Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TRABAJO FINAL DE CARRERA

**TÍTULO DEL TFC: Adaptive Loading Algorithms for optical OFDM transmission systems**

**TITULACIÓN: Ingeniería Técnica de Telecomunicación, especialidad Sistemas de Telecomunicación**

**AUTOR: Norman Toporcer Korec**

**DIRECTOR: María Concepción Santos Blanco**

**FECHA: 13 de julio de 2011**

**Título:** Adaptive Loading Algorithms for optical OFDM transmission systems

**Autor:** Norman Toporcer Korec

**Director:** María Concepción Santos Blanco

**Fecha:** 13 de julio de 2011

Resumen

Actualmente, en la era de la tecnología, la demanda de grandes volúmenes de información precisa de una constante evolución de los sistemas de comunicación. Para realizar transmisiones de datos, existen diversas opciones para modularlos. De entre éstas, la multiplexación por división de frecuencias ortogonales, OFDM, está cobrando especial protagonismo por las ventajas inherentes que conlleva y, recientemente, se está iniciando su utilización en sistemas de transmisión ópticos. En este trabajo se documentan los aspectos más relevantes de OFDM empleado en fibras ópticas y se realiza la implementación, en un software de simulación para este tipo de medios, de técnicas y algoritmos adaptativos que pueden mejorar la eficiencia de las comunicaciones. Estas técnicas se basan en el ajuste del reparto de datos y de su nivel de potencia en la banda de transmisión considerando cuál es el comportamiento del canal.

**Title:** Adaptive Loading Algorithms for optical OFDM transmission systems

**Author:** Norman Toporcer Korec

**Director:** María Concepción Santos Blanco

**Date:** July 13, 2011

Overview

Nowadays, in the technological era, the demand of large volumes of information precises a constant evolution of the communication systems. To carry out data transmissions, several options exist to modulate the data. From all of them, the orthogonal frequency division multiplexing, OFDM, has proven successful in many applications due to its inherent advantages and, recently, its use in optical transmission systems is getting attention. In this thesis, the most relevant aspects of OFDM over optical fibers are documented and an implementation is done, in a simulation software for this type of physical mediums, of techniques and adaptive algorithms that may improve the efficiency in the communications. These techniques rely on the adjustment of the distribution of data and power within the transmission band considering how the channel behaves.

# CONTENTS

# LIST OF FIGURES

# INTRODUCTION

The orthogonal frequency division multiplexing, OFDM, is one of the most used modulations in wireless communications. Currently, OFDM is present in many fields of digital telecommunications such as wireless local and metropolitan area networks, LAN and MAN respectively, communications through the electrical grid, also known as power line technology, and the cellular mobile's fourth generation standard with the Long Term Evolution technology, LTE, and mobile WiMAX. The digital subscriber line standard, DSL, relies on the same principle as OFDM as well.

Optical transmission systems like fibers are growing in popularity because of their positive aspects: they can reach higher speeds and distances than other physical mediums and tend to be roughly constant in time, so they don't suffer relevant changes in their behavior. For this reason, optical networks have begun to experience widespread deployment offering higher data rates to users, and infrastructure and maintenance savings to operators.

As the next generation of optical networks is envisioned, significant advances in performance at lower cost are expected, which leads to many technological challenges to overcome. Recently, some research works have appeared trying to asses the potential of OFDM through optical networks [1].

Another important aspect of OFDM is the ability to adapt to the channel distortion thanks to the discrete multitone modulation, DMT, which is also the main working theory in DSL. This kind of modulation allows a straightforward channel correction with simple-tap equalization, provided subchannels are made narrow enough to be able to correct every subchannel using a single coefficient.

The purpose of this thesis is to study the use of adaptive loading algorithms as enhancement technique and its impact in an optical OFDM transmission system. An effective power loading and bit allocation algorithm may lead to a higher throughput.

This thesis has been done within the GCO (Grupo de Comunicaciones Ópticas) at UPC, which is involved in the ACCORDANCE STREP project. This project is composed by partners from Estonia, France, Germany, Greece, Spain and United Kingdom and investigates on a new paradigm for the access network: the introduction of OFDMA into a passive optical network architecture offering at the same time optical backhauling for wireless and copper-based networks.

Three are the main objectives pursued in this thesis. First, the principle and the most important aspects of OFDM through optical networks are reviewed due to its recent start in research and popularity [2]. The second objective is to program the routines needed to implement adaptive loading techniques and integrate them in a design that simulates an optical transmission system so the impact of the algorithms can be analyzed. And lastly, to make this code user-friendly. The main reason for this is that the simulation setups developed within this thesis can be used by researchers at UPC or even other partners within the ACCORDANCE project, and therefore special emphasis is placed on the user-friendliness of the code.

Therefore in this thesis some simulations are carried out testing different adaptive loading schemes and analyzing the results. The simulations are run in VPItransmissionMaker™ from VPIphotonics™, VPI from now on. This application allows to design a whole optical transmission system and gives capabilities to code user-defined functions and routines which can be included as fully functional blocks in a simulation.

The organization of this thesis is as follows:

In the initial chapter, the fundamental OFDM aspects are explained such as the theory behind the modulation, how the transmitter and the receiver work, and the overheads in OFDM symbols. There is also an overview of the state of the art in adaptive loading techniques.

The second chapter presents the optical modulation and demodulation techniques and the characteristics of the optical channel. The chapter ends with a presentation of the VPI software.

In the third chapter, a VPI template is studied to review the concepts explained in the previous chapters directly in the software. It is also used as a reference for designing a VPI platform that will support adaptive systems. The whole process is detailed.

The fourth chapter is focused on simulating the different adaptive systems. The results are analyzed and several comparisons are made to assess the improvements achieved.

Finally, in the last chapter, the main conclusions are summarized and some future lines of work are proposed for further development of the studies here performed.

# CHAPTER 1. OFDM BASICS

## 1.1. Introduction to OFDM

The orthogonal frequency division multiplexing, OFDM, is a modulation where the available spectrum is divided into many narrow-band subcarriers which are simultaneously transmitted with information symbols modulated in amplitude or phase per subcarrier. The whole data sequence of an OFDM frame is thus the sum of many frequencies which contain part of the information. Since data is transmitted in parallel, the symbol is made longer so that one symbol at most is affected by intersymbol interference, ISI (see Fig. 1.1) [3].

