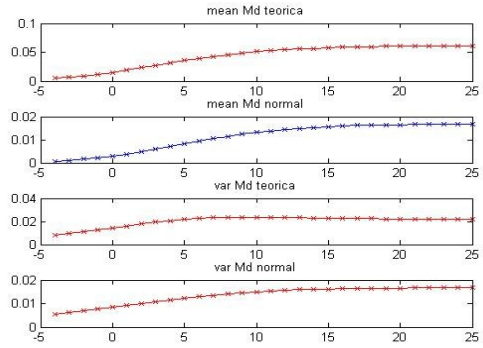


Figure 3.6 Mean and Standard deviation of Minn's Md. Pseudo-noise training sequence

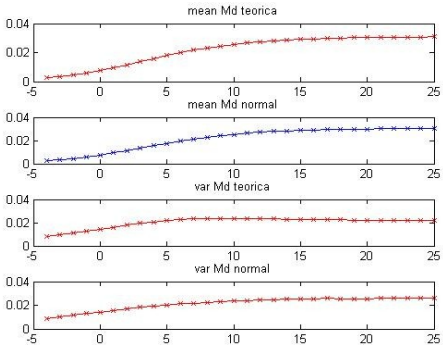


Figure 3.7 Mean and Standard deviation of SC's Md. CCAOS training sequence

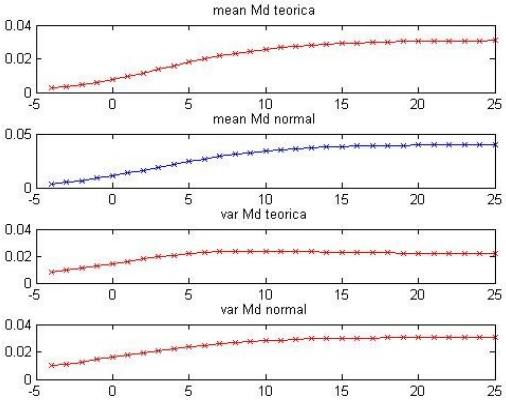


Figure 3.8 Mean and Standard deviation of SC's Md. Psedo-noise training sequence

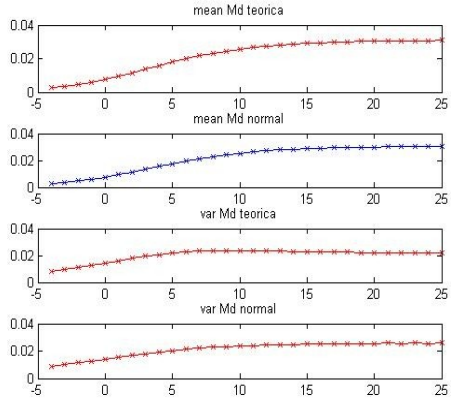


Figure 3.9 Mean and Standard deviation of SC's Md. Golay training sequence

3.1.3 Simulation results

The performance of the introduced and the proposed methods are evaluated by computer simulations in terms of the mean and standard deviation of the timing estimators under AWGN and fading, no ISI present, channel conditions.

The same parameters are used for all the simulations. The system uses 128 sub-carriers with a 128 point IFFT/FFT, the CP is $N_g = 16$. The fading channel has 15 paths with delays 1,2,3,...,15 given by (3.3).

Doppler Spread effect was not considered in the simulations. QPSK modulation is used and 100,000

simulations are run for each SNR. All simulations consider a transmission of a training symbol followed by 10 OFDM symbols.

Results in an AWGN channel:

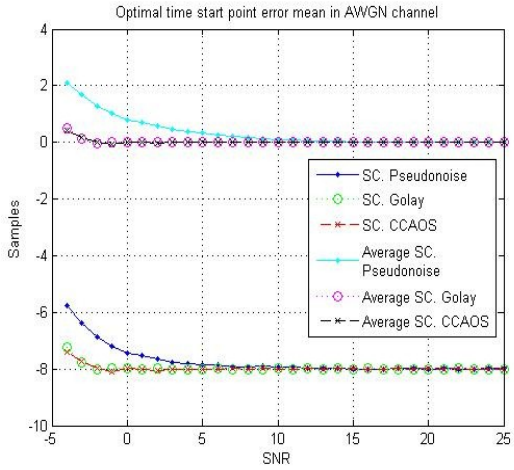


Figure 3.10: Optimal start point mean. AWGN-SC

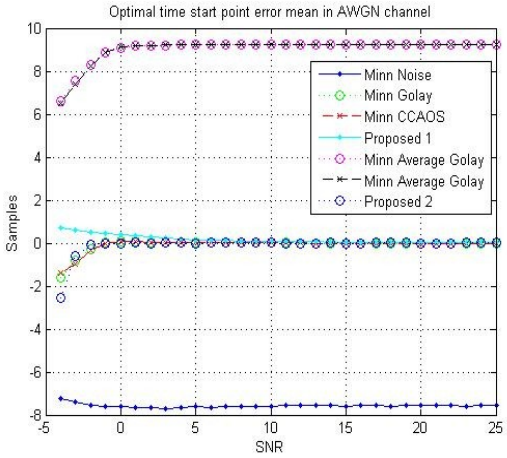


Figure 3.11: Optimal start point mean. AWGN-Minn and proposed

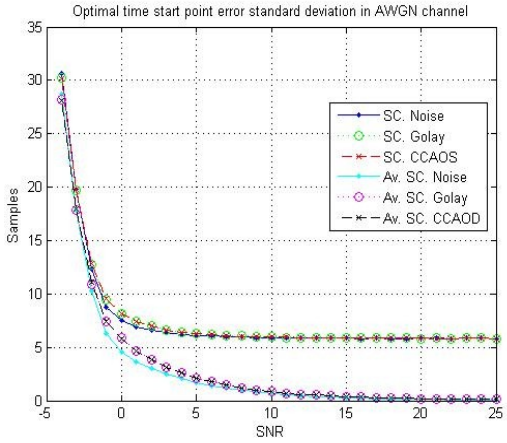


Figure 3.12: Optimal start point std. AWGN-SC

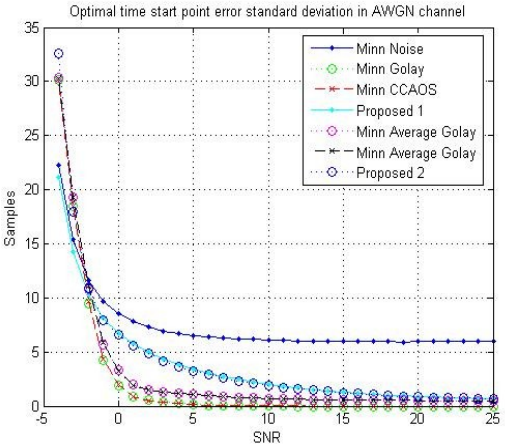


Figure 3.13Optimal start point std. AWGN-Minn and proposed

The figures above show the simulation results of SC, Minn and the proposed method. The plots named *proposed 1* and *proposed 2* are the results of simulating the proposed method with pseudo-noise and CCAOS sequences respectively.

The combination of Minn's method with constant amplitude training sequences is the one showing the best performance over the three studied methods in an AWGN channel. With a SNR higher than 2dB the correct start point is selected.

The mean for the proposed method is close to the mean of Minn's method but the standard deviation is much higher for a SNR lower than 20dB.

The Minn's averaged method shifts the mean of the starting point because of the averaging window and the standard deviation is higher than without averaging.

SC's starting point's mean is systematically shifted to the left $N_g/2$ samples (8 samples) and the standard deviation is around 6 samples because of the uncertainty due to the plateau present in the metric. Averaging the last N_g samples, we can see that the start point error mean is close to zero from -2dB on and the standard deviation decrease as the SNR is increased. The standard deviation is higher than in Minn's method for SNR lower than 20 dB. In this case, we can see that constant amplitude sequences lead to a better performance than pseudo-noise sequences, although the standard deviation is slightly lower than with constant amplitude sequences.

Results in a fading channel:

Now similar simulations were performed but in presence of a fading channel. The figures 3.14 to 3.17 show the obtained results.

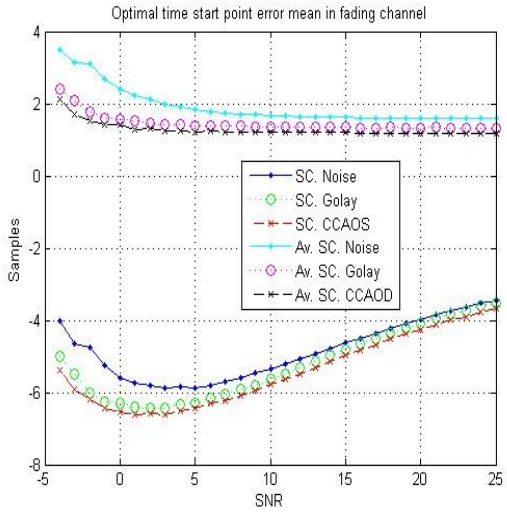


Figure 3.14: Optimal start point mean. fading-SC

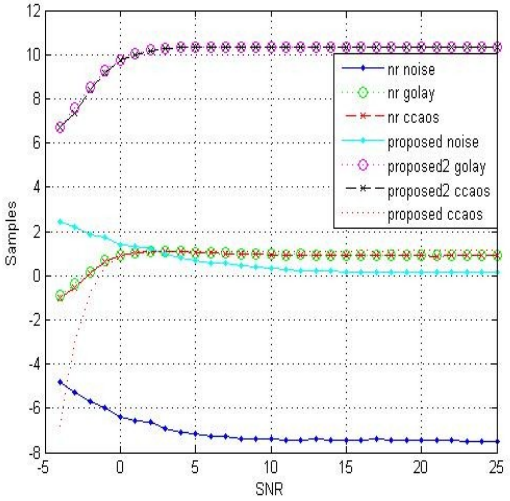


Figure 3.15: Optimal start point mean. Fading-channel. L=4

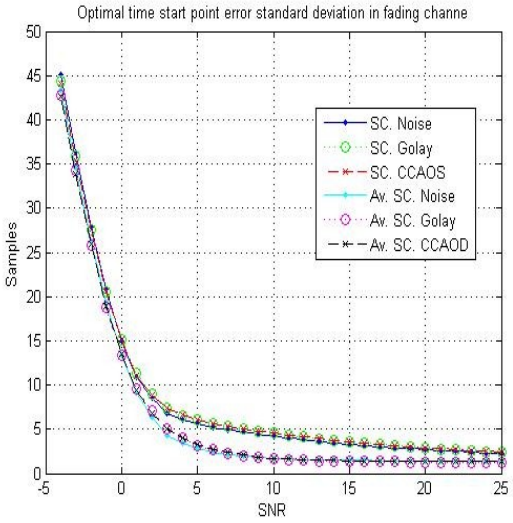


Figure 3.16: Optimal start point std. fading-SC

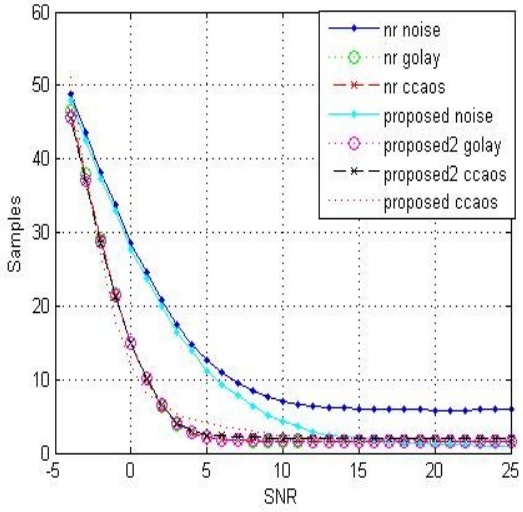


Figure 3.17: Optimal start point std. Fading-channel. L=4

As shown in figure 3.15, Minn's method and the proposed method with CCAOS sequences means are systematically shifted one sample to the right. This is because in fading channels the average Signal to Noise and Interference Ratio (SNIR), without any previous processing, is usually about 2 or 3 dB remaining constant respect the SNR because the discarded paths increase the interference level. We see that, in this case, the best performance estimator for low SNR are Minn's method with constant amplitude sequences because its mean and standard deviation is lower than for the other methods but, for higher SNR, the performance of the mean and the standard deviation of the proposed method with a pseudo-noise training sequence is better because the mean and the std deviations decrease more clearly than for the other sequences in this scenario. The proposed method with CCAOS sequences has the same mean than Min's method as shown in figure 3.15. Its standard deviation, compared with Minn's method estimator, is slightly lower for SNR lower than 2dB, slightly higher for SNR between 2 and 10 dB and the same for SNR higher than 10 dB (see figure 3.17). It is required to note that the plots labeled by *proposed2 CCAOS and Golay* are obtained by averaging the Minn's method with its patters. We observe that the mean is shifted to the right 10 samples and its standard deviation is almost the same than before averaging. These plots were simulated to see evolution of the statistics of the proposed methods when a pattern is present.

SC plain method's mean, when the SNR increases, is one sampled shifted to the rigth. This is caused by the multi-path effect that induces the collapse of the guard zone, so the selected point is shifted to the left $N_g - L_{ch}$ samples for high SNR. Averaging the SC metric estimation samples we can see how the mean decreases when the SNR is increased. More simulations (for SNR between 26 and 40 dB) are needed to assess the tendency of the estimator. The standard deviation of the SC selecting methods decreases as we increase the SNR but, again, more simulations are needed in order to assess the performance of the estimators for higher SNR.

3.2 Frequency Offset Estimation

Here, the results obtained from simulation of the Morelli-Mengali (MM) and SC frequency offset estimation methods are presented. First, an overview of the state-of-the-art in Frequency Offset Estimation methods will be performed, and then the results obtained with the MM and SC methods over simulated channels featuring AWGN and fading will be shown. Conclusions on the performance of these methods will close this section.

3.2.1 Introduction

The state of the art in methods for estimating frequency offset introduced by the lack of synchronization between oscillators in the receiver and transmitter, consist in the one proposed by Morelli and Mengali [5] and the work by Schmidl and Cox [3] and a last one derived from S.C. We will briefly describe these approaches.

3.2.2 Schmidl and Cox frequency estimation method.

The main difference between the two halves of the training symbol shown in figure 3.1 will be a phase difference given by equation (3.17), which can be estimated as per (3.18), being $P(d)$ given by equation

(3.5).

$$\phi = \pi T \quad (3.17)$$

$$\hat{\phi} = angle(P(d)) \quad (3.18)$$

near the best timing point. If $|\hat{\phi}|$ can be guaranteed to be less than π , then the offset estimate is given by (3.19) (called *Adjacent Carrier Interference* ACI), otherwise, the actual frequency offset would be given by (3.21) where *z* is an integer. By partially correcting the frequency offset, ACI can be avoided, and then the remaining offset can be found.

$$(\hat{\Delta f}) = \hat{\phi} / (\pi T) \quad (3.19)$$

The SC method estimates *z* by using a second training symbol, which is inefficient because the total throughput is reduced. In [17], another method to estimate *z* is introduced, without the need of transmitting two training symbols, being calculated as in (3.20), where s_{rec} and s_{tr} are the received and actual training symbol in frequency domain. The frequency offset is obtained as follows:

$$\hat{z} = \underset{z}{arg}\left(max\left(\sum_{k=0}^{N-1} s_{rec}(k) \cdot s_{tr}^{*}(k-z)\right)\right) \quad (3.20)$$

$$(\hat{\Delta f}) = \hat{\phi} / (\pi T) + (2 * \hat{z}) / T \quad (3.21)$$

3.2.3 Morelli and Mengali frequency offset estimation method

M and M. estimator exploits the correlation of the received sample as shown in (3.22). Here N/L is the length of each section of the training symbol and *H* is a design parameter whose optimal value is $L/2$ [5]. Based on $R(m)$ the phases, $\hat{\phi}$,are computed according to (3.23). It is shown in [5] that the $\hat{\phi}$ may be expressed as in (3.24) where γ denotes a channel induced error. All these results holds for $SNR \gg 1$ since the effects introduced by the channel will, with high probability, be smaller when compared with the transmitted samples, allowing this to neglect these terms. Thus for $\nu \le L/2$ equation (3.24) holds.

$$R(m) = \frac{1}{(N - mM)} \sum_{k=mM}^{N-1} r(k) r^{*}(k - mM) \quad 0 \le m \le H \quad (3.22)$$

$$\phi(m) = [angle\, R(m) - angle\, R(m-1)]_{2\pi} \quad 1 \le m \le H \quad (3.23)$$

$$\phi(m) = 2\pi\nu / L + \gamma(m) - \gamma(m-1) \quad (3.24)$$

The estimated of the normalized CFO, $\hat{\nu}$,is obtained from (3.25) where $w(m)$ is a weighting vector given by (3.26).

$$\hat{\nu} = \frac{L}{(2\pi)} \sum_{m=1}^{H} w(m) \phi(m) \quad (3.25)$$

$$w(m)=3\frac{(L-m)\cdot(L-m+1)-H\cdot(L-H)}{H\cdot(4\mathrm{H}^2-6\mathrm{LH}+3\mathrm{L}^2-1)}\quad 1\le m\le H \quad (3.26)$$

3.2.4 Simulation Results

The performance of the proposed methods is evaluated by computer simulations in terms of the mean and standard deviation of the obtained data under AWGN and fading channel (without ISI).

The same parameters are used for all the simulations. The system uses 128 sub-carriers with a 128 point IFFT/FFT and the CP is Ng=16. The ISI channel has 15 paths with delays 1,2,3,...,15 given by equation (3.3).

It was not considered the Doppler spreading effect in the simulations. It was used QPSK modulation and 100,000 simulations are run for each SNR level. All simulations consider a transmission of a training symbol followed by ten OFDM symbols.

In figure 3.18 the mean and the standard deviation (up and down respectively) of the estimator are shown for fading and AWGN channels (left and right respectively). The label L=4 and M=32, label L=2, M=64, represent the Minn training symbol and the S.C training symbol respectively. The label *original V*, represents the normalized frequency offset introduced into the system (V=0.3641).