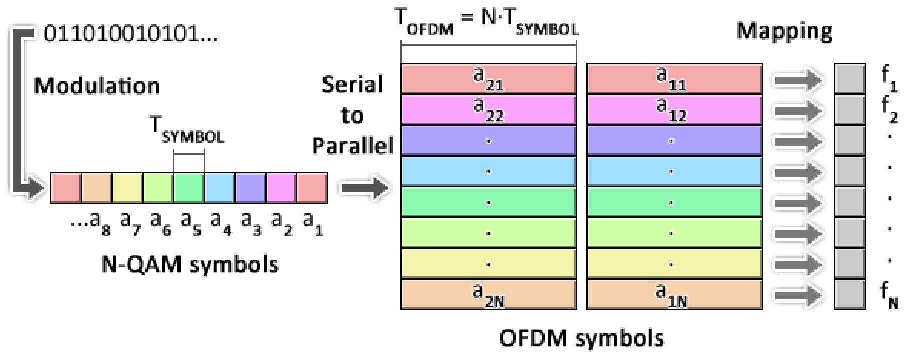


Figure 1.1: Data transmission in OFDM

In order to recover the information encoded into each of the subcarriers while at the same time optimizing the spectral efficiency, the orthogonality condition is required. That is that the symbol period is the inverse of the subcarrier spacing ($T_{OFDM} = 1/\Delta f$). Thus at each subchannel frequency, the whole FFT spectrum over a $T_{OFDM}$ time in each subcarrier frequency, all other subcarriers go to zero (see Fig. 1.2). And in the time domain, the orthogonality condition is understood from the viewpoint that the integral of the OFDM temporal signal multiplied by the targeted subcarrier function in the $T_{OFDM}$ interval is zero for all other subcarriers, thus providing the information in the targeted subcarrier.

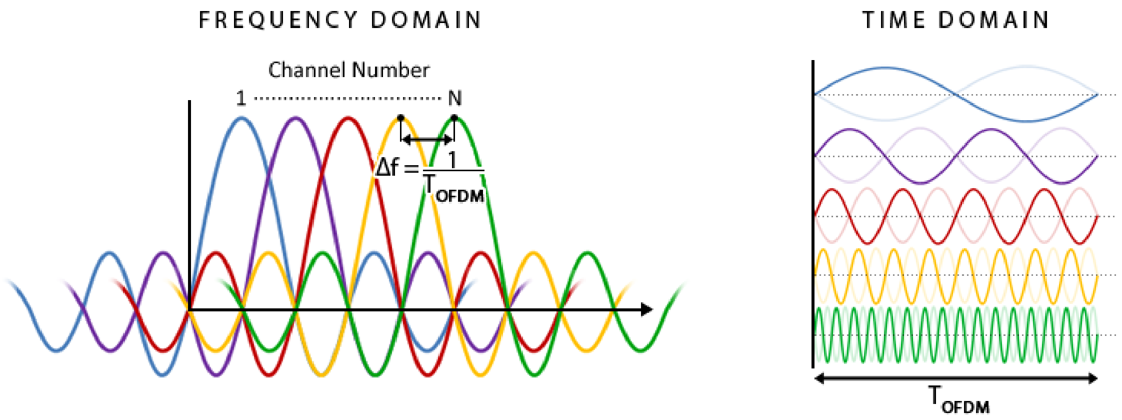


Figure 1.2: Subcarriers' orthogonality in the frequency and in the time domains

The transmission of many narrow-band channels allows the consideration of a constant performance for every subcarrier but different between them. A frequency selective distortion will only affect a few symbols with the rest being correctly decoded, while in single carrier systems, the whole sequence would result affected. On the other hand, with a very simple single one-tap equalization, the channel distortion can be compensated if the channel response is known. To estimate it, training sequences and pilot tones are used and with the information they provide, the equalization coefficients are calculated.

Another important feature of OFDM is that thanks to the periodical nature of the FFT, cyclical extensions of the OFDM symbol can be used in order to ease synchronization and to eliminate ISI from neighboring symbols.

Usually, OFDM symbols are generated in the digital domain, leaving only to the analog domain the transmission and reception of the signal. Therefore, it can take advantage of digital signal processing to form the frames. It is one of the main reasons why it is also used in wired communications, because it is easy to create the subchannels precisely and maintain their orthogonality.

The common practice is to generate OFDM frames starting in the frequency domain with the information modulated to each subcarrier digitally. Then, with an inverse Fourier transform, IFT, the temporal sequence is retrieved. As the whole process is digital, it is required to convert to an analog signal that can be transmitted. In reception, all the steps have to be reversed so, first, an analog to digital converter, ADC, is needed. Next, the Fourier transform has to be performed to recover the data sent. The whole process is explained deeper in the following sections.

## 1.2. Transmitting and receiving OFDM frames

### 1.2.1. Transmitter

Starting from a bit sequence that has to be sent, an OFDM transmitter (see Fig. 1.3) has to parallelize the serial bit sequence with as many outputs as subcarriers are set up. The parallel sequences have to be mapped with the corresponding modulation (in amplitude or phase) per subcarrier.

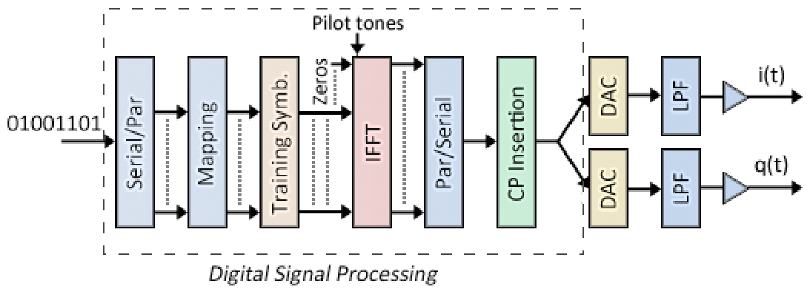


Figure 1.3: OFDM transmitter block diagram

A basic aspect to choose the modulation is the performance of each specific channel in terms of signal-to-noise ratio. Without the use of adaptive loading algorithms, all subcarriers are treated equally so the one with poorer performance would establish the highest modulation level for the rest. This imposition limits the amount of data that can be transmitted causing a loss in efficiency.

Prior to transforming from the frequency to the time domain, a training sequence should be added for synchronization in the receiver. It may be also used as channel estimation for the equalization. In case of enabling pilot tones, they have to be set up at this point.

As inverse Fourier transforms and Fourier transforms have to be performed, there is a condition that has to be fulfilled to improve the computational efficiency by using IFFT and FFT algorithms: the whole sequence must have a length equal to a power of 2. If this condition is not met, zeros can be added to the sequence in order to reach the required length. Both algorithms are low computational cost discrete Fourier transforms due to their high optimization. The great capabilities of the IFFT and FFT in such a small number of operations make them very recommendable for fast and low consumption devices.

Once the temporal responses are obtained, they are serialized to conform a single sequence. After that, the cyclic prefix is added. At this point, the sequence has to be converted from digital to analog domain with two DA converters, one for real part and another for the imaginary part. The output is low-pass filtered to remove spectral replicas and then conditioned for transmission over the channel.

### 1.2.2. Receiver

In reception, the electric voltages (see Fig. 1.4), which have been demodulated, are first low-pass filtered for anti-aliasing, and digitalized with an AD converter. Both branches, as they are considered in-phase and quadrature sequences, are added up to form complex numbers.