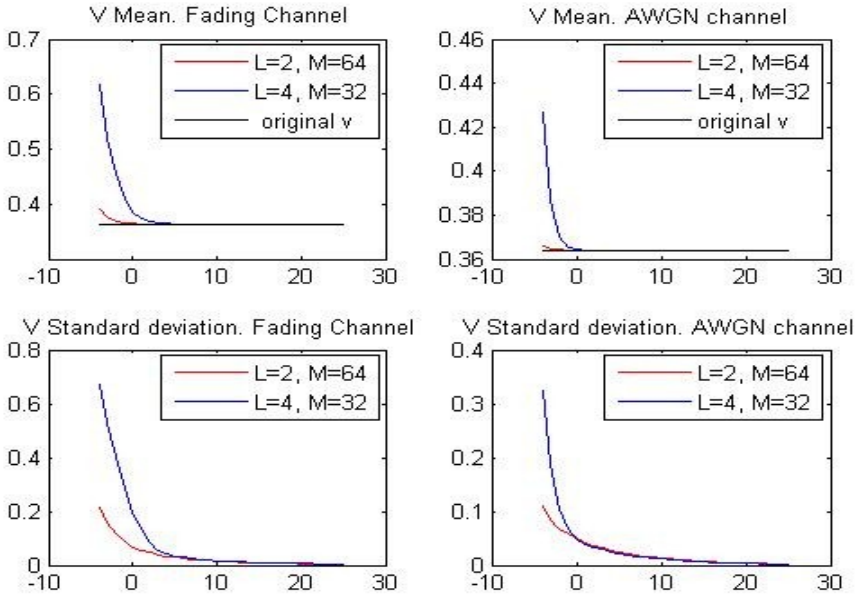


Figure 3.18: Normalized frequency offset. Comparative between MM and SC training training symbol's structure

The estimator is unbiased and consistent because the mean and the standard deviation decreases to as the SNR increases. In figure 3.18 is shown that, for low SNR and in both channels, the performance of Minn's training symbol is lower than Schmidl's training symbol but for SNR higher than 0dB in AWGN channels and for SNR higher than 6 dB in fading channels, both have similar performance. This is caused because the variance of the estimator increases as the number of samples in each segment of the training sequence decrease, as shown in equation (3.27) from [5].

$$var(\hat{\nu}) = \frac{2 \cdot SNR^{-1}}{\pi^2 \cdot N} \quad (3.27)$$

A priori, the performance of the SC training symbols is better but we should keep in mind that the estimation range is $|\nu| \leq L/2$, ν the sub-carrier spacing. Thus there is a trade off between the estimation range and the variance of the estimator.

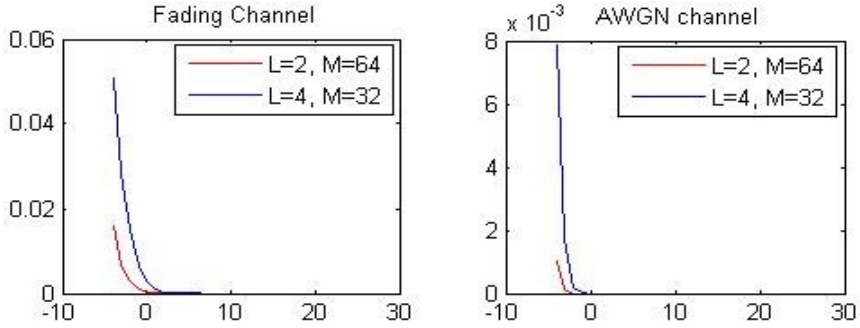


Figure 3.19: Probability of estimation out of the estimation range for Minn and SC training symbol structure

As a least remark, the probability of estimating frequency offset outside the estimation range is shown in the figure above.

3.3 Joint Channel State Information and Fine Timing Estimation

In the first section, a little introduction to channel estimation is given. In the second section, the process to obtain the first channel impulse response, also known as Channel State Information, in the proposed system is exposed. Next, it is proposed a new method to estimate the remaining timing error and obtaining a more reliable CSI estimation. Finally, some aspects about the channel estimation obtained are commented.

3.3.1 Introduction

Obtaining very precise Channel State Information is required as the channel equalization, the decoding and bit allocation processes are based on this estimation. Again, the knowledge of the training symbol during the synchronization process is exploited in order to perform this process. Basically, after performing the time synchronization and frequency offset correction processes, the circular prefix is removed and a FFT is applied to the corrected training symbol's samples. If the frequency offset is completely corrected, the output symbol from the FFT module will be composed by the impulse response of the channel, the frequency domain components of the training symbol and noise. Thus, removing the contributions of the training symbols, obtaining a noisy estimation of the transmission channel.

3.3.2 Channel estimation process

Theoretically, after a perfect timing synchronization and frequency offset correction, the equation 1 expression is obtained, where $x(n)$ are the training symbol's samples in time domain. Thus, after applying the Fourier transform to the training symbol, (3.28) is obtained, where s_n are the frequency domain components of the OFDM training symbol, σ_n^2 is the Gaussian noise variance, H_n is the

impulse response of each the sub-channel in frequency domain.

$$Y_n = H_n \cdot s_n + \sigma_n^2 \quad (3.28)$$

$$normalized\ s_n = \frac{s_n^*}{|s|^2} \quad (3.29)$$

$$\hat{\underline{H}}^H = \underline{H}^H + \sigma_n^2 \frac{\underline{s}^H}{\underline{s}^H \cdot \underline{s}} \quad (3.30)$$

Multiplying (3.28) by (3.29) a noisy channel estimation is obtained in frequency domain (3.30).This channel estimation, for SNR >>1 will be very close to H .

In figure 3.20 we can see the squared normalized error mean and variance of the channel estimations for 5000 simulation runs and Minn's training symbol's structure. The simulation result show that the performance of the channel estimator is bound to the SNR, the estimated error decreases exponentially when the SNR level increases for SNR lower than 17 dB. For higher SNR levels the increase of performance is smaller.

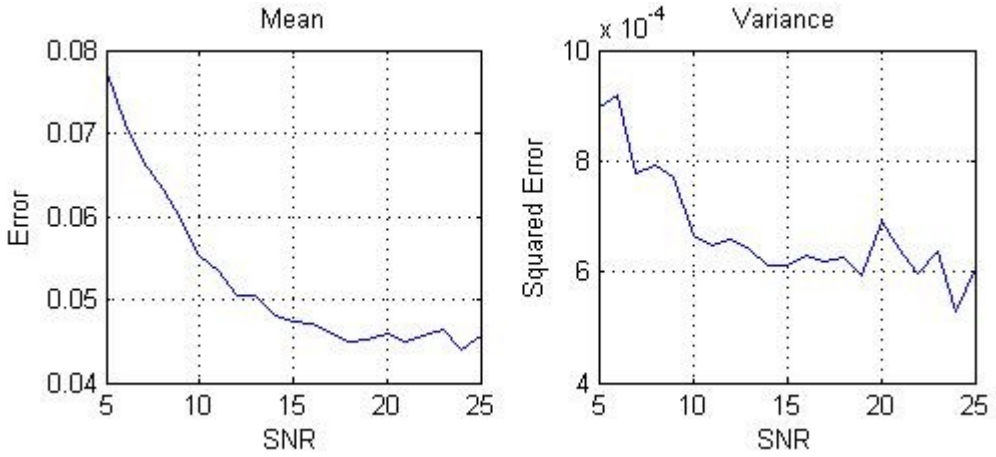


Figure 3.20: Channel normalized squared error mean and variance

3.3.3 Joint fine timing and channel estimation.

The previously obtained channel estimation is not enough precise and we must try to reduce the mean square error of the estimation must be reduced. Thus the following process is aimed at reducing this error by exploiting the fact that, whenever we have a timing estimation error the first samples of the real channel appear at the end of the channel estimation. This happens because the FFT output will detect that there is ISI and will shift circularly the channel estimation to the left if the selected optimal samples is in the CP, and to the right if a sample of the OFDM symbol is selected.

Now it is time to exploit some of the properties of the training symbol that are proposed. Reminding that the channel is composed of 15 taps we can do as follows:

- First, the training symbol is composed of four identical sequences. So four FFT, one of each sequence, are performed obtaining four different estimations of the channel without ISI as in (3.31)

where $\underline{\underline{C}}$ is a frequency channel estimation in each row, $\underline{S}$ the FFT of size M of the training sequence, and $\underline{r}_x$ the received samples once the frequency offset is corrected. This is the first difference between Minn's and SC's methods. Minn's training symbol is composed of four identical sequences with different signs (-1, 1, -1, -1) and the CP. So we can obtain only two ISI free estimations of the channel, the first and the last sequence estimations. The others will have ISI from the previous sequences as they have different signs. Obviously, we can obtain only two ISI free different channel estimation of SC's training symbol as it is composed by only two sequences.

$$\underline{\underline{C}} = \sum_{l=0}^{L-1} FFT(\underline{r}_x) \cdot \underline{S}^* \quad (3.31)$$

- Second, the FFT is applied to each of the obtained channel estimations, as in (3.32), and next a cross-correlation is performed between each one of the sequences in order to reduce the noisy effect on the sequences (3.33); we must remind that the AWGN is uncorrelated so the correlation of two different samples of noise is equal to zero (or very small, generally). Thus six, $Q = L \cdot (L-1)/2$,correlations for $L = 4$ are obtained. The point of performing this operation is that the average noise level will be the same, more or less, for all the training symbol, but it won't be the same if we reduce the number of samples to analyze. So, some of our estimations will have less noise contribution than others and we can, now, discard the ones with high noise level, because we can identify the sequences with high noise level as the noise is equally distributed in all the M samples of our estimation and we know that we can have taps only around the first 15 samples (at the end or after the 15th estimated tap). This is the main difference between the proposed training symbol and Minn's training symbol as, due to the ISI, it is possible that almost all the correlations performed over the different channel estimation have high values at the right of our channel estimation.

$$\underline{c} = FFT(\underline{\underline{C}}) \quad (3.32)$$

$$\underline{\underline{R}}_{ch} = [cor(\underline{c}_1 \cdot \underline{c}_2) \quad cor(\underline{c}_1 \cdot \underline{c}_3) \quad ... \quad cor(\underline{c}_2 \cdot \underline{c}_4) \quad cor(\underline{c}_3 \cdot \underline{c}_4)] \quad (3.33)$$

$$cor(\underline{c}_i \cdot \underline{c}_j) = corr(\underline{h} \cdot conj(\underline{h})) + cor(\underline{h} \cdot \underline{w}_j) + cor(conj(\underline{h}) \cdot \underline{w}_i) + cor(conj(\underline{w}_i) \cdot conj(\underline{w}_j)) \quad (3.34)$$

- After discarding the noisy correlations, he mean over the remaining ones is calculated, reducing the noise effects. After that, we check the positions with possible taps are checked, first at the end of our estimation and later after the L position. Two things must be noted:

  ➔ Generally, the taps at the end of the channel estimation will have a higher energy level that the ones after the 15th tap because the channel model is exponentially distributed. Thus, it will be easier to detect the shift if the original optimal sample position is shifted to the left than when shifted to the right, as the first taps shifted to the left will be, with high probability, higher than the ones at the end, shifted to the right. Because of this, the CCAOS sequence was selected instead of the pseudo-noise sequence as the optimal sample position estimated by the proposed method with CCAOS sequences is, generally, shifted to the right less times than when

the training symbol is composed by pseudo-noise sequence. Thus it will be easier, generally, to detect these shifts.

➔ The minimum level of energy that will be estimated as a tap, must be set keeping in mind that if it is high, some taps won't be detected which means that these taps will be seen for the system as ISI. If a low energy level will be set, the probability of introducing noise into the channel estimation is higher, which leads to a higher probability of distorting the channel estimation. Thus, this level selection must be done carefully keeping in mind both things.

- After setting our energy levels, we select the amount of taps that we are going to shift the optimal sample position that we already have. Then, we repeat the first step at the new optimal sample point. Next a IFFT of each channel sample and the autocorrelation of each sample in time domain. After that, we discard the samples with high energy levels in all the taps (as in the second step).Finally, the mean of each tap of the remaining channel estimations is computed, discarding all the taps in the positions higher than 15. Then result is the new channel estimation.

3.3.4 Simulation Results.

Comparing figure 3.20 and 3.21 we can see that after the correction process, the normalized squared error is reduced about one order of magnitude comparing.

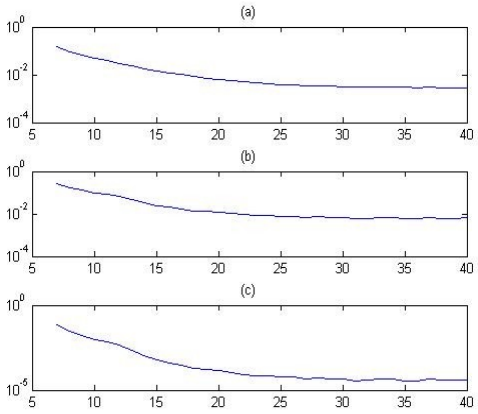


Figure 3.21:Normalized square error (a) mean. (b) Standard deviation (c) Variance

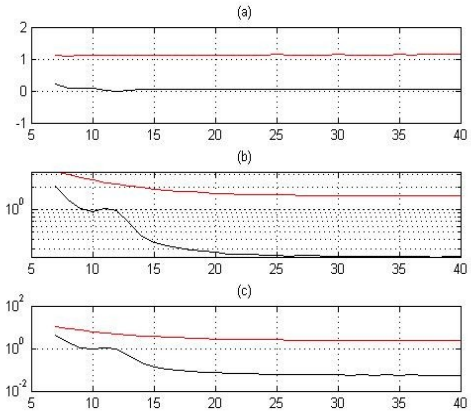


Figure 3.22: Optimal sample error distance. (a) Mean. (b) Standard deviation. (c) Variance. Red line: rough estimation. Blue line: fine estimation.

In figure 3.22 we can see the comparison between the rough and the fine estimation results. After the fine estimation process, the average error distance between the optimal frame start point and the obtained is reduced two order of magnitude and the variance, another two orders of magnitude.

3.4 The Fine Frequency Offset Estimator.

3.4.1 The chosen estimator.

The last step in the proposed synchronization process is calculating the remaining frequency offset. It will be obtained using the SC frequency offset estimator introduced in the section 3.2.2. Another reason for choosing the proposed training symbol is that we can apply the SC frequency offset estimation after

estimating the offset with the Morelli and Mengali estimator. The point, again, is that there isn't ISI between the sequences that constitute the training symbol. Due to it, any lack of precision is introduced when estimating it. So the remaining frequency offset estimation can be obtained as in (3.18), at the optimal sample selected after the fine timing estimation process.

3.4.2 .Simulation Results

In figure 3.23 we can see the statistics of the remaining frequency offset. This remaining error bounds the maximum length of the transmitted frame. The effect introduced by the frequency offset can be seen as a progressive shift of the transmitted constellation symbols. Due to this, for long frames or high frequency offset errors, the received constellation symbols won't be in their positions and the equalizer won't be able to correct that.

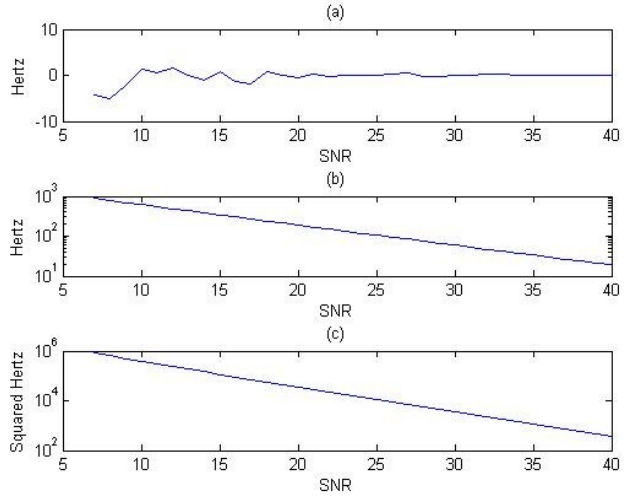


Figure 3.23: (a) Mean of the remaining frequency offset. (b) Standard deviation of the remaining frequency offset. (c) Variance of the remaining frequency offset

3.5 Conclusions

In this chapter, some timing and frequency offset synchronization methods were exposed and their performance simulated over AWGN and fading channels. A method to estimate the channel impulse response was proposed. First, the performances of each of the estimators proposed and introduced are resumed. After, some points that must be kept in mind when designing a receiver were addressed.

It was shown that SC timing estimator method is robust because the performance increases as the SNR is increased, being the starting point shifted. Minn's method works well in AWGN channels, but the mean value might exhibit shifts from the exact start point due to the multi-path effect and also implies a slightly higher computational load. The proposed method has a performance slightly inferior than the Minn's method in a fading channel, as its standard deviation is slightly higher for SNR between 3 and 10 dB.

It was observed that the performance of the different timing estimators is bound to the chosen sequence that constitute the training sequence. Generally, for low SNR levels, the variance of the training sequences formed by pseudo-noise sequences is lower than the ones formed by constant amplitude sequences but the mean is more shifted from the optimal sample point. Because of this, generally, the

sampled estimated by the constant amplitude sequences will be closer to the optimal sample. As the SNR increases, the performance of the pseudo-noise sequence increases because the distance between the optimal and the selected sample decreases, for both fading and AWGN channels. In this case, its performance is better than the constant amplitude sequences.

As a last remark about the timing estimators, it was seen that there is a lower bound for the timing estimation distance in fading channels. This is because in fading channels the average *signal to noise and interference ratio*, SNIR, without any previous processing, is usually about 2 or 3 dB remaining constant although that the SNR increases because the discarded paths are seen by the system as interferences.

SC frequency offset estimator is the most precise of all the studied. It has lower mean and variance than MM estimator but its estimation range is smaller. Thus combining them we obtain both, a estimation range bound to L with the same precision than the SC estimator with almost no increase in the computational cost as the last one can be obtained from the resulting data of calculating the timing estimation.