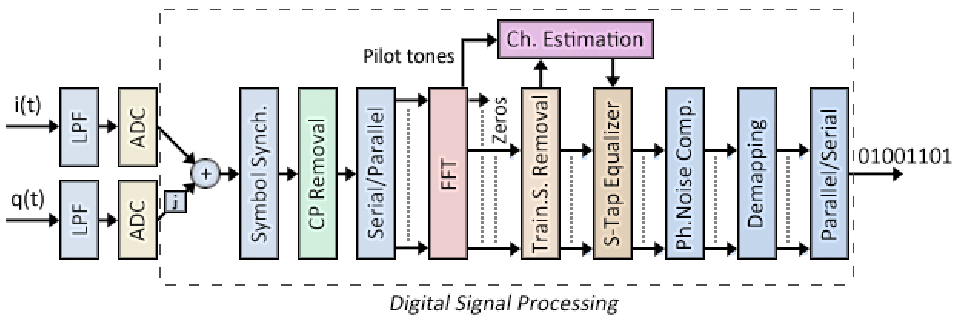


Figure 1.4: OFDM receiver block diagram

The next step is to synchronize the signal with the preamble added in the transmitter. When the synchronization has been achieved, the cyclic prefix is removed resulting in the recovery of the orthogonality. The obtained sequence is then parallelized so the FFT can be applied. With fast Fourier transform, the sequences in the frequency domain per subchannel are retrieved. If there were zeros added to the sequences when the OFDM frame was built, they have to be removed conveniently so they are not considered information.

Back in the frequency sequences, the training symbols added to each subcarrier are removed but used in the channel estimation module, and the same do the pilot carriers, to update the equalization coefficients. Then the single-tap equalization is done per subchannel. Consecutively, the phase noise can be compensated. After that, each subcarrier is demodulated with its own constellation and, at the end, the restored bit sequences are serialized to recover the information sent.

## 1.3. OFDM overheads

The OFDM modulation has different overheads that may be added to its symbols in order to prevent the effect of some undesired phenomenons that occur during the propagation of the signal.

A problem that OFDM has to face is the high peak-to-average power ratio, PAPR, on its signal if compared to single carrier modulations. This may lead to non-linear distortions generated when the electrical signal goes through the different devices that compose the transmitter and the receiver, such as amplifiers. Signal clipping and predistortion are techniques that may be used to reduce the PAPR levels [4].

The main OFDM overheads are now detailed.

### 1.3.1. Training sequences

At the beginning and after every specific number of OFDM frames, it is required to send some preestablished training symbols known in the receiver for its clock resynchronization. As the training sequence is well known, it can also be used in the channel estimation stage for an equalization intended to undo, as much as possible, the alterations produced by the channel's linear distortion (1.1).

$$Y(\omega) = X(\omega) \cdot H_{Ch}(\omega) \tag{1.1}$$

The received signal in the frequency domain, $Y(\omega)$, is obtained from the sent signal in the frequency domain, $X(\omega)$, and the channel's transfer function, $H_{Ch}(\omega)$. As the equalized signal, $Y_{Eq}(\omega)$, is obtained (1.2), the equalization is done by means of discrete coefficients so different complex values are calculated and applied in a per subcarrier basis.

$$H_{ChEst}(\omega) \approx \frac{1}{H_{Ch}(\omega)} \Rightarrow Y_{Eq}(\omega) = X(\omega) \cdot H_{Ch}(\omega) \cdot H_{ChEst}(\omega) \approx X(\omega) \tag{1.2}$$

Where $H_{ChEst}(\omega)$ is the estimated channel response, intended to compensate $H_{Ch}(\omega)$, and from which the equalization coefficients are calculated.

Training symbols should be sent periodically, the time interval between training sequences depends on how fast the channel dynamics change, so new and more precise coefficients can be calculated and used in the single one-tap equalization stage.

For the training sequence, a BSPK modulation tends to be used because of its high signal-to-noise ratio which helps to discern clearer the distortions originated by the medium. The BSPK also has a good PAPR level, which is recommended to remain almost unaffected to the non-linear behavior of the components present in the transmitter and the receiver. Then, with this modulation, the channel estimation is more precise.

### 1.3.2. Cyclic prefix

One of the main problems in most telecommunications is the appearance of undesired propagations of the transmitted signal because of reflections produced in the medium. These additional propagations, which may be considered as echoes with a phase lag comparing with the signal arriving more directly, with a gap in time and with possible loss or gain, may result problematic as in reception all signals are superposed preventing to just consider the one with the best SNR. This phenomenon is known as multipath effect because of many signals arriving to the receiver going through different paths.

Since there is no possible differentiation between the desired signal and the echoes in the receiving antenna, the altered replicas add a certain level of interference in the detection of symbols. This phenomenon, known as intersymbol interference, causes a temporal overlap of symbols during a time equal to the channel's impulse response.

This ISI can be avoided with a guard time between OFDM symbols. The guard time must be equal or longer than the channel delay spread [5] and in order to ease synchronization taking advantage of the FFT periodical nature, the symbol can be cyclically extended during this time.

In an optical transmission system, intersymbol interference is mainly due to the chromatic dispersion. This dispersion is related to the group velocity not being the same for the different subchannels, which is explained more in detail in the next chapter.

The cyclic prefix, CP, is the overhead used to avoid chromatic dispersion (and multipath effect too) that relies on copying part of the beginning or the ending of the OFDM symbol temporal sequence and adding it at the opposite side, either at the end or at the start (see Fig. 1.5).

This way, the period for each symbol is slightly increased. Consequently, the cyclic prefix introduces flexibility in the start time of data retrieval which helps to synchronize in the receiver at the expense of additional overhead. It also implies a loss of periodicity in the original signal: the period from a single symbol is a bit longer but when considering all OFDM symbols in a continuous form, the periodicity is not maintained (see Fig. 1.6). Hence the orthogonality is lost, to recover orthogonality the cyclic prefix guard time is removed at the receiver prior to the FFT decoding.

If instead of copying part of the temporal sequence the guard time is simply an empty guard period, the flexibility to retrieve the data is lost and, even if the problem of the chromatic dispersion is solved, intercarrier interference would occur because a perfect timing

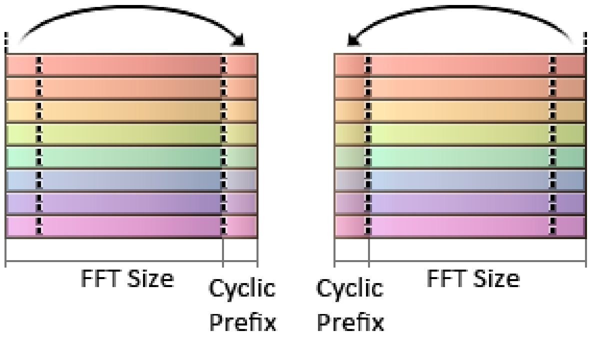


Figure 1.5: Cyclic prefix

is needed to get the symbol.

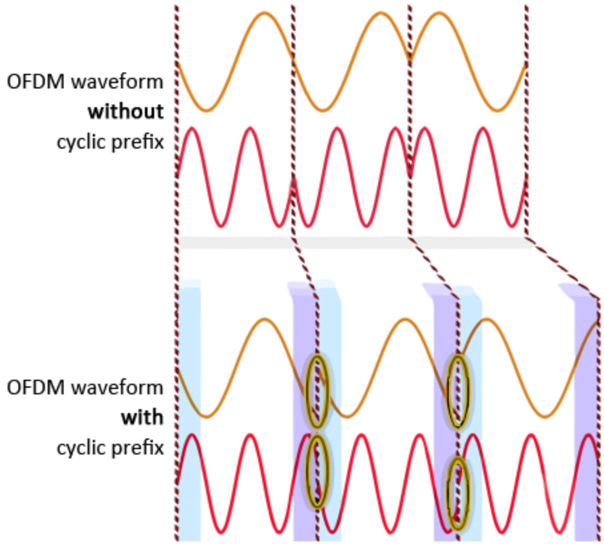


Figure 1.6: Loss of periodicity adding cyclic prefix

The guard time is a value fixed by the delay spread, so in order to make the cyclic prefix overhead smaller in terms of percentage, the OFDM symbol has to be longer by increasing the size of the FFT.

### 1.3.3. Pilot subcarriers

In OFDM, it is essential to know the exact frequencies of the subcarriers, thus it is very sensitive to frequency drifts. So, even though the subchannels are digitally generated with precision, they may be altered because when an optical signal propagates through a physical medium it can suffer frequency drifts that modify the spectrum.

In systems that suffer from such effects, the use of reference or pilot subcarriers is necessary. They are not used for data transmission. In fact, they are unmodulated tones (see Fig. 1.7) detected in reception to know if a variation in frequency has happened. If that is the case, it is considered that the subcarriers also were drifted and have to be corrected prior to their demodulation.

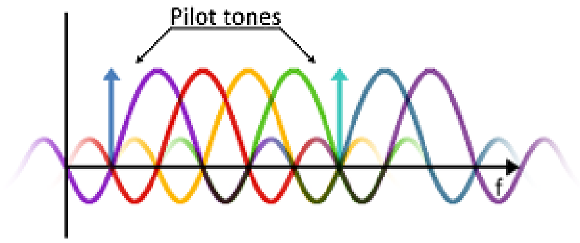


Figure 1.7: Pilot tones

Moreover, pilot subcarriers can be used in the phase noise compensation stage. To do this, the pilot tones have usually the neighboring suchannels disabled as guard band to avoid crosstalk from them. The pilot carriers contain the phase difference, so after filtering them, they can be used to compensate the phase noise [3].

The simulations carried out in this thesis do not use pilot carriers, just training sequences and cyclic prefix. It is considered that the optical transmission does not significantly suffer from frequency spectrum drifts nor has a remarkable phase noise.

## 1.4. Adaptive loading techniques

In OFDM, two different adaptive techniques exist that allow either to distribute the amplitude of the symbols from one or more subcarriers (power loading), or else to vary the amount of information that each subcarrier contains (bit loading). In this section, their basics are presented.

Adaptively loaded OFDM systems are highly efficient because they can optimize their data rates and improve the bit error rate in reception by taking the maximum profit of the channel behavior.

### 1.4.1. Power loading

With this technique, OFDM subcarriers have the amplitude of their modulated symbols modified. In a block diagram, the power loading module would be located right after the QAM modulation of subcarriers and prior to the IFFT (see Fig. 1.3) [6].

The principle is that some subcarriers may be amplified to have more power so they are less vulnerable to distortions, and its signal-to-noise ratio increases. However, any additional power used to amplify is taken from other subcarriers, so in average, the same total power is always used. In consequence, power subtracted subchannels are more vulnerable to fading and their SNR decreases. So a trade-off has to be established and that is why there are many power loading algorithms: they all try to maximize the efficiency of the system. Power loading can be considered a redistribution of power technique or a weight system where some subcarriers are given higher priority than others.

For effective power allocations, it is necessary to know how the channel behaves. With the so called channel state information, CSI, this is possible: transmitting well-known sequences, the signal-to-noise ratios, the error vector magnitudes and the bit error rates can be calculated. Such information should be enough to see if there is fading or interferences, and which subcarriers are affected.

The water pouring method [7], sometimes also referred as water filling method, is the most common way to have a variable power allocation. The goal is to have a graph that represents the channel response or the signal-to-noise ratio per subcarrier (sometimes their inverse is represented) so it can be seen easily how much affected (or not) are the subchannels by attenuations and fading. In the water pouring analogy, water (power) pours from better SNR subcarriers to poorer signal-to-noise subchannels so that all achieve the same water (power) level (see Fig. 1.8).

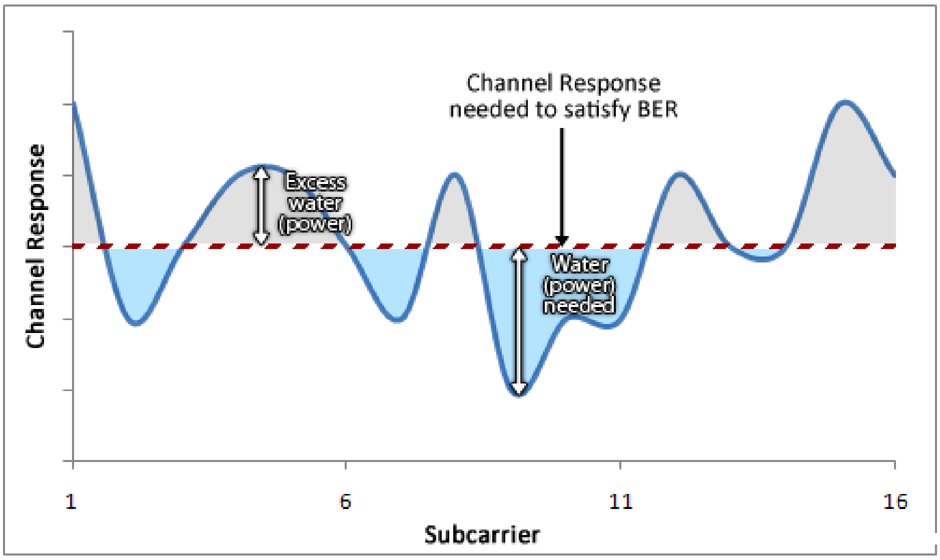


Figure 1.8: Water pouring concept

The algorithms which try to find the best possible distribution have to do many iterations testing different power allocations and retrieving the channel state information under those determined conditions. An important aspect they all consider is that no more than the minimum power that satisfies the preestablished constraint should be given to the subcarriers, so more spare power is available to be supplied to the rest [8][9].

### 1.4.2. Bit loading

One of the main issues that OFDM faces is that in any transmission, all the subchannels have their symbols modulated with the same constellation. This implies that the system sees its throughput limited by the poorest performance subcarrier or subcarriers. A reliable communication requires that the bit error rate of the system is below a certain value, so the subchannel with the lowest signal-to-noise ratio establishes the maximum modulation level which may be used to satisfy this requirement. Rising the modulation, even if the error probability in some subcarriers is low enough, would imply a BER increase. An option to avoid this limitation is to disable these problematic subcarriers.

For this reason, bit loading systems are useful: they add flexibility with high SNR subcarriers being assigned more bits to transmit than the ones with lower SNR, which are assigned modulations with less levels (see Fig. 1.9) [10].