Although the normalized mean squared error of the obtained channel estimation is between 5 and 10dB lower than the first estimation, it will be seen in Chapter six that it isn't still enough to obtain good results. Thus, it was considered a channel estimation method whose performance will increase for longer training sequences. It will be comment, in Chapter seven, which could be the estimated effects of increasing the length of the training sequence.

It was seen that the performance of the proposed methods are determined by the SNR, the selected training sequence and the selected training symbol structure itself. SNR influences heavily the performance of frequency offset and CSI estimators. Transmitting all the available energy during the training symbol will improve the overall performance of the transmission system. However, we need to take into account that the interference level should be kept under control in order to avoid generating interference to other systems which might be using adjacent bandwidth.

Finally, we must remember that the remaining frequency offset is an upper bound for the length of the transmitted frame. On the other side, there is a parameter to control the ISI introduced in the system, the decision levels of the channel estimation. As we commented, the maximum ISI level allowed in our system is bound to the distortion normalized mean square error of the channel estimation. So, there is a trade off between these parameters. One must keep in mind that these missed small taps, for low SNR, will almost be composed by noise. Thus, selecting them, we will be introducing distortion in the channel estimation as well.

Now, Alice and Bob, know that there is a module inside their transmitter and receiver that synchronize their signals and estimates the changes that the transmitted signal suffered when it was conveyed through the channel. Next, they will know how these changes are corrected.

4 The Encoder and Decoder

In this chapter the decoding stage will be introduced. First, an overview about coding theory will be done in order to clarify the algorithms, approximations and architectures to perform the decoding process in the receiver. Second, the message passing algorithm and LDPC codes are introduced and the obtained simulation results over AWGN channel are commented. At last, the simulation results of the decoding process in AWGN channel are exposed.

4.1 First Approach

In a typical receiver, the data received from the channel is processed with an equalizer to mitigate the effects of ISI. The equalizer produces estimates of the data, which are passed onto the decoder for the error correction stage.

Once data is equalized, the decoding process starts exploiting the bit redundancy introduced by the transmitter. In classical decoding theory, hard decision samples of the bits, $b_i \in GF(2)$, are estimated from the symbols at the output of the equalizer and passed to the decoder to perform its task. This stage is usually performed by different ECC usually, which can be grouped into two main families, Convolution Codes (CC) and Parity Check codes (PC).

A new approach appeared in 1993 based on message passing algorithms where data is quantized in real numbers and decoded multiple times achieving performances close to the Shannon limit. A new decoding theory appeared since then based on these algorithms and techniques, named the Soft Input Soft Output (SISO) decoding. This new concept has been applied to Channel Estimation and Channel Equalization improving the overall performance of these systems.[10] [11]

4.1.1 About classical coding theory

In this section, we will discuss classical codification algorithms highlighting the differences between the classical and the SISO approaches.

A CC encoder is a finite-memory system that generates n_0 binary digits every k_0 presented at its input. The decoder calculates all the possible paths, as the probability of choosing a path different from the all-zero path. Error events are introduced when the selected path diverges from the correct path until it merges into it. When the decoder chooses an incorrect path forming an error event, the errors accumulated during the periods of divergence cannot be corrected, since after merging again the correct and error event paths will accumulate the same metrics.

A PC code is a channel binary block code with a codebook C, formed with a set of 2^k codewords of length n generated by a matrix of dimension nxk . Their correction capacity is given by its minimum Hamming distance d_{min} , the minimum number of bits that differ between codewords of the codebook. These codes can correct received words with $t=(d_{min}-1)/2$ errors, searching if any codeword is present

within the Hamming distance t, if this is the case the decode declares that codeword to be the received codeword, if not declares an unrecoverable error.

As we can see, a simple way to increase performance of both CC and PC coding schemes is to enhance their correcting capacities by adding more redundancy. The robustness of the codes against the transmission channels effect will be increased but at the cost of reducing the throughput the channels. A trade-off must be established between achieving capacity and error correcting capacities and neither are good choices.

4.1.2 About soft input soft output algorithms and LDPC codes.

In 1993 C. Berrou, A. Glavieux, and P Thitimajshima [18] presented their approach to error correcting coding, termed turbo-coding. Coding theory, after 50 years of slow progress, had suddenly made a giant leap: the new technique, with complexity only a small factor larger than that of standard coding schemes, enabled performance approaching the Shannon capacity of the additive white Gaussian noise channel within a fraction of a dB. The original seed for this revolution had been first proposed by R. Gallager in 1960 in his PhD thesis, Low-Density Parity-Check Codes [12]. In this thesis Gallager introduced two ground-breaking concepts: a bounding technique to assess the maximum-likelihood performance of coding systems and low-density parity-check (LDPC) codes together with their associated iterative decoding algorithms.

4.2 SISO, Message Passing Algorithm and LDPC Codes

In this section, the passing messages algorithm will be exposed and the LDPC presented but firstly a little explanation of which qualities are needed to achieve channel capacity in a code.

4.2.1 Performance bounds for transmission on noise channels.

Until turbo-codes were proposed, improving the performance of transmission systems could be realized decreasing the transmission rate (transmitting less bits through the assigned bandwidth), increasing channel capacity (increasing the transmitted power) or increasing the code block length (increasing complexity process delays).

Shannon's law provides hints on the features that capacity-achieving codes should have. “If the source entropy is less than the channel capacity, then the source output can be reproduced at destination with arbitrary small probability using a sufficiently complex encoder and decoder”. Thus all rates below capacity are achievable. For every rate (R) below channel capacity (C) a sequence $(2^{(nR)}, n)$ with $P_e = 0$ if:

- There are not encoder/decoder complexity constraints.
- The Code has a infinite block length.
- There are non-constructive proof based on calculating the average error probability of error over a random choice of codebooks.

4.2.2 Low density parity check codes.

LDPC codes are linear codes with sparse parity-check matrix H , selected to provide the random-

like desire feature of capacity-achieving codes. They have a natural representation in terms of a bipartite graph, a graph with two kinds of nodes with edges connecting two nodes of different types. The graphs are drawn according to the following rule:

- There are as many variable (bit) nodes as coded bits.
- There are as many check nodes as parity check constrains (code redundancy).
- Check node *j* is connected to a variable node *i* whenever elements in matrix $H_{(i,j)}$ is a 1.

A $code(n,k)$ has 2^k codewords of length n, as for parity check nodes, and the rate being defined by $r=k/n$, being H of size $(n-k)\ x\ n$.

In figure 4.2 we can see an example of how graphs are drawn for LDPC codes,. for the Hamming code $H(7,4)$ (Hamming codes are not LDPC codes but they are generated by matrix, thus we can extend this example to assess how they work) with the defined matrix H. There are as many bit nodes (designed by "=") and check nodes (designed by "+") as columns and rows the matrix H has . The bit node grade is defined as the number of edges at the bipartite graph departing from the nodes (d_b) and, similarly, the check node grade as the number of edges at every check node (d_c).

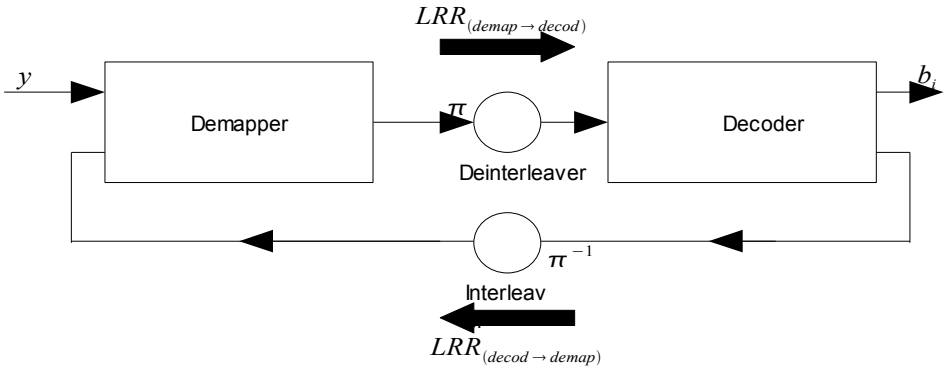


Figure 4.1: SISO demapper-decoder information feedback

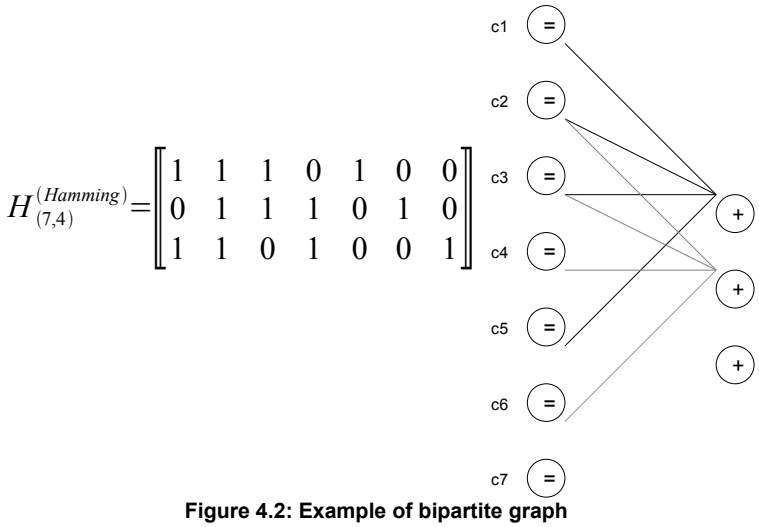


Figure 4.2: Example of bipartite graph

A LDPC code is regular if the number of 1 in every column and row is constant. In a general, the

number of ones in matrix H must be sparse, so there are a few edges departing from every node. The complexity of the decoding process increase with the number of edges at bit nodes and check nodes because actualization operations are done for every single edge in every node every decoding iteration.

LDPC codes generation won't be considered in this dissertation but remarking the following points is needed:

- Code performance mostly depends on the code degree profile (d_b and d_c).
- The matrix H design has influence on the error floor of the code.
- The performance of the code increases if the number of cycles in the graph is small, although the performance of the decoding algorithm is good if they exist. Thus we must avoid the short cycles because they limited the performance of the codes.
- The ones in matrix H are located in pseudo-random positions, meaning that random-like appearance of capacity-achieving codes is obtained.
- Decoding complexity is linear with codeword length n and encoding complexity grows as n^2 multiplied by a small factor.

4.2.3 Passing message algorithms for LDPC codes. soft input soft output quantification.

The passing message algorithm, also known as sum-product for binary factor graphs algorithm , is the basis of the iterative decoding architecture. In every iteration we do as follows:

- In the first iteration messages are passed to bit nodes from the demapper.
- In the second step in each iteration, bit nodes pass these messages to the check nodes connected to their edges.
- In the next step, check nodes weigh the received messages and forward the computed output messages to the bit nodes connected to their edges.
- Keeping in mind that the output messages from every node don't take into account the input messages received from this node. In other words, when calculating the output message from node i to node j, the a priori information received at node i from node j must not be taken into account. If not, there will be a performance degradation in the decoding process.
- In the following iterations, in bit nodes, the incoming messages from the demapper and the check nodes are added and, keeping in mind the upper point,are sent to the check nodes. And so on for every iteration.

Now it will be defined what these messages, also called The log-Likelihood Ratio (LLR), are. LLR are defined by (4.1) being the equivalent representation of binary variables and quantifying the probabilities of a given bit into a real number, $LLR \in \Re$. The LLR is used for better numerical stability, because the might differ in several orders of magnitude.

$$LLR(b_i)=\log\frac{(P(c_i=1))}{(P(c_i=0))} \quad (4.1)$$

Once defined, the next question is how a LLR can be obtained. As we now, in an AWGN channel the received sample y is defined by (4.2), being $w(n)$ the sampled noise Gaussian distributed $N(0,\sigma_n)$ and $s(n)$ the transmitter symbol.

$$\begin{array}{l} y(n)=s(n)+w(n) \\ w(n)=y(n)-s(n) \end{array} \quad (4.2)$$

The probability of y conditioned s is Gaussian distributed, as shown in (4.2), so they can be evaluated as shown in (4.3) in a Gaussian complex baseband channel.

$$p(y/x=s_j)=\frac{1}{(\pi\ \sigma_n^2)}*e^{\left(\frac{-\left[(y(n)-s_j(n))\right]^2}{(2\sigma_n^2)}\right)} \quad (4.3)$$

We can obtain then $P(b_i=0)$ as in (4.4):

$$P(b_i=0)=\sum_{S:ci=0} p(y/x=S)\,p(x=S) \quad (4.4)$$

In the first step, the probability of receiving $b_i=0$ or $b_i=1$, $i\in Q$ being the constellation size defined as 2^Q , is the same. Thus all the symbols have the same probability, $1/2^Q$ but, in next iterations they will not be equal. Then the probability of the received symbol being s_j is defined by (4.5), being $P_a(b_i)$ the bit a priori probabilities calculated on the previous iteration. We must not take into account the probability of the bit that we are calculating when we obtain the probabilities of the symbols.

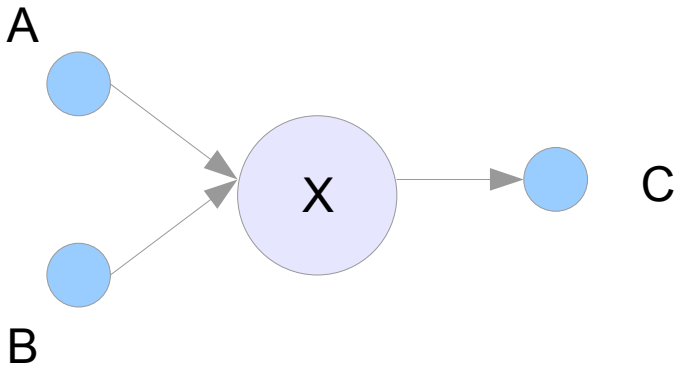
$$P(x=s_k)=\prod_{b_j=1,i\neq j,\,j\in s_k} P_a(b_j) \quad (4.5)$$

The messages passed from the demapper to the bit nodes are obtained by evaluating the following expression (4.6):

$$LLR(b_i)=\log\left(\frac{\left(\sum_{S:ci=1} p(y/x=S)\,p(x=S)\right)}{\left(\sum_{S:ci=0} p(y/x=S)\,p(x=S)\right)}\right) \quad (4.6)$$

In bit nodes, the incoming messages from the demapper are added with the incoming messages from the check nodes and forwarded to their check nodes (4.7). The output message from a bit node to a check node doesn't depend on the incoming message from that check node, as shown in figure 4.3.

$$LLR_{(X\rightarrow C_n)}=\sum_{(j=1\ j\neq n)}^{N}\left(LLR_{(C_j\rightarrow X)}\right)+LLR_{(demap)} \quad (4.7)$$



$$LRR(X \rightarrow C) = LRR(A \rightarrow X) + LRR(B \rightarrow X)$$

Figure 4.3: Bit node function scheme

An example of a check node operation, with $d_c = 3$, can be found in the figure 4.4 and equation (4.8). Operating as in [12], a generalized equation for a check node with d_c edges is obtained as shown in equation (4.9). Once again, the incoming LLR from X to A is not taken into account when calculating the LLR from A to X.

$$LLR(A \rightarrow X) = \log\left(\frac{(P(Y=1)P(Z=0) + P(Y=0)P(Z=1))}{(P(Y=1)P(Z=1) + P(Y=0)P(Z=0))}\right) \quad (4.8)$$

$$LRR(A \rightarrow X_m) = (-1)^{(d_c-1)} \prod_{i=1, i\neq m} sign(LLR(X_i \rightarrow A)) \psi\left(\sum_{i=1, i\neq m} \psi\left(\left|(LRR(X_i \rightarrow A))\right|\right)\right) \quad (4.9)$$

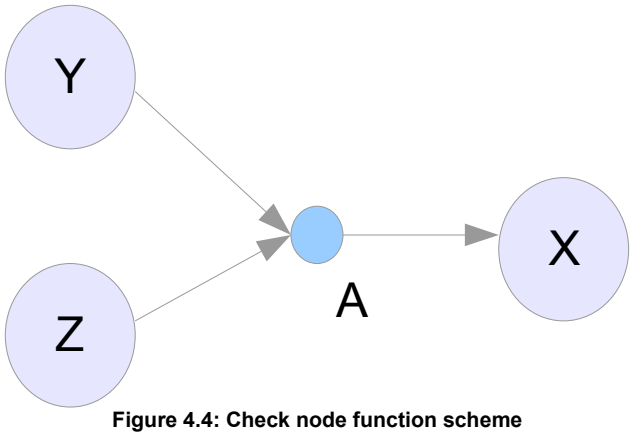


Figure 4.4: Check node function scheme

The last step, when the message decoding process is ended, is to calculate the $\hat{b}_i$ as follows:

$$\hat{b}_i = sign(LLR_{(demap \rightarrow decod)}(b_i) + LLR_{(decod \rightarrow demap)}(b_i)) \quad (4.10)$$

4.2.4 Influence of the constellation labels, the block-length over the BER and the number of decoding and demapping iterations

In figure 4.5 are depicted BER simulations of 16QAM constellation symbols over an AWGN for different block lengths and constellation labeling and different decoding schemes. As shown in figure below, the behavior of the BER depends on the labeling of the modulated symbols. The GRAY labeling performance for a non iterative detection and for a iterative detection, are very close but for the Set Partitioning labeling

the gain is higher when we use an iterative detection scheme. It is needed to state that the slope at the convergence zone depends on the labeling as well.

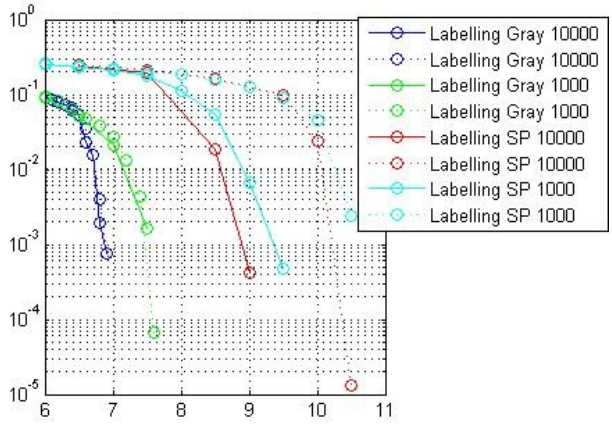


Figure 4.5: BER comparison for different labeling and codeword length for 16QAM constellation.