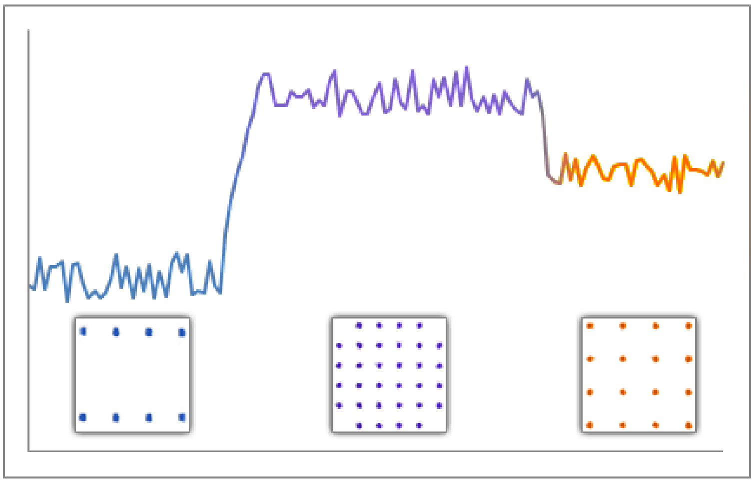

Figure 1.9: Different constellations used in an OFDM symbol

In an OFDM symbol, all the subcarriers must have the same temporal duration filling in a $T_{OFDM}$ symbol period. When using bit loading, this does not happen unless the subchannels with less allocated bits wait to have an equal duration (see Fig. 1.10), which is fixed by the subchannel with more bits assigned. This is because they all use the same digital-to-analog converter, so slower subcarriers see their symbols repeated.

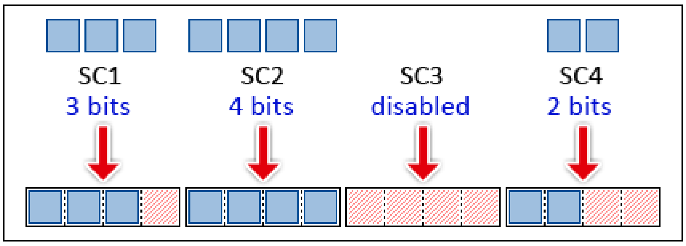


Figure 1.10: Space assignation with 4 subcarriers

Since the slow subcarriers are not taking profit of the full bit rate that system can provide, there is a loss in effective bit rate which is to be compensated by BER improvement.

There are similarities between bit loading and power loading. Both use the CSI to adjust their distributions and the water pouring method can be used for variable bit distributions too. The reasoning is the same, if the SNR of a subchannel positively surpasses the threshold, maybe it can modulate more bits, whereas having subchannels under the threshold would imply a less ambitious bit allocation.

The bit loading algorithms are based in several equations to find the most efficient distributions. They all share the starting point which is the Shannon–Hartley theorem (1.3) that specifies the maximum rate at which information can be transmitted over a communications channel of a specified bandwidth in presence of noise.

$$C = B \cdot \log_2 \left(1 + \frac{S}{N}\right) \tag{1.3}$$

A reworked version of the theorem, specifically intended for bit loading systems, is considered (1.4) [11].

$$b_i = \log_2 \left(1 + \frac{e_i \cdot CNR_i}{\Gamma}\right) \tag{1.4}$$

Where $b_i$ is the number of bits allocated to the subchannel $i$, $e_i$ is the power of the subchannel $i$, $CNR_i$ is the channel-to-noise ratio of the subchannel $i$ (the gain of the subchannel divided by the noise power in that frequency) and, lastly, $\Gamma$ is the SNR gap that indicates the required SNR increase so the subchannel is able to transmit at $b_i/B_i$, where $B_i$ is the bandwidth of the subchannel $i$. The SNR gap for a M-QAM with a BER constraint can be calculated as (1.5) [11]:

$$\Gamma = \left| \frac{1}{1.5} \ln\left(5 \cdot BER\right) \right| \tag{1.5}$$

These two expressions, which are good approximations in multicarrier systems (the SNR gap is obtained from a gap-approximation analysis based on the target bit error rate), are the basics for a bit loading algorithm. From these same equations, power allocations can be calculated too with the $e_i$ variable (1.6) [11].

$$e_i = \left(2^{b_i} - 1\right) \cdot \frac{\Gamma}{CNR_i} \tag{1.6}$$

The way the algorithms organize their iterations is not always the same, but two major procedures exist. The first one does not use formulas to distribute the bits as it starts allocating the bits corresponding to the highest possible modulation level given by the velocity of the system [9]. After transmitting the OFDM symbols, the subcarriers that are not below the given BER threshold, see their allocation reduced in one bit and then the cycle is repeated. This does not require any additional processing such as the channel state information or the use of equations, and it guarantees the maximum effective bit rate for a target BER, however it is a slow procedure as many iterations may be done.

The second major procedure consists in transmitting a signal to retrieve the CSI. With this information, the bits are effectively allocated using equations. For example, there is one really simple that allows a distribution depending on the signal-to-noise ratio and the target BER (1.7) [13]:

$$BER \leq 0.2e^{\frac{-1.5 \cdot SNR}{M-1}} \tag{1.7}$$

Which if operated, can be expressed as (1.8):

$$M \equiv 2^{b_i} = 1 - \frac{1.5 \cdot SNR}{\ln(5 \cdot BER)} \tag{1.8}$$

With such formula (or with different ones), the distribution is accurate and it may just need slight variations (which can be done with the new channel state information retrieved) to have a good bit loading [12]. So it is faster as less iterations are done, but it does not necessarily guarantee the maximum effective bit rate for a target BER.

Both procedures can be used altogether allowing to find optimum bit distributions looping the cycle the smallest possible number of times.

### 1.4.3. Bit and power loading

Whereas bit loading impacts directly in each subcarrier by adjusting the modulation level they support given the CSI, power loading amplifies or attenuates the resulting modulated symbols. Thus both systems are conceived to add a flexibility that may be used to maximize the throughput in a transmission and to make it more reliable.

The same theory is behind both methods but with an inherent increase in complexity as more parameters have to be considered, even though some equations have variables for the two adaptive systems. It is difficult to find the limit between how advantageous is to reduce the power in a subcarrier that sees also its bit allocation decreased, in order to have power available for other subcarriers.

The bit and power loading algorithms, also known as BPL, start applying the bit loading algorithm first pushing the subcarriers either to the highest supported modulation level [9] or with a distribution that first considers the CSI (without applying power loading) [14][15][16]. With this setup, preestablished symbols are transmitted and in reception, the new channel state information is retrieved.

If the value is below the required bit error rate, depending on the initial bit distribution the configuration is maintained (highest possible number of bits tolerated by the system) or slightly changed trying to increase the amount of bits per frame (efficient distribution). In the case of surpassing the BER threshold, power loading is enabled taking into account the channel state information. Then, the signal with a variable power distribution is transmitted and the BER is measured again. The procedure is repeated: if the value is low enough, the configuration can be kept or small changes may be done. Otherwise, the bit loading algorithm decreases the number of bits used in the subcarriers which do not satisfy the BER constraint and the steps are done again and looped until the transmission is properly set up [9].