In the present work, the system is designed to transmit codewords of length 1000, and the QPSK, 16QAM and 64QAM constellation. All them are mapped following the Gray criterion, the labeling that minimizes the effects of errors in the conversion of analog signals to digital. The number of decoder iterations was set to 5 and the number of demapper iterations to 4. Improvements in BER can be easily obtain increasing these values but at the cost of increasing the computational time as well.

4.3 Performance of the Decoder in an AWGN channel.

In this section the simulation results of the implemented LDPC decoder for each one of the constellations of the symbol are presented.

4.3.1 Performance in an AWGN channel

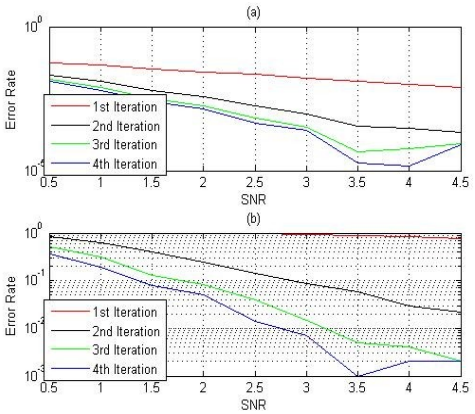


Figure 4.6: QPSK constellation. (a) BER. (b) WER

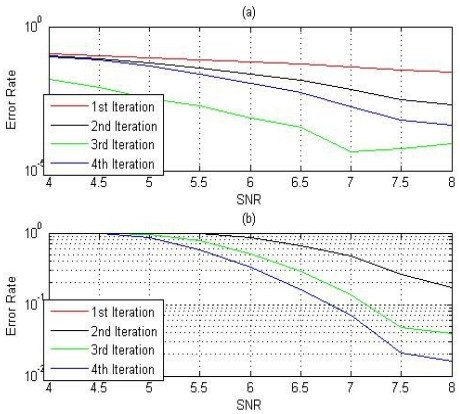


Figure 4.7: 16-QAM constellation. (a) BER. (b) WER

2000 codewords are transmitted for each simulation. The number of demapper iterations was set to 4 and the number of decoder iterations to 5, as mentioned earlier. Thus the results are reliable for WER and BER above 10^{-2} and 10^{-4} respectively.

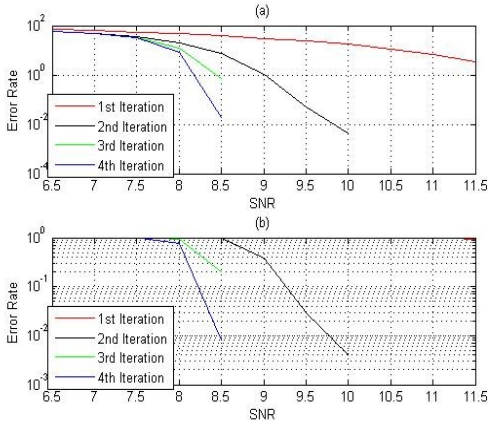


Figure 4.8: 64-QAM constellation. (a) BER. (b) WER

Figure 4.6 to 4.8 show the BER and WER evolution of QPSK, 16-QAM and 64-QAM constellations with Gray labeling for the given iterative detection scheme. Each plot shows the demapper output in a given iteration. We observe that at each demapper iteration the error rate of the system decreases. Thus there is a performance improvement in each iteration. The waterfall region of each constellation will be around 3.5dB for QPSK modulation, 7dB for 16QAM and 9dB for 64QAM.

4.4 Conclusions.

In this chapter the message passing algorithm and the LDPC decoding algorithm were introduced. The hints required for obtaining a, so called, 'good' code were also adressed.

It was shown that the performance of the LDPC codes is bound to the codeword length, the chosen symbol label and the number of decoder and demapper iterations. Thus, we know that the performance of the proposed system can be increased by changing these values.

Now, Alice and Bob know that there is a block in the receiver aimed to correct the errors introduced onto the message. They know that it passes iteratively the information received to some kind of nodes, the check nodes, where the information is mixed and forwarded to other nodes, the bit nodes. There, the messages received from the check nodes help to increase the probability of obtaining the bit transmitted by Alice. Thus, there is a cooperative process where the bits with high reliability help to their mates with less reliability to correct their values which leads to an increase of the reliability of the data estimation after each iteration.

5 SISO Channel Capacity and Bit Allocation Algorithms.

In this chapter, the optimal bit allocation to achieve the capacity of the current channel is introduced. In the first section, some previous concepts are explained. Second, the theoretical concept of channel capacity is exploited to reach an implementable result. Next, the algorithm to calculate the capacity of every sub-channel is given. Finally, an overview about the average transmission throughput is given.

5.1 Introduction

5.1.1 Some concepts.

The goal of a transmission system is achieve the theoretical capacity bound given by Shannon cited in section 4.2.1. Reaching this goals is possible when the data stream is transmitted with bit error rate equal to zero matching the channel capacity bound. In a AWGN channel, the capacity bound is given by (5.1) being C the achievable capacity in bits per second, SNR, the average Signal to Noise ratio of the samples conveyed through the channel (linear value) and B the available transmission bandwidth.

$$C = B \cdot \log_2(1 + SNR) \quad (5.1)$$

SNR (linear)	1	3	7	15
Capacity (Mbps)	5	10	15	20

Table 5.1: Some channel capacities for given a SNR

The *transmission throughput* is defined as the amount of information data bits conveyed through the channel. First, we should define the transmission efficiency as in (5.2), being N_{synch} the number of synchronization symbols transmitted in a frame, N_{OFDM} the number of OFDM symbols transmitted, T_{CP} the per cent of time that the transmission of the Circular Prefix last and N_{total} the total number of transmitted symbols. Following, the throughput can be calculated as in (5.3), being R the transmission rate. As we know, the data stream is composed by bits of two kinds the information and redundancy bits the first generated by the user, carrying the information to be transmitted, and the second kind added by the ECC in order to enable a data correction process at the receiver. So the throughput is defined as the average information data received by the destination user.

$$\eta = 1 - \frac{N_{synch} + N_{OFDM} * T_{CP}}{N_{total}} \quad (5.2)$$

$$Throughput = \eta \cdot C \cdot R \quad (5.3)$$

5.1.2 About bit allocation

The power available is a very important design constrain when receivers are modeled. For mobile receivers, the energy available is limited to the supplied by its battery, thus one of the main goals when designing this kind of devices is to assign optimally the power available to maximize the transmission

throughput.

Optimal bit allocation spreads the available energy through the assignment bandwidth trying to achieve the fading channel capacity. Basically, by using the estimated frequency impulse response of the channel, in a process named Water-filling, the sub-channels with a lower attenuation are selected and the energy assigned to each of these sub-channel is optimized maximizing the amount of data transmitted.

When there isn't any knowledge of the channel impulse response, we can't apply Water-Filling. In this case the optimal bit allocation algorithm is UPA: Uniform Power Allocation. We transmit the same amount of power in each sub-channel. Then we expect that the receiver could correct the errors introduced when the message is conveyed through the channel.

5.1.3 Adaptive modulation

After obtaining the optimal power allocation per channel, each sub-channel is filled with constellation symbols carrying the amount of information that can be received free of errors. The amount of information bits transmitted depends of the constellation label selected and of the redundancy added by the encoding stage. Thus, after obtaining the optimal power allocation per sub-channel, there is this symbol mapping stage, this process is known as the adaptive modulation stage.

5.2 Achieving Channel Capacity

This section begins with a notion of mutual information in communication to continue with the analysis of SISO channel capacity. The following explanation is extracted from: M. Lagunas “Array processing: Spatial diversity in radio-communications” 2006 [1]. The vectors are denoted with a trace below the letter and the matrices with two traces below its letter.

5.2.1 Mutual information analysis

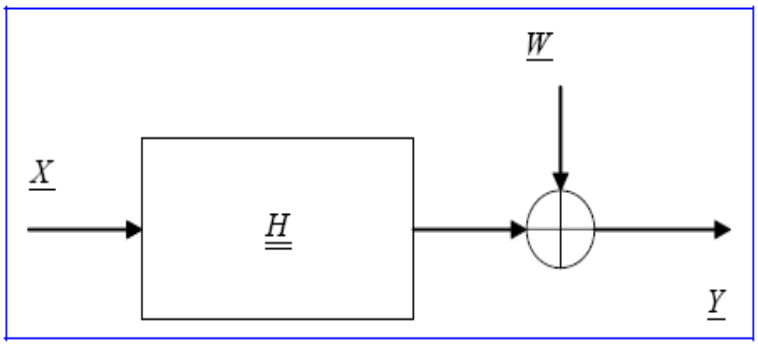


Figure 5.1: Channel, seen by Information.

Considering the transmission system depicted in figure 5.1 where vector X is the transmitted signal, matrix H is the one of SISO channel considering flat fading and Y is the received signal, mutual information between transmitted signal and received one is the following, where f( ) means probability distribution:

$$I(\underline{X},\underline{Y}) = \iint f(\underline{X},\underline{Y}).Ln\left[\frac{f(\underline{X},\underline{Y})}{f(\underline{X}).f(\underline{Y})}\right].d\underline{X}.d\underline{Y} \quad (5.4)$$

Gaussian distribution for the transmitted signal and noise is assumed as it is not far from reality, but it's not obvious than transmitted signal has also this same distribution. Only when we have a linear process, this supposition is correct. One more interesting thing is that it is possible to prove than mutual information is maximal when distribution is Gaussian, so an upper value can be found, denominated SISO channel capacity (C).

This way transmitted signal and noise Gaussianity are assumed, being the probability of $\underline{X}$ defined in (5.5) and (5.6) with the following statistics:

$$\underline{X} = G\left(\underline{0},\underline{\underline{Q}}\right) \quad \underline{W} = G\left(\underline{0},\underline{\underline{R}}_0\right) \quad (5.5)$$

$$\Pr(\underline{X}) = \det\left(\underline{\underline{Q}}^{-1}\right).\exp\left\{-\left(\underline{X}^H.\underline{\underline{Q}}.\underline{X}\right)\right\} \quad (5.6)$$

Noting the two basic terms that compose a Gaussian distribution, there is a term corresponding to covariance matrix inverse that doesn't depend on integration variables and another which depend, inside exponent. Following, we substitute (5.6) in (5.4) obtaining (5.7) where the first term is the logarithm of determinants quotient and the second terms is zero because both terms have zero mean. Thus we obtain (5.8).

$$C = \iint f(\underline{X},\underline{Y}).Ln\left[\frac{\det\left[\underline{\underline{Q}}\right].\det\left[\underline{\underline{R}}_{yy}\right]}{\det\begin{pmatrix}\underline{\underline{Q}} & \underline{\underline{R}}_{xy}\\ \underline{\underline{R}}_{yx} & \underline{\underline{R}}_{yy}\end{pmatrix}}\right].d\underline{X}.d\underline{Y} + \quad (5.7)$$
$$-\iint f(\underline{X},\underline{Y}).\left[\underline{X}^H.\underline{\underline{R}}_{xy}.\underline{Y} + \underline{Y}^H.\underline{\underline{R}}_{yx}.\underline{X}\right].d\underline{X}.d\underline{Y}$$

The capacity expression in (5.8) can be simplified as we can see in (5.9), where understanding determinant as an energy or power measure of their corresponding matrix, there is a relationship between received signal and the part of received signal which doesn't corresponds to transmitted signal. This SNR notion in capacity expression will be revised afterwards.

$$C = Ln\left[\frac{\det\left[\underline{\underline{Q}}\right].\det\left[\underline{\underline{R}}_{yy}\right]}{\det\begin{pmatrix}\underline{\underline{Q}} & \underline{\underline{R}}_{xy}\\ \underline{\underline{R}}_{yx} & \underline{\underline{R}}_{yy}\end{pmatrix}}\right] \quad (5.8)$$

$$C = Ln\left[\frac{\det\left[\underline{\underline{R}}_{xy}\right]}{\det\left[\underline{\underline{R}}_{yy} - \underline{\underline{R}}_{yx}.\underline{\underline{Q}}.\underline{\underline{R}}_{xy}\right]}\right] \quad (5.9)$$

As usually noise is uncorrelated with transmitted signal, the received signal covariance and also crossed covariances can be written as follows:

$$\begin{aligned}\underline{\underline{R}}_{yy} &= \underline{\underline{H}}.\underline{\underline{Q}}.\underline{\underline{H}}^{H} + \underline{\underline{R}}_{0} \\ \underline{\underline{R}}_{xy} &= \underline{\underline{Q}}.\underline{\underline{H}}^{H} \\ \underline{\underline{R}}_{yy} &= \underline{\underline{H}}.\underline{\underline{Q}}\end{aligned} \quad (5.10)$$

Being the capacity denominator a measure of received noise. It's possible to simplify the expression in (5.9) as follows

$$C = Ln\left[\frac{\det\left[\underline{\underline{R}}_{0} + \underline{\underline{H}}.\underline{\underline{Q}}.\underline{\underline{H}}^{H}\right]}{\det\left[\underline{\underline{R}}_{0}\right]}\right] = Ln\left(\det\left[\underline{\underline{I}}_{n_R} + \underline{\underline{R}}_{0}^{-1}\underline{\underline{H}}.\underline{\underline{Q}}.\underline{\underline{H}}^{H}\right]\right) \quad (5.11)$$

$$C = Ln\left(\det\left[\underline{\underline{I}}_{n_R} + \underline{\underline{R}}_{0}^{-1}\underline{\underline{H}}.\underline{\underline{Q}}.\underline{\underline{H}}^{H}\right]\right) = Ln\left(\det\left[\underline{\underline{I}}_{n_R} + \underline{\underline{H}}^{H}\underline{\underline{R}}_{0}^{-1}\underline{\underline{H}}.\underline{\underline{Q}}\right]\right) \quad (5.12)$$

From (5.12), we can see that white noise suppose is not necessary if we consider than channel has inside the noise contribution. Thus always white noise will be considered, and if there isn't, we must include their contribution as indicated in the last expression. In order to simplify a bit more, we will suppose that transmitter uses Uniform Power Allocation, that is, we will suppose that the transmitter covariance matrix is equal to unity matrix, obtaining the expression in (5.13).

$$C_{SISO} = Ln\left[1+\gamma\right] \quad (5.13)$$

$$\gamma = \frac{E_T.Trace\left(\underline{\underline{H}}^{H}.\underline{\underline{H}}\right)}{N_0} \quad (5.14)$$

5.2.2 Channel capacity of a flat fading channel.

The expression above shows as the channel capacity depends on the channel. Thus we will have a different throughput during the transmission if we assume that the channel completely changes in every frame. From now on, for the design process of the water-filling module we will assume that

$$
\begin{aligned}
C &= \ln\left(det\left[\underline{\underline{I}} + \underline{\underline{Q}}\,\underline{\underline{R_H}}\right]\right) \\
\underline{\underline{R_H}} &\equiv \underline{\underline{H}}^H \underline{\underline{R_o}}^{-1} \underline{\underline{H}} \\
\underline{\underline{R_H}} &= \underline{\underline{U_H}}\,\underline{\underline{D_H}}\,\underline{\underline{U_H}}^H \\
diag\ \underline{\underline{D_H}} &= 1/N_o\left[\lambda_{H_1} \ \ldots \ \lambda_{H_N}\right]
\end{aligned}
\tag{5.15}
$$

5.3 Bit Allocation Algorithm to Achieve Channel Capacity with CSI.

For channel design, we will follow two basic steps. The first one refers to the Power Allocation and the second one will be the constellation (Adaptive Modulation) design.

5.3.1 Power allocation.

As shown in [1], we must design the covariance matrix $\underline{\underline{Q}}$ that diagonalizes the channel matrix. All the processes will have it as a goal, reducing the design problem to channels without ISI, with a gain determined by the correspondent eigenvalue.

The first step, the maximum capacity design is aimed to determinate the coefficients of the matrix $\underline{\underline{Q}}$. This matrix must be diagonal if we assume that there isn't ISI, thus matrix $\underline{\underline{R}}_H$ is diagonal. The maximum capacity design is restricted to a determinant of a diagonal matrix which contains $\underline{\underline{Q}}$ entries by $\underline{\underline{D}}_H$ entries, with the restriction of constant transmitted energy (5.16). Thus,

$$
\begin{aligned}
C &= \sum_{q=1}^{N} \ln\left(1 + \frac{1}{N_o} \cdot z(q) \cdot \lambda_H(q)\right) \\
&\sum_{q=1}^{N} z(q) = E_T
\end{aligned}
\tag{5.16}
$$

The solution to this problem is given by minimizing the Lagrangian function with the given restriction. After minimizing the following result is obtained:

$$
z(q) = \begin{cases} \mu - \dfrac{1}{\lambda_H}(q) > 0 & when \quad \mu - \dfrac{1}{\lambda_H}(q) > 0 \\ 0 & when \quad \mu - \dfrac{1}{\lambda_H}(q) \le 0 \end{cases}
\tag{5.17}
$$

Where the μ parameter assures an active $z(q)$ sum to available energy. From this solution, we can extract that all channel eigenmodes won't be used; only when available transmitted energy is high, then all modes will be activated.

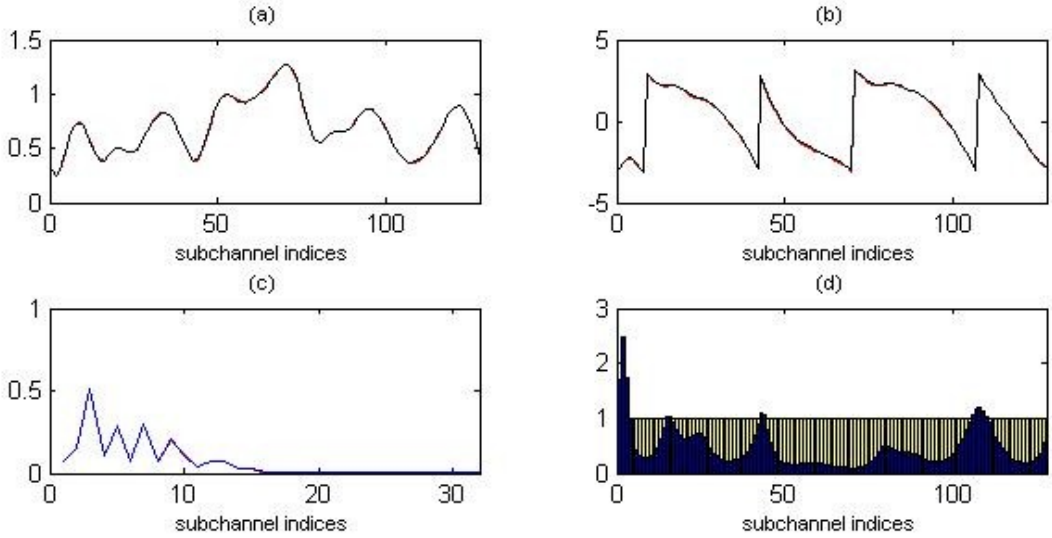


Figure 5.2: (a) $|H(f)|$ (b) $angle\, H(f)$ (c) $|h(n)|$ (d) Yellow: Energy level per sub-channel. Blue: Attenuation of each sub_channel.

In figure 5.2d the result of the Water-filling algorithm for the channel given in figure 5.2a, 5.2b, 5.2c and an average SNR level of 8 dB is shown. It must be noted that this algorithm assigns more power to the best channels, keeping constant the value of the total energy level plus the noise to carrier ratio in each sub-channel, as seen in figure 5.2d.

5.3.2 SNR required to guarantee a free error transmission.

To note that although the average SNR is 8 dB, the SNR of each sub-channel is different (figure 5.3a). Thus, we can now apply the data previously obtained in order to know the SNR levels of each sub-channels. This information will be of great utility because, how we will be shown in the next chapter, it will help to equalize properly the data streams transmitted in each sub-channel. It will also help to map the optimal label with the proper rate onto each sub-channel in order to achieve the free error transmission requirement.

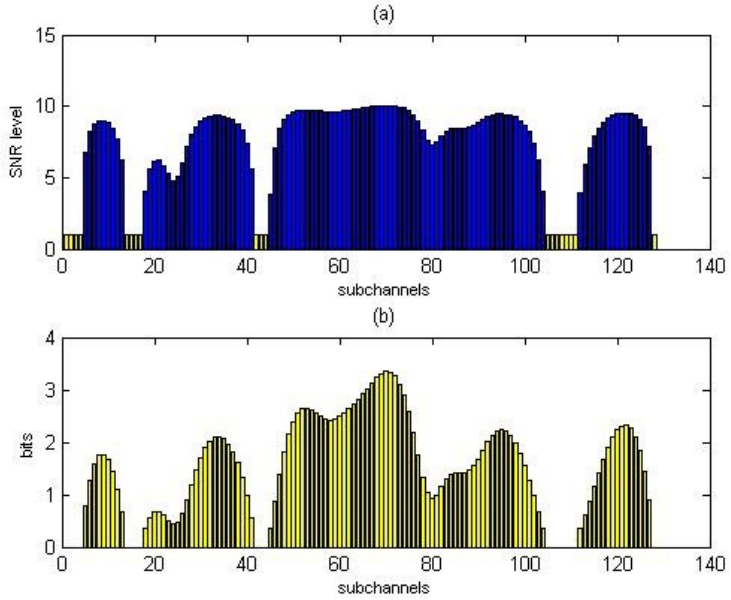


Figure 5.3: (a) SNR levels of each sub-channel. (b) Amount of information bits to allocate to each sub-channel.

As showed in Chapter 4, every constellation label required a different SNR level to almost guarantee the free error transmission condition to achieve channel capacity.

5.3.3 About constellation labels and bit rates

Now the sub-channel must be filled with data streams. Each one of them carries a QPSK, a 16-QAM or a 64-QAM modulated symbols or two, four or six coded bits respectively. The decoder has a rate ½, so each one of them will carry only one, two or three information bits respectively. In figure 5.3b we can see that the optimal bit load of each sub-channel is between 0 and 3 bits. Thus, the data streams won't be optimally filled.

To optimize more the system, it is necessary to add another decoder and/or more constellation labels to map tin a more optimal way number of information bits given by the WF algorithm onto each sub-channel. This solution seems easy to implement, a priori, but the complexity and the processing delay of the receiver will increase because two different decoder modules would be needed, at least. Thus the overall monetary cost of both, the receiver and the transmitter, will increase.

The criteria followed for the assignment of a given label to each sub-channel will be the SNR obtained after the power allocation optimization, thus a study of the BER evolution for every labeling is required in order to know the SNR that guarantees a free error transmission. In Chapter 6 Section 3, the performance of the joint decoding and equalization system will be given. In figure 5.3 we can see the capacity of each sub-channel given by the WF algorithm.

SNR range	Bits to map into the data stream
< 7.45 dB	1
7.45dB – 11.2dB	2
> 11.2dB	3

Table 5.2: SNR levels for free error transmitting a constellation symbol

Table 5.2 we can see the border levels of SNR chosen, in the proposed system, to map each type of constellation. Although that we will be transmitting QPSK modulation, it is possible, under the required SNR level to guarantee the free error transmission condition, we know that the LDPC code will be able to correct them if the rest of the symbols are received with a high reliability level. So, increasing the power level required to transmit the other constellation symbols will allow to receive a great number of correct codewords, thus we won't be able to achieve the capacity of the transmission system because the average throughput of the system will decrease.

5.4 Signal mapping and demapping

The following figures show the mapper and demapper structures. First, the Water-filling module is feed by the CSI received from the receiver and the Water-filling, computing the optimal number of bits to allocate in every data stream. Following, the encoded bits are passed to the Stream Mapping module. There, the bit stream is split in packets of size M_i , being M_i the number of bits to be mapped into the previously selected constellation, multiplexing every data stream (each one with the correspondent constellation symbol) into their sub-channels, setting up the OFDM symbols.

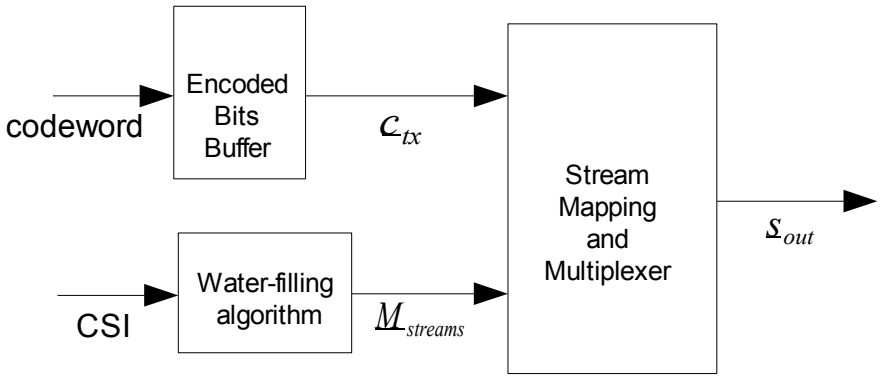


Figure 5.4: Mapping scheme

As said before, there are channels with and without data stream. Thus, at the receiver, the active sub-channels are de-multiplexed and its data stream passed to the equalizer input symbol entrance.

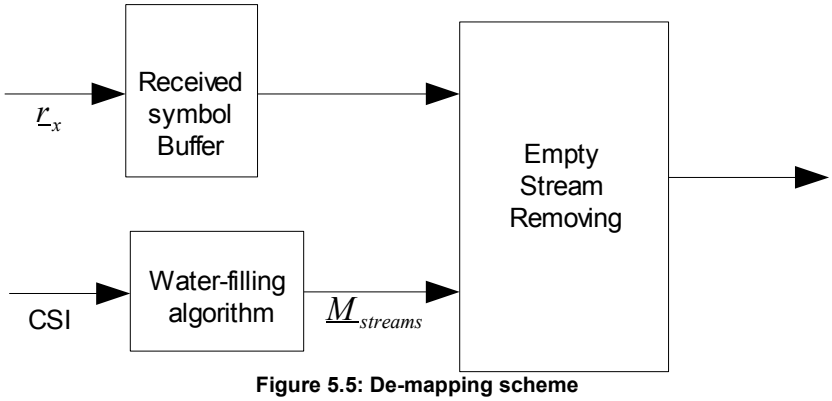


Figure 5.5: De-mapping scheme

The system requires buffers to store the remaining coded bits to be transmitted (in the transmitter) and the remaining symbols to be processed (in the receiver) because generally, only a fraction of a codeword is transmitted in each OFDM symbol. Thus, the receiver must concatenate the received data streams, gathering all the streams with bits pertaining to the same codeword, passing them to the equalizer.

All this process must be synchronized between the transmitter and the receiver (because suddenly, the constellation size may change, but as both the receiver and transmitter are running the algorithms, both will realize the same changes at the same time.

5.5 Conclusions

In this chapter the flat fading channel capacity expression for Gaussian distributed signals was obtained. After optimizing this expression, the Water-filling algorithm was obtained. It was introduced that there is a different SNR in each sub-channel. Thus we know the noise level of each data stream which will be very useful in the next chapter. After that, the effect of mapping the results obtained from the WF algorithm onto the available constellations in the system was analyzed Next, a constellation assignment for the optimal energy levels of each data stream obtained from the WF algorithms was proposed. Finally, a scheme for the adaptive modulator was proposed.

The solution when transmitter knows perfectly the channel is Water-Filling, a procedure which establishes that, to obtain capacity is necessary to distribute power keeping constant the sum of this times

the inverse of channel gain, thus this solution assigns more power to the best channels. It was also shown that UPA is the optimal bit allocation when there isn't a perfect CSI.

The WF algorithm enables the optimization of the available energy in order to achieve the flat fading channel capacity. Although this is the upper bound of the achievable throughput, sometimes we can transmit above this capacity level. It happens in the proposed system because there isn't enough labels to map properly the WF results. When it happens, the transmitted information will have a higher error probability, though the BER will increase. The most important parameter that must be controlled, when setting mapping levels, is the average transmission throughput, because we will be transmitting at rates around the capacity, sometimes slightly higher and sometimes much lower, but the average must be kept under the average channel capacity. This effect will be discussed more deeply in Chapter 7.

Now Alice and Bob also know that their transmitter and receiver are all the time adapting the transmitted data streams to the behavior of the channel. They also know that the adaptation criterion optimizes the available amount of energy of each OFDM symbol trying to achieve the maximum throughput possible, the channel capacity. Thus, they begin to see their devices in a different light.

6 MMSE Turbo Equalization

In this chapter the equalization stage will be introduced. First, we do an overview about equalization and coding theory in order to clarify the algorithms, approximations and architectures to perform the decoding and equalization processes at the receiver. Second, the MMSE with a priori information equalization method is introduced. Next, the simulation results of the joint equalization and decoding system are analyzed and exposed. Finally the conclusions extracted from the simulation results are exposed.

6.1 First Approach

In a typical receiver, the data received from the channel is processed with an equalizer to mitigate the effects of ISI. The equalizer produces estimates of the data, which are passed onto the decoder for the ECC. For complexity reasons, the equalization task typically consists of linear processing of the received channel output (linear equalizer) and possibly past symbol estimates (decision feedback equalizer). The parameters of these filters can be designed according to a variety of different optimization criteria, such as the zero forcing (ZF) or Minimum Mean Square Error (MMSE) criteria.

Optimal equalization methods for minimizing the bit error rate (BER) and the sequence error rate are also nonlinear and are based on maximum likelihood (ML) estimation, which turns into maximum a posterior (MAP) estimation in presence of *a priori* information about the transmitted data. Reasonably efficient algorithms exist for MAP/ML sequence estimation, the Viterbi algorithm, and MAP/ML symbol estimation, the BCJR algorithm. We refer to these estimation methods as MAP/ML equalization.

Once data is equalized, the decoding process starts exploiting the bit redundancy introduced by the transmitter. In classical decoding theory, hard decision samples of the bits, $b_i \in GF(2)$, are estimated from the symbols at the output of the equalizer and passed to the decoder to perform its task. This stage is realized by different ECC usually, which can be grouped into two main families, Convolution Codes (CC) and Parity Check codes (PC).

A new approach appeared in 1993 [18] based on message passing algorithms where data is quantized in real numbers and decoded multiple times achieving performances close to the Shannon limit. A new decoding theory appeared since them based on these algorithms and techniques, named the Soft Input Soft Output (SISO) decoding. This new concept has been applied to Channel Estimation and Channel Equalization improving the overall performance of these systems. This joint process is also known as joint decoding and equalization.

6.2 MMSE Turbo Equalization

6.2.1 About turbo equalization.

A number of iterative receiver algorithms have been developed that achieve near-optimal performance by repeating the equalization and decoding tasks on the same set of received data, using feedback

information from the decoder in the equalization process. This method, which is called turbo equalization, was originally developed for concatenated convolution codes (turbo coding [9]) and is now adapted to various communication problems, such as trellis coded modulation and BICM schemes.

For channels with large delay spreads (long duration impulse responses) and for large constellation sizes, MAP/ML-based equalization suffers from impractically high computational complexity. This situation is only exacerbated in the context of turbo equalization, with the need to perform equalization and decoding several times for each block of data.

At the receiver, an approach with a linear equalizer was chosen, instead of MAP/ML-based equalization,where the filter parameters are updated using the MMSE criterion. This differs from conventional MMSE-based equalization methods, figure 6.1.b,in that the MMSE criterion is evaluated over both the distribution of the noise as well as the distribution over the symbols. In the context of turbo equalization, figure 6.1.c, the symbol distribution is no longer independent and identically distributed as is typically assumed for MMSE-based equalization, due to the information fed back to the equalizer from the error correction decoder. As a result, the coefficients of the equalizer change as a function of time and must be recomputed for each data symbol to be estimate. Figure 6.1.a shows the basic encoding chain employed in the transmitter. Figure 6.1.b show the most basic design, the data are directly extracted from the symbols. In the rest of the Chapter the structure shown in figure 6.1.d will be explained.

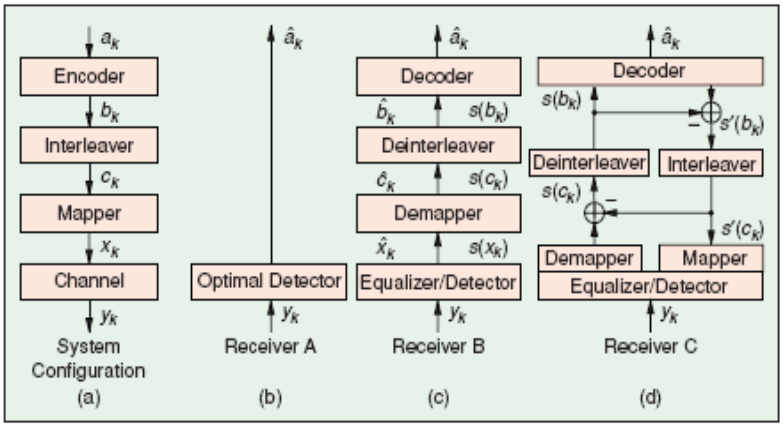


Figure 6.1: Different signal equalization strategies

6.2.2 MMSE equalization method using a priori information. A little overview.

We are trying to obtain a linear estimate $\hat{\underline{s}}$ of the transmit symbol using the observation $\underline{y}_n=[y_{n-N_2},\ y_{n-N_2+1},\dots y_{n-N_1}]$ of $N=N_1+N_2+1$ received symbols y_n given in (6.1).

$$\hat{\underline{s}}_n=\underline{a}_n\cdot\underline{y}_n+b_n \quad (6.1)$$

where $\underline{a}_n=[a*_{n,N_2}, a*_{n,N_2-1},---a*_{n,-N_1},]^T$, $a_{n,k},b_n\in C$ are the coefficients of the estimator. The parameters N_1 and N_2 specify the non causal and the causal part of the estimator filter, respectively, and N is the overall filter length. When we allow both $\underline{a}_n$ and b_n vary with n, we find that the choice

$\underline{a}_n = Cov(\underline{y}_n, \underline{y}_n)^{-1} \cdot Cov(\underline{y}_n, s_n)$ and $b_n = E(s_n) - \underline{a}_n \cdot E(\underline{y}_n)$ minimizes the cost $E(|s_n - \hat{s}_n|^2)$ [15]. This process will be generalized in the proposed system as in [9] to obtain MMSE with a priori information algorithm.

The joint equalization and decoding scheme is similar to the depicted in figure 6.1.d. In the demapper stage, the symbol probability estimation algorithm is slightly modified and new modules are introduced implementing the following functions:

- Symbol mean and variance estimation. Calculates the mean and variance of each symbol on basis of the a priori information available.
- Symbol equalizer. Processes the received signal samples, filtering the ISI.
- Symbol probability estimator. As the estimated symbol should be Gaussian distributed, the obtained estimated symbols are modified by some parameters in order to match this specification.
- Symbol extrinsic probability estimator. It has the same function as the demapper presented in section 4.2.3 (4.6).