The Levin–Campello algorithm [14] does a more refined process to reach a good bit and power loading distributions, which could be considered as the second option presented where the CSI is used in the first bit distribution. This algorithm, instead of considering the bits per symbol that each subchannel is assigned (separately), it focuses in the total throughput, $b$, which has to be maximized by finding the best power combination (1.9).

$$b = \sum_{n=1}^{N} \log_2 \left( 1 + \frac{P_n \cdot SNR_n}{\Gamma} \right) \tag{1.9}$$

Where $\sum_{n=1}^{N} P_n$ is the fixed power budget constraint and $SNR_n$ is the signal-to-noise ratio for the $n$th subcarrier.

# CHAPTER 2. OPTICAL TRANSMISSION SYSTEMS

## 2.1. Introduction

In order to transmit an OFDM signal through an optical fiber, E-O (electrical to optical) modulation and O-E (optical to electrical) demodulation stages have to be implemented.

The usual, simplest and most economical way of E-O/O-E conversion is IM/DD, intensity modulated and direct detection systems. The signal in the transmitter is modulated on the intensity (power) of the optical signal and in the receiver, the reverse operation is performed.

In presence of channel linear distortions, for example chromatic dispersion in single mode fibers, SMF, due to the non-linear process of converting from electrical to optical and optical to electrical, IM/DD systems would suffer from non-linear distortion. To avoid this problem, spectral guard bands would have to be allocated or more advanced E-O/O-E systems such as optical IQ modulation and coherent detection could be used.

This review of optical modulation and demodulation systems will allow to understand the implications of applying OFDM modulation systems to be transmitted over an optical fiber, a topic discussed in the next chapter. Also, some aspects from the VPI software are briefly explained to know better the working environment used to create an optical transmission system in which to run simulations.

## 2.2. Optical modulation and demodulation

For transmissions through optical channels, the modulation of an optical signal and its demodulation are necessary. Basically two options exist to perform both operations: for the first one, intensity modulation and optical IQ, whereas the demodulation may be done by means of direct detection or coherent detection [17]. Since this thesis focuses on IM/DD systems, only intensity modulation and direct detection are explained.

From the IFFT performed in the OFDM modulator up to the FFT made in the demodulator, there are many stages in which the digital sequence and analog signal are processed, including the transmission through the medium. To achieve the lowest deviation between the IFFT output and the FFT input, a linear relationship between the baseband electrical signal and the optical field is precised: each discrete OFDM subcarrier frequency in baseband electrical field has to be mapped to a single discrete frequency in the optical domain. However, this is not the case in IM/DD system and it will be explained in the next sections.

### 2.2.1. IM: DML and MZM-QP / DD

To modulate in intensity may be done either with a directly modulated laser, DML, or externally with a Mach Zehnder modulator, MZM. The first option is based in a laser diode modulating the bias current from an electrical signal (see Fig. 2.1).

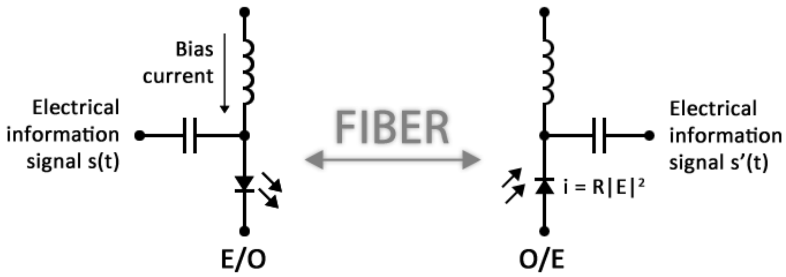


Figure 2.1: IM/DD with DML

A bias current must be supplied to set the operating point in the middle of the linear zone of the transfer function, above the current threshold and below the region where saturation effect sets off (see Fig. 2.2). A characteristic parameter is the slope efficiency which determines the proportional variation of the optical power as the current changes and vice versa , but just when operating in the linear region of the laser diode.

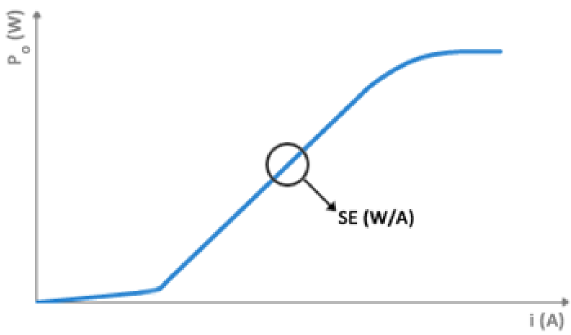


Figure 2.2: Slope efficiency, SE, of a directly modulated laser

A MZM (see Fig. 2.3), on its side, modulates the light output of a continuous wave laser. Due to the electro-optical nature of the modulator, the optical signal generated is altered by means of the refraction index which depends on the applied electrical field. Its main advantage is its bandwidth which is not limited as in directly modulated lasers. And hence the reachable distances and bit rates by the optical signal are longer and higher than using direct modulation.

When a direct detection is used to retrieve the electrical signal, the voltage in the laser driver has to ensure that the Mach Zehnder is working in the quadrature point, QP (see Fig. 2.4), where the transfer function is linear for an ideal electrical intensity to optical power ratio. When it operates within the quadrature point it is commonly called MZM-QP.

A characteristic singularity when modulating in intensity is that optical signals can only carry positive intensity values, whereas transmission using electrical fields can be bipolar as voltages may be positive and negative. This limitation does not allow modulating the

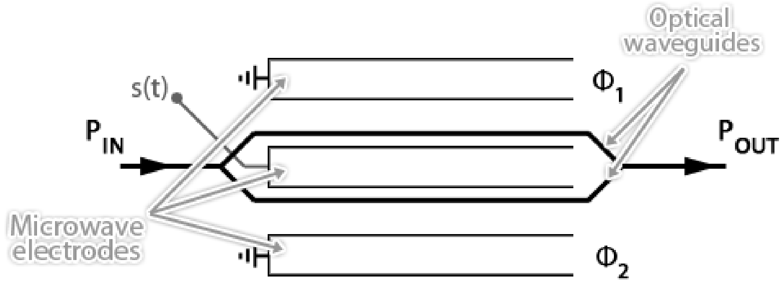


Figure 2.3: Mach Zehnder modulator

real and imaginary components of a typical OFDM signal. There are two ways to generate the optical signal to avoid this limitation: with hermitian symmetry [17] and with an electrical IQ stage (explained in the next section) prior to IM modulation.

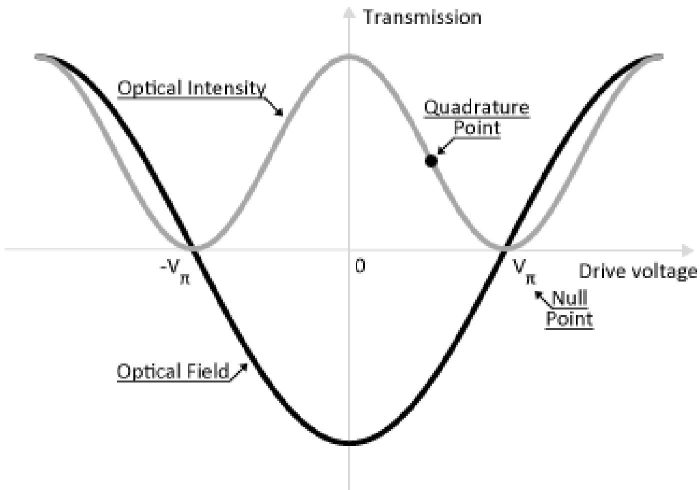


Figure 2.4: Optical intensity and optical field transfer function

In direct detection, DD, a photodetector converts the incoming optical signal to output an electrical signal (see Fig. 2.5) by means of mixing the optical carrier with the frequencies of the OFDM band.