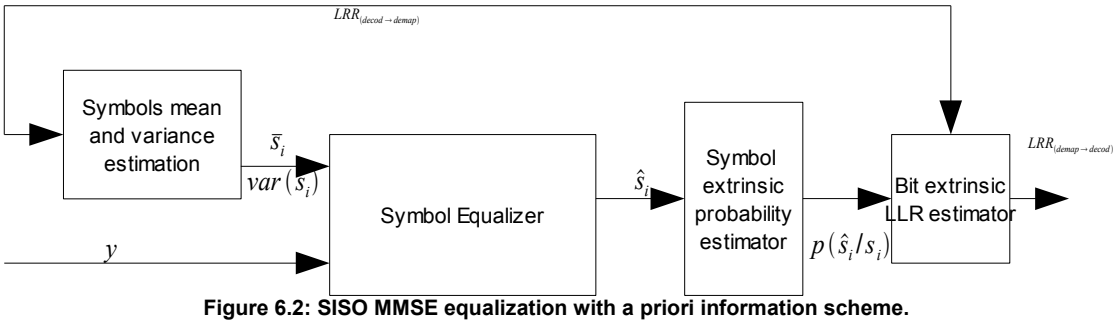


Figure 6.2: SISO MMSE equalization with a priori information scheme.

In figure 6.2 the followed equalization scheme is depicted, where y represents the data stream inputs, $LLR_{demap \to decod}$ are given by equation (4.6) and $LLR_{decod \to demap}$ by equation (4.10) in page 34.

Next, the modules functions and their algorithms are introduced.

6.2.3 Symbol mean and variance estimator.

The first step is to calculate an estimation of the mean and the variance of each symbol on basis of the a priori information available. For each possible transmitted symbol $s_j \in S$ in every sub-channel, we compute the $P(s_j)$ based on the corresponding bits a priori probabilities received from the decoder or, if it is the first equalizer iteration, assuming the same probability for all as in 4.5. Following the mean and the variance of each symbol can be computed as (6.2) and (6.3) respectively.

$$\bar{s}_j = E\{s_j\} = \sum_{s_j \in S} s_j P_a(s_j) \quad (6.2)$$

$$var\{s_j\} = E\{(|s_j|)^2\} - (|\bar{s}_j|)^2 \quad (6.3)$$

where the first term of equation (6.3) is obtained from equation (6.4).

$$E\{(|s_j|)^2\}=\sum_{s_j\in S}(|s_j|)^2 P_a(s_j) \quad (6.4)$$

6.2.4 Symbol equalizer

We must remind the approaches given in Chapter 2 that will be assumed to develop the expressions in our framework. Assuming that:

- There is only a single user transmitting information through the channel.
- As a first approach, perfect timing synchronization and the use of a CP. Thus there isn't ISI in our system.
- A FIR model of the channel (2.2) is used.
- The only noise present in the system is uncorrelated AWGN.
- The WF process is fed with a perfect channel impulse response estimation.
- The transmitted information is uncorrelated.
- A number of codewords are received in a frame, forming a data block with all them. Thus, all the codewords are equalized together and next each codeword decoded, obtaining their LLR. Next, all the LLR obtained are fed to the equalizer and another equalizer iteration is run and so on until the joint process finish.

The process in [9] is developed as in [10]. Thus, the expressions that follow hold.

- The channel matrix as in (6.5), where $\underline{\underline{\Xi}}$ is a matrix with the frequency channel estimation in its diagonal and otherwise is zero.

$$\underline{\underline{\Xi}}=diag\{H(1),H(2),...,H(N)\} \quad (6.5)$$

- The covariance matrix of the signal, $\underline{\underline{\triangle}}$, is a matrix with the variance of the symbols in its diagonal and zero elsewhere (6.6)

$$\underline{\underline{\triangle}}\equiv diag\, var\{\{\underline{s}\}\} \quad (6.6)$$

- The covariance matrix of the noise is a matrix with the noise variance of each data stream in its diagonal and zero elsewhere(6.7).

$$\underline{\underline{\Psi}}=diag\, var\{\{\underline{\sigma}\}\} \quad (6.7)$$

- Thus, the filter coefficients are given by the following expression:

$$\underline{\underline{\Gamma}}=(\underline{\underline{\Xi}}^H\cdot\underline{\underline{\triangle}}\cdot\underline{\underline{\Xi}}+\underline{\underline{\Psi}})^{-1}\cdot\underline{\underline{\Xi}} \quad (6.8)$$

- And then, the estimated data block will be obtained as in (6.9) because we don't have any ISI.

$$\hat{\underline{y}}=\underline{\underline{\Gamma}}\cdot(\underline{y}-\underline{\underline{\Xi}}\cdot\hat{\underline{s}}+\underline{\underline{\Xi}}\cdot\hat{\underline{s}})=\underline{\underline{\Gamma}}\cdot\underline{y} \quad (6.9)$$

After this process, estimates of the transmitted symbol are obtained. In the first iteration, the values in the diagonal of $\underline{\underline{\triangle}}$ are 1 but if we use QAM-based modulations, in the next iterations these values will increase or decrease because the symbols won't have anymore the same probability. Thus, the values of the variances of each sample will decrease when the LLR of the sample indicate that the transmitted symbol is the received one, otherwise they will increase. The initial MMSE process has been reduced to a typical equalization process where the samples are filtered depending of the channel properties and the noise but now they will be pondered by the variance of the estimated symbol as well.

6.2.5 Equivalent AWGN channel assumption.

At the output of the equalizer, in order to be able to demodulate the symbols, we assume that the estimated $\underline{\hat{y}}$ is the output of an equivalent AWGN channel, having $\underline{s_i}$ as its input (6.10), where μ_i and η_i are given by (6.11) and (6.12) respectively [8],[9].

$$\hat{y} = \mu_i \cdot s_i + \eta_i \quad (6.10)$$

$$\mu = \underline{\underline{\Gamma}}^H \cdot \underline{H} \quad (6.11)$$
$$\underline{H} = \underline{\underline{\Gamma}} \cdot \underline{1}$$

$$\eta = \underline{\underline{\Gamma}}^H \cdot \underline{H} - \underline{\underline{\Gamma}}^H \cdot \underline{\underline{\Xi}} \cdot \underline{\underline{\Xi}}^H \cdot \underline{\underline{\Gamma}} \cdot \sigma_{\hat{s}} \quad (6.12)$$

6.2.6 Symbol and bit extrinsic probability estimator

Now we can obtain the symbol extrinsic probability from (4.3) and then the bit extrinsic probabilities from (4.6) obtaining the symbol's probabilities from (4.4) and (4.5) how was shown in page 33.

6.3 Simulated Results.

In this section we present the simulation results of our system when it is fed by a perfect CSI, a perfect timing synchronization and a perfect frequency offset estimations for the different scenarios: Water-filling and UPA bit allocation.

We transmit 1000 frames per SNR level in each simulation and each frame carries 2 codewords. Thus 1 million of information bits are transmitted per each SNR.

We will allocate the constellation symbols in each data streams following the rule given in table 5.2.

6.3.1 UPA or WF bit allocation methods?

In figures 6.3 and 6.4 we show the simulation results for UPA and WF bit allocation respectively. The constellation decision levels were optimized to guarantee a WER around 10^{-2} for WF bit allocation.

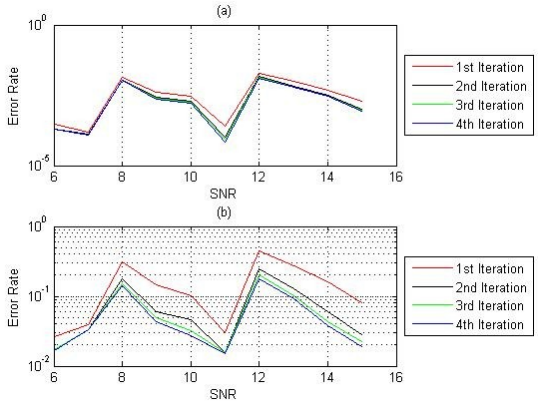


Figure 6.3: Uniform power allocation (a) BER (b) WER

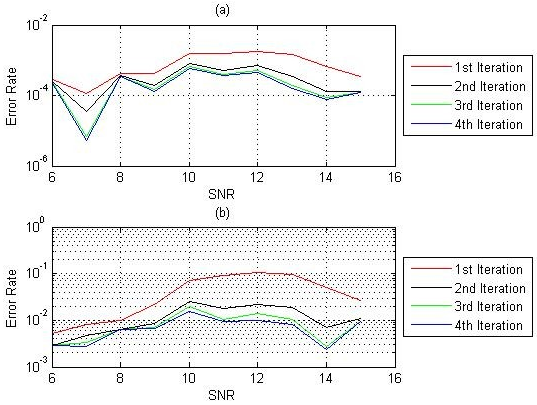


Figure 6.4: Water-filling. (a) BER (b) WER

In the above figures, we observe that after each equalizer iteration, the error rate decreases. Thus we could increase the performance of our system increasing number of equalizer iterations. Both figures show that after each iteration the increase of the performance is minor. Although this, the figures show that slight decreases of BER implies higher WER reductions. As we know, the channel will introduce errors that our joint equalization and decoding stage tries to correct. If the channel introduces a high number of errors in our transmitted codeword, our correcting system won't be able to extract the initial information. Thus we will have a codeword error event. While simulating the system we observed that sometimes we received codewords with a high amount of errors and sometimes with a few. In this last case, we know that increasing the number of iterations, usually, we will be able to extract the transmitted information.

Figure 6.3 and 6.4 show different evolutions of the error rate. For UPA allocation, the error rates are higher than for WF allocation. Thus the performance of the system is higher for WF than for UPA allocation. WF bit allocated packets require an average number of retransmissions lower than packets allocated by the UPA criterion, in order to be received by the receiver without error. Thus we can achieve higher throughput levels with WF than with UPA. We must remind that in UPA bit allocation we transmit the same amount of power in each sub-channel and that these channels won't have the same gain, thus the average error rate will be higher. We also must remind that sometimes the channel capacity will be very low, thus we will be transmitting over the channel capacity which will increases the error rate as well.

6.3.2 Comparison between the throughput obtained with UPA and WF bit allocations

Figure 6.5 shows the BER (a) and WER (b) obtained at the output of the correcting stage. WF allows us, systematically, to obtain lower WER after this stage. Thus we will require to retransmit less times the packets in order ensure that they information is free of error. So the throughput of the system will be higher for WF than for UPA bit allocation criteria, in our system.

Figure 6.5.c shows how the bit rates evolves as the SNR increases. First, we note that the bit rates are higher, generally, when we use the WF bit allocation criterion. Second, whenever the UPA transmission rates are higher than the WF bit rates, the WER is also higher. Thus the average throughput for UPA is lower than for WF allocation for all scenario.

Figure 6.5.c also shows that the flat fading channel capacity is lower than the AWGN capacity. Our

system transmits a lower rate than the maximum allowed by the flat fading channel.

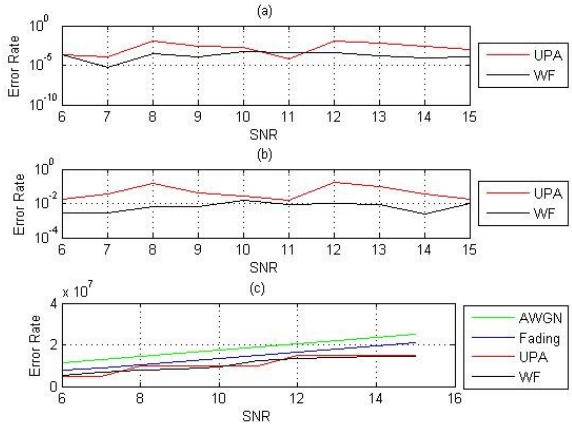


Figure 6.5: Error rate obtained after the joint equalization and decoding stage. (a) BER. (b) WER. (c) Bit rates obtained.

Finally, we observe that the rate of our system is constant for SNR higher than 12dB. Thus for SNR higher than 12 dB, a priori, there isn't any increase of performance when SNR is increased.

6.4 Conclusions

In this chapter we introduced the MMSE equalization method for SISO channels without ISI. We analyzed the simulation results obtaining an upper bound for the performance of our model.

We noted that the the performance of the WF bit allocation criteria is systematically higher than the UPA criteria but we can only use it when there is CSI information available

We showed that the WF bit allocation bit rates increases between 6 dB and 12 dB. After that, the transmission rate is constant. Thus we must analyze, at least, the performance of the system for SNR in this range.

We showed that increasing the number of iterations of the joint decoding and equalization method, the can reduce the BER and the WER of our system. Thus, we can increase the performance of our system just increasing the number of runs of both, the decoder and equalizer, or of each one per separate. This will lead to an increase of the computational load. We saw that slight reductions of the BER generates higher reductions of the WER. Furthermore, for similar BER the WER is not the same. Thus we will follow the design criteria of keeping the WER under a given bound.

Finally, we note that we can decrease the error rates by increasing the decision levels to map our systems' constellation into the data streams. Thus the rate will decrease. We couldn't afford the computational load of the optimization process to obtain these decision levels which, a priori, seems that are different for each SNR level. Thus we set the same levels for each SNR just trying to keep the WER under a given value.

7 Analysis of the Synchronization Errors and the Water-filling Algorithm.

In this chapter the errors introduced by the remaining synchronization error are characterized and the effect introduced in the average throughput of the system as well. First, the errors introduced by the channel impulse response, the frequency offset and the frame start sample are characterized. Second, the optimal values of the system's configuration parameters are discussed. Third, the performance of the proposed synchronization scheme and the WF module are exposed. Finally, the resulted obtained after simulating the system's performance are analyzed.

7.1 Simulation Error Sources

In this section the kind of effects introduced in the transmission system by the remaining synchronization and channel estimation errors are exposed.

7.1.1 Error introduced by the frame start timing synchronization error event.

As as first approach, the proposed system believes that we are transmitting the same sequence continuously because when the last samples of the signal are appended at the beginning of the OFDM symbol (the CP). The ISI produced by these samples is introduced in the useful part of the symbol, somehow we are cheating the receiver. Thus, the N transmitted samples of the OFDM symbol are received L times but with different phases and amplitudes (the phase and amplitude of each channel tap).

The amount of energy received per sample is given by equation (7.1). Whenever the optimal sampling position is chosen, the maximum amount of energy transmitted by the channel will be received as the contribution of all the taps of the channel will be present in the IFFT symbol window.

$$E_i = \sigma_{s_i}^2 \cdot T_s \sum_{l=1}^{L} |h_k|^2 \quad (7.1)$$

Generally, the amount of energy received will decrease whenever a wrong optimal sampling position is selected, but furthermore the interference level will be increasing because this energy will be received by other OFDM symbol or interpreted by our current symbol as an interference or noise. Thus timing synchronization errors decrease the performance of our system.

In figure 7.1 we see the shifts produced when there is a timing error event. Whenever we select a CP sample instead of the optimal sampling position (at the left of the optimal position), we have a channel shift as shown in Figure 7.1.b (the channel is shifted to the right). Figure 7.1.c shows what happen when the selected frame start position is at the right of the optimal position (the channel is shifter to the left).

This last kind of shifts are easily corrected by the fine timing estimator because its energy level is, usually, higher than when a CP sample is selected. Anyways, the amount of interference allowed in our system can be controlled by increasing or decreasing the power level required to estimate that there is a

channel tap at the end or after the sample L of our channel estimation. Thus we remind that whenever we set a low value, the chance of introducing distortion in our channel impulse response estimation increases.

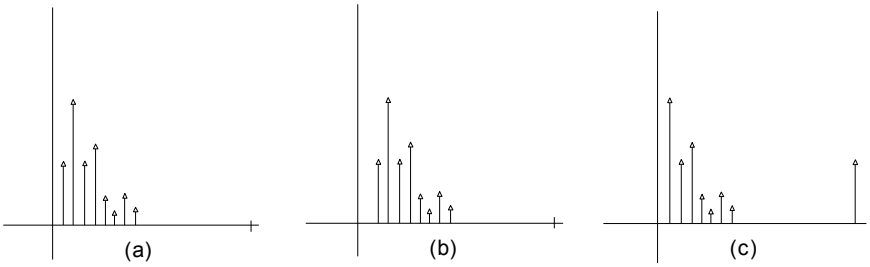


Figure 7.1:Channel Impulse Response. (a) Perfect timing. (b) Positive timing error. (c) Negative timing error.

Whenever there is a timing error event, one tap in our channel impulse response estimation will be missed, at least,,introducing a noise tap instead of a real tap. Thus the total amount of received energy decreases meanwhile the channel distortion and noise level increase. This effect can be modeled by means of the signal to noise and interference ratio as in (7.2), supposing that the distortion introduced by the noise is very small.

$$SNIR = \frac{\sum_{k \in L_{detect}} |h_k| \cdot \sigma_s^{\,2}}{\sum_{k \in L_{not\,detect}} |h_k^{\,2}| \sigma_s^{\,2} + \sigma_n^{\,2}} \quad (7.2)$$

In equation (7.2) we see that for high SNR levels, the SNIR will depend only of the quotient of the estimated channel taps and the missed channel taps. Usually, this value will be very high, as our energy level will be set slightly above the noise level. If the tap with the highest energy level has a power of 0.49 (0.7) and we set the minimum energy level to the 0.001, if we miss a tap, for a high SNR level, we will have at least, an SNIR of 25dB.

7.1.2 Errors introduced by the remaining frequency offset.

As a first approach, this effect can be modeled as a cosine multiplied to the received signal. Thus, we won't appreciate any effect over the frame if $T_s \cdot Af \cdot n \ll 1$ because the value of this cosine will be almost 1. Though the value of n will be increasing which will attenuate the received signal, increasing the error rate. The real effect of the remaining frequency offset is given by (7.3). It will introduce a progressive shift of the received constellation symbols which effects will increase in every OFDM symbol as n increases. Thus,for long frames or high frequency offsets, the data streams of the last OFDM symbols will be, systematically, shifted and we won't be able to correct this effect by further processing.

$$y(t) = x(t) \cdot e^{-j \cdot 2 \cdot \pi \cdot Af \cdot T_s \cdot n} + w(n) \quad (7.3)$$

The easiest method to control its effect is decreasing the number of transmitted OFDM symbol in each frame. This shift can also be estimated because reliable symbol estimations are obtained after the equalization and decoding stage of the information data received in the beginning of the frame, for low frequency offset values.

$$(m \cdot \sigma_{offset} + \mu_{offset}) T_s \cdot N \cdot N_{symbols} < \frac{\varphi}{2 \cdot \pi}$$
$$N_{symbols} < \frac{\varphi}{2 \cdot \pi \cdot (m \cdot \sigma_{offset} + \mu_{offset}) T_s \cdot N} \quad (7.4)$$

If one choose to control the frame length, one can find an upper bound for the frame length through equation 7.4, where $N_{symbols}$ is the total amount of transmitted symbols by frame, μ_{offset} and σ_{offset} the mean and the standard deviation of the frequency offset for a given SNR level, N the number of sub-channels in each OFDM symbol, T_s the sampling time, φ the maximum shift allowed by us in our system and m a control parameter to set the reliability of receiving a frame without the frequency offset effects.

7.1.3 What does “perfect CSI” stand for?

During the synchronization stage, the channel impulse response is estimated. Thus, as in any estimation process, we will be introducing some error. This estimation can have three kinds of errors:

- As showed in section 7.1.1 that a timing synchronization error event introduces distortion in our channel estimation as well. Because of this, whenever we have this kind of event the equalizer will be fed by erroneous data and the performance of the symbol's estimation process will decrease.

- The received training symbol samples have noise contribution. Thus, the reliability of the channel estimation increases as the SNR increases because this noise contribution will decrease. Then, this error can be modeled, as a first approach, as another AWGN source that, generally, will have similar statistics that the received noise.

- It won't be possible to perform the WF bit allocation because, for low SNR level, the frequency impulse response of the system will have low reliability and we could allocate high energy levels to data streams with a high level of attenuation. Thus the system's error rate increases and the average throughput of the system decreases because more data retransmissions are required to transmit our data packets.

While simulating the system, it was observed that, for SNR levels of 17 dB, the reliability of estimated CSI allowed to receive data streams carrying QPSK symbols. Next, the SNR level of the data OFDM symbols was reduced, keeping the value of 17dB for the training symbol, until 6dB. We found that it was the minimum SNR level required to receive data streams with BER under 10^{-4}. After that, a simulation system was set up to find the reliability level required to receive free error data. The bit allocation given by the WF module to transmit the data streams was used and the training symbol's SNR changed until the results shown in table 7.1 were obtained.

SNR level	Constellations	Channel Statistics	
		Mean	Variance
>17dB	QPSK	1.0e-2	2.9e-4
>20dB	QPSK, 16QAM	6.4e-3	1.4e-4
>30dB	QPSK, 16QAM, 64QAM	3.2e-3	4.5e-5

Tabla 7.1: Reliability required in the CSI to receive a given constellation.

7.2 The Simulation System.

In this section how one can configure the parameters of each module will expose.

7.2.1 The channel estimator.

As we said in section 3.3.3, we can set the minimum energy level that our symbol recognize as a tap. We must remind that if we set a low value for this variable we can increase the error distance instead of correcting it. Thus we must be careful when deciding it. We can control the maximum ISI level allowed in our system by limiting the timing synchronization events, as we state in section 7.1.1

7.2.2 The Water-filling module

In the proposed system, there are 3 different constellations available to map the data streams of each sub-channel. Thus it isn't possible to allocate optimally the parameters given by the WF module. Although this, it will be shown that it is possible to transmit at high rates controlling the codeword error rate.

There are different strategies to optimize this values. E.g. one could set different decision levels for each SNR. Due to the complexity of the system, it is unaffordable to optimize these values by this method. Thus an error rate was fixed, $WER \ O(10^{-2})$, and a search of the levels that allow us to obtain it performed. The error sources of this system were the AWGN, the flat fading channel and the remaining frequency offset. With this method, the constellation decision levels shown in table 5.2 were obtained. With these values, the system will be working, locally, close to the channel capacity.

An approach to reduce the BER of our system is increasing the energy levels required to transmit a given constellation. It doesn't implies, necessarily, a reduction of the average throughput of the system. E.g. Alice would like to send some IP packets to Bob, of 1500 bits each one. Alice's transmitter will encode each IP packet into three codeword of length 1000. Thus, if our system has a WER of 0.1 it will require, in average, 1.37 IP packets to transmit it. If we reduce the WER, to 0.05, by increasing slightly the power required to transmit a given constellation, the rate will decrease but we will require to transmit each IP packet only 1.17 packets. Thus the transmission will be more efficient and sometimes the average amount of transmitted information bits will increase.

7.2.3 The joint equalizer and decoding module

In Chapters 4 and 6 was showed that we could easily increase the error correction capacities of the decoding stage increasing the number of decoding or demapper iterations but increasing the complexity of our system. The proposed system performs four equalizer iteration and five decoder iteration per each equalizer iteration obtaining an average target WER, $WER \ O(10^{-2})$, to ensure that it will be required to transmit the information at most 1.02 times, when considering the transmission of IP packets.

7.2.4 The frame length.

A frame with a given maximum size is proposed to ensure that the frequency offset effect doesn't corrupt the last OFDM symbols of our frame. In the table that follows we can see the selected frame lengths for each SNR.

Frame Length		Parameters			
Optimal	Selected	SNR*	Maximum Error	Standard Deviation	m
3,37	5	6	π/8	483	1,5
4,26	7	7	π/8	382	1,5
5,41	8	8	π/12	301	1
6,81	8	9	π/12	239	1
8,66	8	10	π/12	188	1
7,23	8	11	π/12	150	1,5
6,84	7	12	π/24	119	1
8,57	7	13	π/24	95	1
10,85	10	14	π/24	75	1
13,56	10	15	π/24	60	1

Table 7.2: Frame length chosen for each SNR

The maximum shift allowed is one half of the phase difference between a given symbol and its neighbors. The m is a parameter to ensure that the last OFDM symbols of each frame, usually, won't be systematically shifted. A value of one stands for a frequency offset value that the 70% percent of the times will be lower than the standard deviation value. The training symbols are transmitted with the double (in dB) power than the symbols carrying information. E.g. whenever data streams are transmitted with 6 dB of SNR, the average power transmitted is 4 times higher than the noise power, the training symbols will be transmitted at 12dB, the average power transmitted is 16 times higher than the noise level. Thus, for a data stream with 10dB of SNR, the average signal power 10 times higher than the noise power, the training symbols are transmitted with 20 dB, the average training symbol power will be 100 times higher than the noise power. So the table shows the SNR of each OFDM symbol carrying information and the statistics of the synchronization data obtained with a SNR two times higher. This will increase the performance of the timing estimator, as well. Thus, the simulations showed that, for SNR of 6 and 7 dB, there is present still a high standard variation level of the distance between the optimal and the selected sample position, as shown in figure 3.22. Thus it is expected that this effect will be shown in the final simulation, thus it could be analyzed.

Finally, the selected values for the frame length is higher than the optimal ones because it won't be filled completely with data. Thus the frame length changes after each channel estimation. It is known that the data received in the OFDM symbols in the beginning of the data frame will be more reliable, so one must try to dismiss the WER rate with this configuration. Although of this, when transmitting at SNR levels lower than 7dB, almost all the OFDM symbols data streams will be allocated with QPSK constellations and we know that OFDM symbols are required, at least, 4 to transmit codeword. Due to this, one decided to increase the frame length for this SNR levels although of increasing the WER, as well. For 8dB one decided to increase the maximum frame length in order to be able to transmit more codewords. One decided to keep constant the frame length between 8 dB and 11 dB to see the evolution of the system's performance for a constant frame length, although was known that more OFDM symbols could be transmitted. Between 12 and 13dB the frame length dismiss because one wanted to see which was the differences between frames with and without high probability of error. And last, one increased the frame length, for the last two SNR levels, because one knew that 64QAM symbols were going to be transmitted and one was interested to know if we the frame lengths could be increased keeping the WER under the decided value, thus maintaining the bandwidth efficiency level.

The reader must know that one could run this simulation just once because between 12 hours and 36 hours were required to simulate the performance of the system for a given SNR. One must note that the reliability of the last simulations is higher than the ones at the beginning because more bits are transmitted. But the goal was obtained: we have a gaze of the effects introduced by the WF and the synchronization module in our system.

7.3 The Simulation Results.

In this section, the results obtained when simulating the design system are presented and exposed. The performance of the overall system comparing the results presented here to the ones presented in chapter six is analyzed.

7.3.1 The simulation system

The simulation system is divided into two parts:

- First, a training symbol composed of 4 CCAOS sequences of 32 samples each one is generated. Second, a CP of 16 samples, that is the copy of the last 16 samples of the training symbol, is appended to the beginning of the training symbol. The training symbol is surrounded by two OFDM symbols with data and conveyed through the modeled channel. Next the error distance between the optimal and the selected training position and the rough frequency offset are estimated. Next the fine timing shift is estimated. Then the channel impulse response, in this last position, is estimated. Finally the remaining frequency offset is estimated.

- Second, one set up a transmission system where we generate the information OFDM symbols are generated, appending its CP to each one, and the frame conveyed through the real channel obtained at the optimal sample point. Next the remaining frequency offset as added to the frame. Thus, the channel and the remaining frequency offset effects are introduced in the frame. Next the system uses the selected frame start position discarding all the CP. Thus the timing synchronization error is introduced into the frame. These samples and the “channel estimation” are passed to the equalizer. This channel estimation is the real channel at the selected sampling position but whenever there is an error event in the timing synchronization process, a noise tap instead of the discarded tap is introduced. And last, the data streams are equalized and the estimated bits obtained.

One had a dilemma when choosing the CSI we were going to select to fed the equalizer. This option was chosen because the WF working range is between 6 and 13 dB. After that, it ever assigns the 64QAM to almost all the data streams. One also knew that the reliability of our channel estimation was enough to receive QPSK when the training symbol has 17dB. The main problem is that the remaining frequency offset and the timing error, for this SNR, were usually low. Thus we discarded to transmit the training symbols with this SNR level and instead we decided to fed the equalizer with an hybrid CSI: whenever there is a timing error event, its error is introduced in the CSI. If not, the system uses the perfect CSI. By this method, one can try to obtain data to analyze the remaining frequency offset and the timing errors.

7.3.2 Simulation Results

The system is simulated by means of Monte Carlo simulation. 10000 frames are transmitted per each SNR level. The system parameters are the ones given in section 7.1 and 7.2. The system followa the WF bit allocation criterion to map the bits to the given constellation in each data streams.

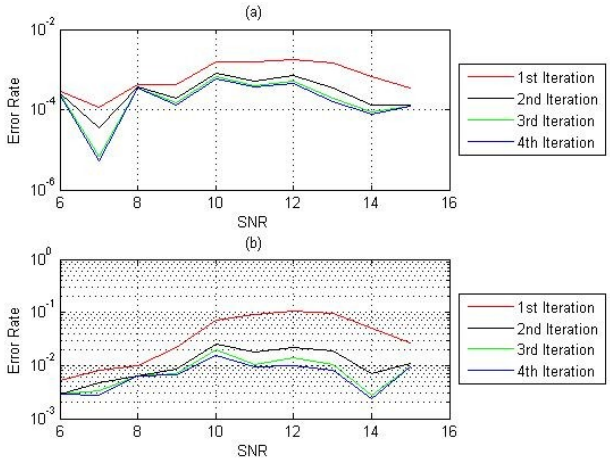


Figure 7.2: Ideal performance of the designed system. (a)BER (b)WER

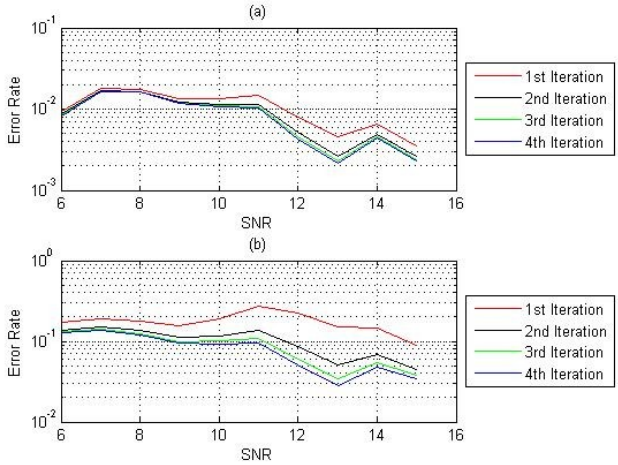


Figure 7.3: Obtained performance of the designed system. (a)BER (b)WER

Figure 7.3 shows the error rate evolution of the final system after iterations 1, 2, 3 and 4. The first point to note is that there isn't almost any increase of performance between each equalizer iteration for SNR lower than 8dB. The bit error rates obtained are almost two orders of magnitude above the of the ones of the ideal system. Though we obtain $WER\ \ O(10^{-1})$. We note that we designed the system to obtain $WER\ \ O(10^{-2})$, thus the WER increases one order the magnitude only after including the frequency offset and the timing error sources. For SNR higher than 11 dB, figure 7.3.b that the iterative correction process starts to work because we obtain an increase of the performance after each equalizer iteration. Finally it must be noted that although there isn't almost any increase of performance in the BER, for low SNR, an increase of performance by means of WER can be observed.

One must establish the framework to analyze what happens for each of range of SNR.

- 6 to 8 dB. If we remind what was said in section 7.2.4, the maximum frame length between a 30-40% higher than the optimal value. Due to this, we will have a high probability of receiving codewords with systematical errors. This kind of errors can't be corrected by interleaving the codeword but we could reduce them if we apply a block interleaver operation that carry mixed reliable and non reliable information. For these SNR we will receive, with high probability, ISI in our frames because the standard deviation of the timing error distance is close to 1. Thus, the CSI that is passed to the equalizer will have errors, with high probability, as well. Generally, we will be transmitting QPSK and 16 QAM symbols mapped into the data streams filling almost all the information OFDM symbols.

- 8 to 13 dB. One chose to fix the frame length for this range of SNR. The timing error distance standard deviation tends fast to the its asymptotic limit. Thus the reliability of the CSI increases with the SNR and the interference level decreases, as well. For the last SNR we will reach to asymptotic value of the standard deviation. The receiver will receive a mix of QPSK, 16QAM and 64QAM

constellation symbols mapped into the data streams. Thus, for SNR of 8 and 9 dB we will transmit 16QAM and a few QPSK and 64QAM symbols. For 10 and 11 dB the ratio of 64QAM symbols will increase as the number of data streams with 16QAM decreases. There will be a residual amount of data streams mapped with QPSK symbols. For SNR higher than 11 dB, 64 QAM symbols and some 16 QAM symbols will be, mainly, transmitted.

- 14 to 15 dB. The frame length was fixed to 10. Thus this value is lower than the optimal one, so distortion generated by the frequency offset should be lower. The timing error distance reached its asymptotic value, thus its effect is constant. 64QAM symbols and a residual number of 16QAM symbols will be transmitted. The CSI will be perfect almost all the time.

Analysis of the obtained results for SNR between 6dB and 8dB:

Figure 7.3(a) show the evolution of the BER after each iteration. In this range of SNR, the error rates are almost the same for each iteration. Thus the equalizer doesn't work properly. If we compare the 7.3(a) and (b) we observe that although the BER increases for the SNR levels of 7 and 8dB, the WER tends to decrease. Thus this increase in the BER is almost concentrated in the same codewords. This means that we will be receiving some word with a high number of erroneous bits and some others without. Thus some of the word will be very corrupted by the influence of the frequency offset and some others won't.

We know that data streams mapped with 16QAM symbols will start being transmitted at SNR levels higher than 7 dB. It seems that due to it, there is an increase of the BER for those SNR levels. We know that the 16QAM is less robust to the phase shifts than the QPSK modulation. Thus the effects of the remaining frequency offset are accentuated by the presence of data streams mapped with 16QAM modulated symbols.

We can't extract any conclusion about the effect of the timing synchronization error from from this data. One guess that it also increases the error rate of the data stream mapped with 16QAM, as the waterfall region of this modulation is around 7 dB. The effect of the timing synchronization error over the QPSK symbols is attenuated because the system is transmitting them much further from their waterfall region.

Analysis of the obtained results for SNR between 8dB and 13dB:

This range of SNR can divided into three parts. From 8 to 9dB, from 10 to11dB and between 12 and 13 dB. For the first span of values, figure 7.3.b shows that there is a progressive increase of the equalizer's performance. Thus the effect of the frequency offset decreases as the SNR increases. It seems that the performance of the equalizer is bounded to the effects of the remaining frequency offset.

For the SNR between 10 and 11 dB, the system starts to map 64QAm symbols into the data streams. The figures show an slight increase of the error rates in the first iteration but after the equalization process we obtain the same error levels than for the rest of the previous SNR. Thus the equalizer can correct this initial increase of errors caused by mapping the 64QAM modulation symbols into the data streams. It must be noted that the optimal frame lengths obtained for these SNR levels are around the selected ones. Thus we see that the selected values, for the reliability of the transmission and phase shift error allowed, were properly assigned because the average error rate kept constant.

For the span of SNR between 12 and 13dB the maximum frame length value was set to seven, a value slightly lower than for the others SNR of this group, eigth. We see that the equalizer has better performance than for the others SNR simulated. It seems that the amount of systematical errors is lower, enabling a more optimal equalization process.

Analysis of the obtained results for SNR between 14dB and 15dB:

A higher value for the maximum frame length was set in this span, ten. The chosen values are below the optimal one, the first one a 9% less and the second one about a 30%. We observe that the BER and the WER increases for a SNR of 14 dB and decreases again for a SNR of 15dB. Again, the effects of the frequency offset are increasing the error rates. Although of this, we observe that the obtained WER is around the same values than for SNR of 12 and 13 dB.