Accordingly, the direct detection method is a cost-effective solution more convenient for applications intended for short distances and or low data rates.

## 2.2.2. IM/DD OFDM systems: Electrical IQ

Having the real and imaginary components of the OFDM symbols separated in two channels (see Fig. 1.3), the electrical IQ converts them to a real signal (see Fig. 2.6). It uses a mixer to up convert the in-phase and quadrature signals to an intermediate frequency, IF, using a local oscillator and then they are added. Two DACs are needed to convert both branches to analog signals with their whole bandwidth used for the data generation. The IF signal is finally intensity modulated generating a double sideband, DSB, optical signal.

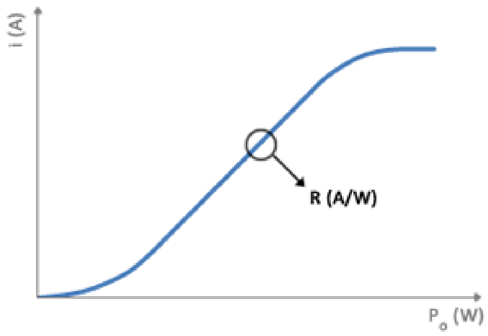


Figure 2.5: Responsivity, R, of a photodetector

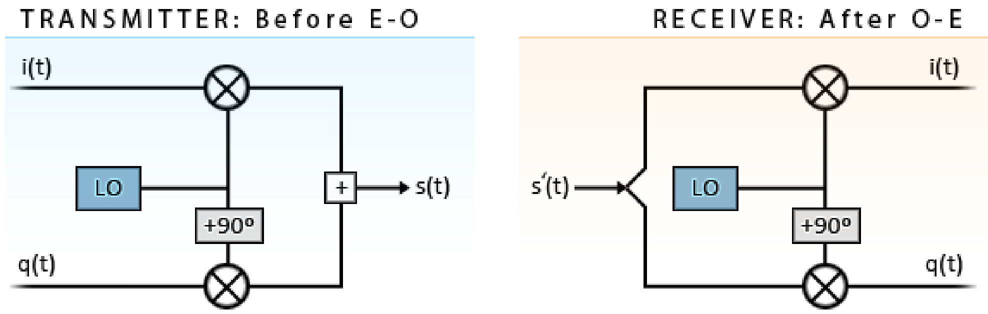


Figure 2.6: Electrical IQ block diagram: left is for transmitter and right is for receiver

The same scheme is used in reception, but inverted (see Fig. 2.6), so after converting from the optical to the electrical signal, an electrical IQ mixer is used to down convert the signal to baseband.

When the mixing has taken place, the real and imaginary parts are present separately in the two branches. Two low-pass filters, one per branch, are used to remove the alias at high frequencies. Finally, two analog-to-digital converters are needed to recover both digital sequences so they can be operated with in the receiver.

### 2.2.3. IM/DD spectrum

In intensity modulation, with the electrical IQ method, a guard band has to be left as intermodulation products appear (see 2.7) with decreasing power as they get further from to the central frequency.

Their origin is in the optical modulation: the optical power generated (2.1) is related to the modulation index from the laser bias, $m$, and the modulated signal, $s(t)$ (2.2). And where $P_0$ is the carrier power.

$$P_{opt}(t) = P_0 (1 + m \cdot s(t)) \tag{2.1}$$

$$s(t) = a(t) \cdot cos(\omega_{RF} \cdot t) \tag{2.2}$$

The optical field equals to the square root of the optical power (2.3), that expressed in its Taylor polynomial (2.4) shows the appearance of the intermodulation products.

$$E_{opt}(t) = \sqrt{P_{opt}(t)} = \sqrt{P_0} \cdot \sqrt{1 + m \cdot s(t)} \tag{2.3}$$

$$E_{opt}(t) = 1 + \frac{m}{2}s(t) - \frac{m^2}{8}s^2(t) + \frac{m^3}{16}s^3(t) - \ldots \tag{2.4}$$

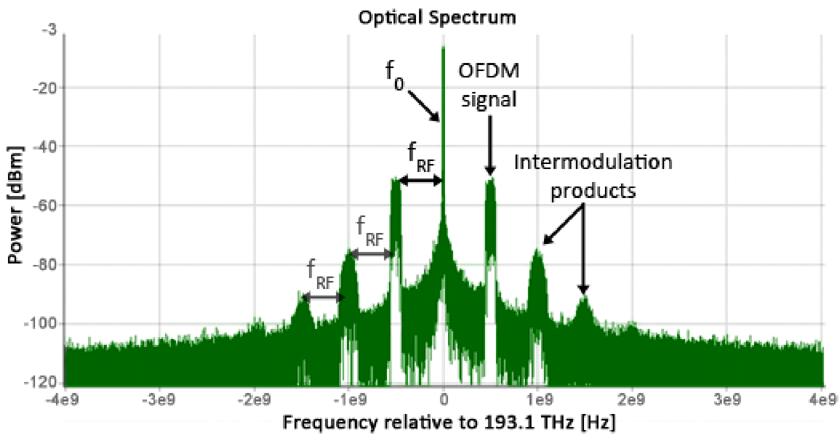


Figure 2.7: Intermodulations generated in intensity modulation (spectrum from VPI)

In direct detection, a guard band also has to be used because intermodulation products arise in the optical to electrical conversion stage, which is characterized by the equation (2.5):

$$i = R \cdot |E|^2 \tag{2.5}$$

Where $R$ is the responsivity of the photodetector and $E$ is the optical field.

Therefore, to remain unaffected, the first half of the whole band should be reserved for a guard band whereas the second half should be used for the optical OFDM signal (see Fig. 2.8).

As the intermodulations products appear in the guard band, with a band-pass filter it is possible to just recover the OFDM signal and proceed for its demodulation. Nevertheless, since the optical bandwidth used is the double of the information bandwidth, this scheme is spectrally inefficient.

To avoid the effect of the intermodulation products, the RF frequency, $f_{RF}$, should be at least 1.5 times the OFDM signal bandwidth.