Analysis of the throughput of the system:

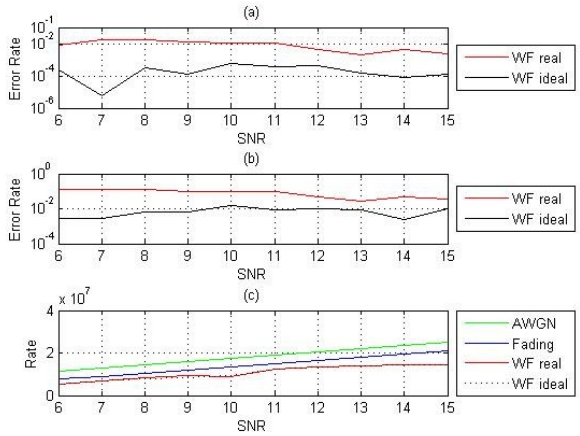


Figure 7.4: (a) BER, (b) WER, (c) Bit Rate of the system

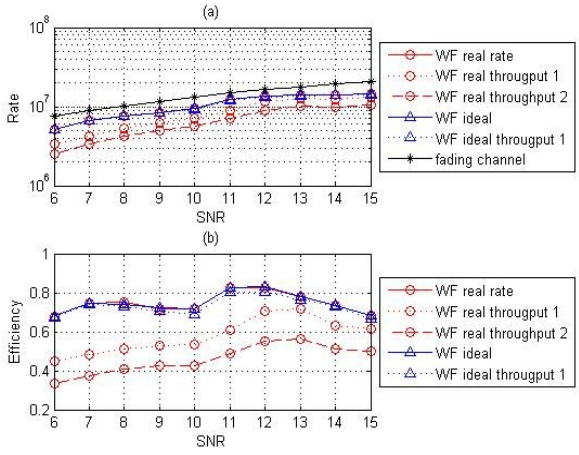


Figure 7.5: (a) Bit Rate, (b) Efficiency of the system

In figure 7.4.c we depicted the average rates of the fading channel, our ideal and real simulation system. In figure 7.4.a and 7.4.b the final BER and WER are shown respectively in order to help to analyze the obtained results. The simulated bit rates of our real and ideal system are, theoretically, the same. We can see that they diverge slightly for some SNR levels, but the differences are not remarkable. We see, in figure 7.4.c, that the bit rates of both, the ideal and the proposed system, increase as the fading channel capacity increases. Thus the WF bit allocation is increasing step by step the bit rates of our system. For SNR higher than 12 dB the bit rate of our system doesn't increase anymore, as it reach the maximum Shannon capacity bound for the 64QAM with rate ½, 15 Mbps.

Figure 7.5.a shows the evolution of the raw bit rate of our system. Ideally, it should evolve linearly until it reach the maximum capacity but it doesn't. This effect was caused when one decided to reduce the bit rate and the WER by selecting a high decision level to start to map the 64QAM constellation. Thus this plot depicts how this effect decreases the bit rate for a SNR of 10dB.

Figure 7.5.b shows the efficiency of our transmission system. The plot labeled by “WF real rate” was obtained dividing the average bit rate of the system by the average fading channel capacity. Thus, the proposed system has a rate between the 65 and 85% of the maximum allowed to ensure the free error

transmission condition. The plot labeled by “WF real throughput 1” is obtained taking into account that the system has an error rate different than zero (see equation 7.5). We observe that, in this case, the system efficiency oscillates between the 50 and the 70%. Finally, the plot labeled by “WF real throughput 2” is obtained taking into account that we are not transmitting all the time information OFDM symbols, (see equation 5.2). In this case, the efficiency of our system oscillates between the 35% and the 55%.

$$\eta_2 = \frac{1}{\overline{N}_{tx}} \qquad \overline{N}_{tx} = 1 + \frac{P_e}{P_{ok}} \tag{7.5}$$

SNR	Bit Rate	Efficiency 1	Throughput 1	Efficiency 2	Throughput 2	PER	#retransmission
6	5,273 Mbps	0,662	3,490 Mpbs	0,747	2,605 Mbps	0,662	0,511
7	6,799 Mbps	0,643	4,373 Mbps	0,784	3,428 Mbps	0,643	0,554
8	8,438 Mbps	0,681	5,744 Mbps	0,796	4,574 Mbps	0,681	0,469
9	9,297 Mbps	0,739	6,873 Mbps	0,796	5,478 Mbps	0,739	0,353
10	9,102 Mbps	0,746	6,788 Mbps	0,796	5,405 Mbps	0,746	0,341
11	12,617 Mbps	0,738	9,309 Mbps	0,796	7,416 Mbps	0,738	0,355
12	13,473 Mbps	0,856	11,539 Mbps	0,784	9,044 Mbps	0,856	0,168
13	13,950 Mbps	0,917	12,792 Mbps	0,784	10,027 Mbps	0,917	0,091
14	14,242 Mbps	0,862	12,271 Mbps	0,814	9,997 Mbps	0,862	0,161
15	14,423 Mbps	0,900	12,981 Mbps	0,814	10,572 Mbps	0,900	0,111

Tabla 7.3: Bit rates achieved at each SNR level.

In figure 7.5.b we can observe the different behavior of the plots labeled by real WF. We observe that the throughput 2 kept at increasing for SNR between 11 and 13 dB although that the raw efficiency of the system decreases. Here we can observe that the reduction of the WER though setting a shorter frame length can increase the efficiency of the transmission system. One also note the evolution of the efficiency between the SNR levels of 6 and 9dB in the plots labeled as raw and throughput real WF. Although that the channel efficiency decreases for the SNR 8 and 9dB, the WER was kept approximately constant and the maximum frame length increased which produced an increase of the efficiency of the final throughput.

7.4 Conclusions

In this chapter one exposed the simulation performance of the designed transmission system. It was shown how the remaining estimation errors reduce the performance of our ideal design. The effects that the configuration parameters of our system have on the final performance of the system were showed and discussed. Thus, one shown how these parameters can be optimized in order to increase the average performance, the efficiency, of the proposed system system.

One showed that the election of the border SNR levels to allocate the different labels has an effect in the error rate and the efficiency of the system.

We saw that the presence of systematical errors, introduced by the remaining frequency offset, decrease the performance of the equalizer. In the proposed system, a higher bandwidth efficiency level can obtain by increasing the reliability parameter of the equation (7.4).

It was shown that the bit rate of our system reach its maximum level for a SNR of 13. Though the

average throughput of our system can, still, be improved by increasing slightly the frame length of our system.

It was also shown that we must be careful whenever we start to map the data streams' information into a new constellation label. We must reduce the maximum shift allowed or increase the reliability of the estimation, all aimed to keep controlled the errors produced by the remaining frequency offset.

The reasons which produces that sometimes WER can decreases while the BER remains constant were also discussed. The difference between the WER and the BER evolution gives us some clues of how the errors are distributed in received codewords. Thus WER parameter is more sensible to the average distribution of the errors and the BER to the average number of received error.

8 Conclusions and Future Work.

8.1 Final Conclusions

In this thesis, one proposed and introduced some timing and frequency offset synchronization methods and a synchronization scheme based on one proposed by Minn. One also proposed a method to estimate the channel impulse response.

We observed that the performance of the different timing estimators is bound to the chosen sequence to constitute the training symbol. Generally, for low SNR levels, the variance of the training sequences formed by pseudo-noise sequences is lower than the ones formed by constant amplitude sequences but the mean is further from the optimal sample point. Because of this, generally, the sampled estimated by the constant amplitude sequences will be closer to the optimal sample. As the SNR increases the performance of the pseudo-noise sequence increases because the distance between the optimal and the selected sample decreases, for both fading and AWGN channels. In this case, its performance is better than the constant amplitude sequences.

One proposed a new training symbol that enables further data processing which allows us to obtain an improvement in the overall performance of the synchronization system.

We saw that there is a lower bound for the timing estimation distance in fading channels. This is caused because in fading channels the average SNIR, without any previous processing, is usually about 2 or 3 dB remaining constant respect the SNR because of discarding paths corresponding to interferences. We proposed a new method aimed to correct this affect. Though this method, one obtained an unbiased estimator reducing the standard deviation around 14 dB for high SNR levels and obtaining a less extent result for lower SNR levels.

Due to the characteristics of the training sequence that we employ, we can combine the SC and the M and M frequency offset estimators. Thus the obtained estimation range is bound to L and it has the same precision than the SC estimator, without almost any increase in the computational cost because the last one can be obtained from the data obtained when calculating the timing estimation.

Although that the normalized mean squared error of the obtained channel estimation is about 10dB lower than our first estimation, we found that we still require a more reliable channel estimation to enable an optimal equalization process.

We saw that the performance of the proposed methods are determined by the SNR, the selected training sequence and to the selected training symbol structure itself. SNR influences heavily as in the performance of frequency offset and CSI estimators. Transmitting all the available energy during the training symbol transmission time will improve the overall performance of the transmission system. However we need to take into account that the interference level should be kept under control in order to avoid generating interference to other systems which might be using adjacent bandwidth.

As one showed, the remaining frequency offset is and upper bound for the length of our transmitted frame. On the other side, we have a parameter to control the ISI introduced in our system, the decision levels of the channel estimation. As one commented, the maximum ISI level allowed in our system is bound to the distortion normalized mean square error of our channel estimation. So there is a trade off between these parameters. We must keep in mind that these missed small taps, for low SNR, will almost be composed by noise. Thus selecting them, we will be introducing distortion in our channel estimation as well.

The performance of the proposed scheme will increase whenever longer training sequences are used. We know that the performance of all the synchronization process depend of the training sequence. By increasing its length, we will obtain a more precise estimation of the timing error, the frequency offset and we will reduce the mean squared error error of the channel estimation. Thus, the ISI level of our system will decrease and the remaining frequency as well. We know that the maximum frame length is bound to the standard deviation of the frequency offset. So, by reducing it, we can increase the number of OFDM transmitted in each frame, the efficiency. We also will be able to obtain a more reliable channel estimation and thus we will be able to obtain a more precise fine timing estimation. Thus, the overall performance of the system will increase as the efficiency of the system increases by reducing the average number of required packet retransmissions.

One showed that increasing the number of iterations of the joint decoding and equalization method, we can reduce the BER and the WER of our system. Thus, we can increase the performance of our system just increasing the number of runs of both, the decoder and equalizer, or of each one per separate. This will lead to an increase of the computational load. We saw that slight reductions of the BER generates higher reductions of the WER. Furthermore, for similar BER the WER is not the same. The difference between the WER and the BER evolution gives us some clues of how the errors are distributed in received codewords. Thus WER parameter is more sensible to the average distribution of the errors and the BER to the average number of received error.

We saw that the presence of frequency offset, introduces systematical errors that the equalizer can't correct.

One obtained the flat fading channel capacity expression for Gaussian distributed signals and after an optimization process, Water-filling algorithm. After that. the average SNR of each sub-channel and the noise of each one were estimated. Thus, now we can obtain the noise level of each data stream. We require to introduce somehow that we are not allocating uniformly the energy whenever we equalize. Thus one decided to add this effect in the noise variance of each sub-channel because the computational load will be lower than introducing it directly when obtaining the statistics of the received symbols.

The solution when transmitter knows perfectly the channel is Water-Filling, a procedure which establishes that, to obtain capacity is necessary to distribute power keeping constant the sum of this times the inverse of channel gain, thus this solution assigns more power to the best channels. Also one introduced that UPA is the optimal bit allocation when there isn't a perfect CSI. We observed that the average rate BER when we follow the WF strategy is higher than for UPA.

The WF algorithm enables the optimization of the available energy in order to achieve the flat fading

channel capacity. Although this is the upper bound of the achievable throughput, sometimes we can transmit above this capacity level. It happens in our system because the system doesn't have enough labels to map properly the WF results. When it happens, the transmitted information will have a higher error probability, though the BER will increase. The most important parameter that we must control, when setting mapping levels, is the average transmission throughput because we will be transmitting at rates around the capacity, sometimes slightly higher and sometimes much lower, but the average must be kept under the average channel capacity. We showed that the bit rate of our system reach its maximum level for a SNR of 13. Though we can still improve the average throughput of our system by increasing slightly the frame length of our system.

Finally one showed that we require a cross-layer design in order increase the efficiency of the transmission systems. The performance of all the modules that compose the system is bounded. Thus we shall design the system on the whole. The configuration of all the parameters of the system must be aimed to maximize the efficiency of our system, by increasing the performance of the error correcting stage, increasing the bit rate of our system or decreasing the average time that we are transmitting synchronization symbols. Thus we must control the effects introduced by the remaining frequency offset by means of the equation (7.4). There is a trade-off when choosing the value of the reliability in equation (7.4), if we increase m the WER decreases but the efficiency of the system as well. Thus we know that we can obtain higher throughputs when we increase it because of the reduction of the average number of retransmitted packets. We must be careful when we start to map the data streams' information into a new constellation label. We must reduce the maximum shift allowed or increase the reliability of the estimation, all aimed to keep controlled the errors produced by the remaining frequency offset.

Summarizing one designed and simulated a transmission systems. After the simulation stage, one obtained throughputs between 3Mbps and 10Mbps. Assuming the approaches one did in the last section of chapter two, we think that it works. One can still improve its performance by optimizing the parameter 'm' or the decision levels to map each constellation but we think that we reached our main goal. One developed a synchronization stage module and the WF bit allocation module and simulated them. Although one couldn't optimize the performance of the module due to the complexity of the system and the lack of time and adequate computing facilities, a first simulation model was obtained and it can be used as a basis for a further development of the system.

8.2 Future work

We must still develop some modules of the system to obtain a working system. The most important issue that we still must face is to obtain a channel impulse response with enough reliability. After achieving this goal, we can improve the efficiency of the system in different ways:

- We must obtain the optimal values for the WF decision borders.
- We can introduce an iterative process to estimate the remaining frequency from the previously decoded data. We have a data estimation and the channel estimation. Thus we can obtain the estimated data in time domain and obtain the remaining frequency offset by multiplying the estimated data and the conjugate of the received data. We must be careful when doing this process

because we can obtain some samples with wrong phases. Thus we should test the data before we estimate the remaining frequency offset, i.e. with SC method. Another approach to estimate the remaining frequency offset could be dome by introducing pilot tones in some sub-channels.

- We could also introduce an iterative process to obtain a more channel impulse response estimations. Also from the previously decoded data. We could try a probabilistic approach if we assume that the error between each sample of the channel estimation have the same distribution than the noise present in the system.

- We can introduce another decoder, thus we will have a broader spectrum of bit rates to allocate more optimally the WF results into the data streams.

- We can introduce modifications in the frame configuration. I.e, we can remove the CP and apply a MMSE equalization process for a channel with ISI. By this change, the efficiency of the system will be increased around a 10%. Thus we will need to add a pre-processing pos-processing matrix to diagonalize the channel in order to enable the WF bit allocation process.

- We can introduce different encoding-decoding layers, I.e. we can add a block data interleaver to spread more the errors between different codewords. By this method we can improve easily the performance of the system.

- If we would like to design a radio link modem we can introduce a beam forming module to cancel the received interfering signals.

- And finally, we can implement a MIMO equalizer.

9 References

[1] A. Pérez-Neira, M. Lagunas, Array Processing II: Spatial Diversity in Radiocommunications. June 2005. cttc resources.

[2] Hlaing Minn; Bhargava, V.K.; Letaief, K.B., "A robust timing and frequency synchronization for OFDM systems," *Wireless Communications, IEEE Transactions on* , vol.2, no.4, pp. 822-839, July 2003

[3] Schmidl, T.M.; Cox, D.C., "Robust frequency and timing synchronization for OFDM," *Communications, IEEE Transactions on* , vol.45, no.12, pp.1613-1621, Dec 1997

[4] Yaobin Wen; Danilo-Lemoine, F., "A Novel Postfix Synchronization Method for OFDM Systems," *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on* , vol., no., pp.1-5, 3-7 Sept. 2007

[5] Morelli, M.; Mengali, U., "An improved frequency offset estimator for OFDM applications," *Communications Letters, IEEE* , vol.3, no.3, pp.75-77, Mar 1999

[6] Ng, J.C.L.; Letaief, K.B.; Murch, R.D., "Complex optimal sequences with constant magnitude for fast channel estimation initialization," *Communications, IEEE Transactions on* , vol.46, no.3, pp.305-308, Mar 1998

[7] Chu, D., "Polyphase codes with good periodic correlation properties (Corresp.)," *Information Theory, IEEE Transactions on* , vol.18, no.4, pp. 531-532, Jul 1972

[8] Dejonghe, A.; Vandendorpe, L., "Turbo-equalization for multilevel modulation: an efficient low-complexity scheme," *Communications, 2002. ICC 2002. IEEE International Conference on* , vol.3, no., pp. 1863-1867 vol.3, 2002

[9] Tuchler, M.; Singer, A.C.; Koetter, R., "Minimum mean squared error equalization using a priori information ," *Signal Processing, IEEE Transactions on* , vol.50, no.3, pp.673-683, Mar 2002

[10] Wohlgenannt, R.; Kansanen, K.; Tujkovic, D.J.; Matsumoto, T., "Outage-based LDPC code design for SC/MMSE turbo-equalization," *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st* , vol.1, no., pp. 505-509 Vol. 1, 30 May-1 June 2005

[11] ten Brink, S., "Convergence behavior of iteratively decoded parallel concatenated codes," *Communications, IEEE Transactions on* , vol.49, no.10, pp.1727-1737, Oct 2001

[12] Gallager, R., "Low-density parity-check codes," *Information Theory, IRE Transactions on* , vol.8, no.1, pp.21-28, January 1962

[13] Shannon, C., "General treatment of the problem of coding," *Information Theory, Transactions of the IRE Professional Group on* , vol.1, no.1, pp.102-104, Feb. 1953

[14] M. Lamarca. "*Iterative Techniques for an Optimal detection*". Class notes. Fall 2007. UPC

[15] H. Poor, *An Introduction to Signal, Detection and Estimation*, 2nd ed. New York: Springer-

Verlag, 1994.

[16] Golay, M., "Complementary series," *Information Theory, IRE Transactions on* , vol.7, no.2, pp.82-87, April 1961

[17] Guangliang Ren; Yilin Chang; Hui Zhang; Huining Zhang, "Synchronization method based on a new constant envelop preamble for OFDM systems," *Broadcasting, IEEE Transactions on* , vol.51, no.1, pp. 139-143, March 2005

[18] Berrou, C.; Glavieux, A.; Thitimajshima, P., "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on* , vol.2, no., pp.1064-1070 vol.2, 23-26 May 1993