When the signal has been optically modulated, a DSB signal is present at the modulator output. If a double sideband optical signal is transmitted, the presence of significant chromatic dispersion can cause fading with direct detection. The cause is that there is a phase difference between the two bands which is problematic when they overlap in the same

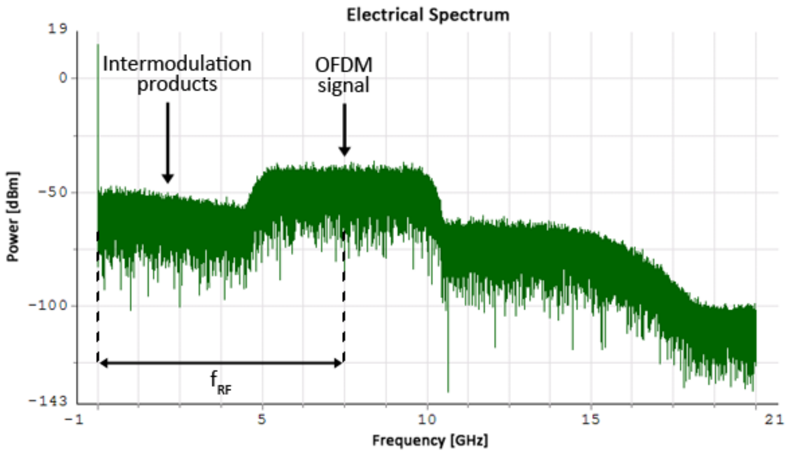


Figure 2.8: Intermodulations generated in direct detection (spectrum from VPI)

electrical frequency band because of the detection. Thus double sideband, which is used in the simulations, is often used in low-cost applications where chromatic dispersion is not present or not a limiting factor [3]. With an optical band-pass filter (SSB filter), one of the bands can be eliminated obtaining a single sideband signal, SSB.

For the spectra in IM/DD systems, the optical modulation index, OMI, is a parameter used to describe the relation between the powers of the optical carrier and the first optical band (see Fig. 2.9).

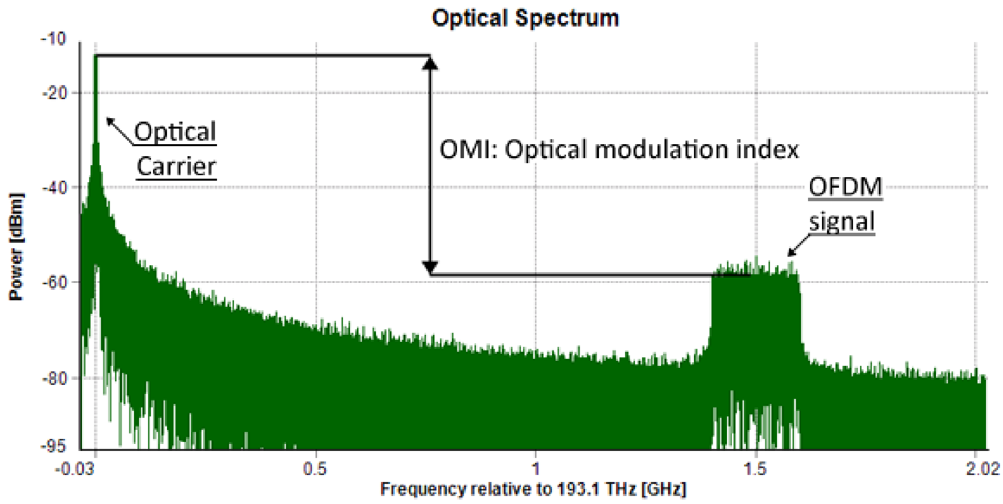


Figure 2.9: Optical modulation index (spectrum from VPI)

## 2.3. Chromatic dispersion

There are two types of optical fibers: single-mode and multi-mode, SMF and MMF respectively. The difference between them is the amount of rays of light, also known as modes, that are transmitted through the fiber. A single ray of light can comprise many waves at different frequencies if they are of the same mode. SMF, the only fiber considered in this document, usually maintains better the light pulses at long-haul than multi-mode fibers, hence MMF is restricted to short-range applications.

The chromatic dispersion is a deterministic distortion related to the phase velocity of the waves when interacting with this type of optical channels. The phase velocity is dependent of frequency. This means that the phases of waves with different wavelengths do not travel at the same speed so the subcarriers arrive at different times to the receiver. If the cyclic prefix is not long enough, the symbols will overlap in the FFT demodulation window (see Fig. 2.10).

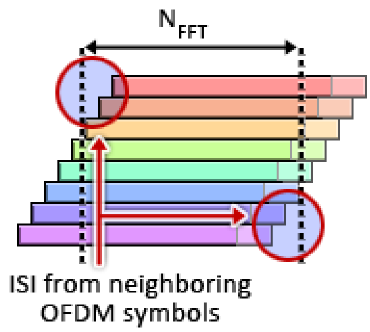


Figure 2.10: Chromatic dispersion effect

Even though VPI has not been documented yet in this thesis, setting up a simple optical transmission network without bit nor power loading, the chromatic dispersion effects can be seen when not using equalization and how the constellation is improved when equalization is applied (see Fig. 2.11).

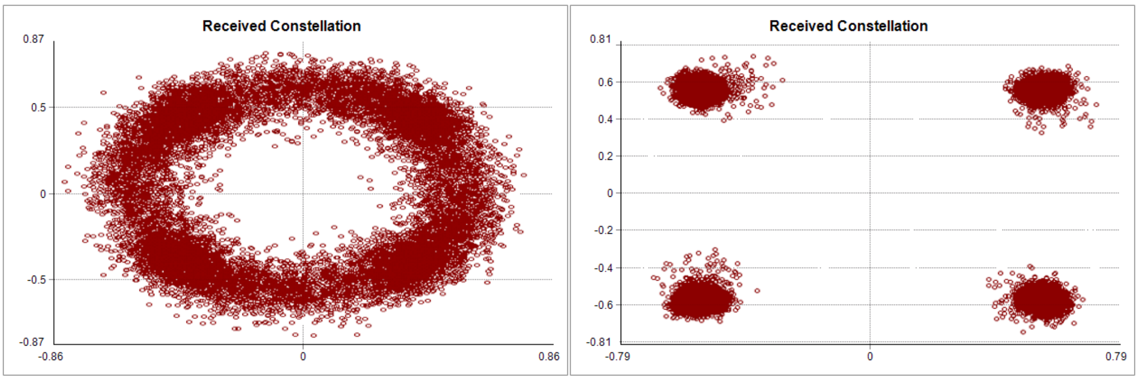


Figure 2.11: Received constellations: left is without equalization and right is with equalization (from VPI: 4-QAM, 64 subcarriers, 10Gbps, optical fiber 1000km long, 20% cyclic prefix)

The group delay (2.6) establishes the delay suffered by the envelope and is given in time per fiber length units. The phase constant, $\beta$, has a nonlinear dependency with the frequency which can be approximated by its Taylor polynomial (2.7).

$$H(\omega) = e^{j\beta(\omega)z} \tag{2.6}$$

$$\beta(\omega) \simeq \beta(\omega_{ref}) + \left.\frac{\partial\beta}{\partial\omega}\right|_{\omega_{ref}} (\omega - \omega_{ref}) + \frac{1}{2}\left.\frac{\partial^2\beta}{\partial\omega^2}\right|_{\omega_{ref}} (\omega - \omega_{ref})^2 + \ldots \tag{2.7}$$

Where $\omega_{ref}$ is the center frequency of the signal's bandwidth. Each $\beta$ term, either the first one or the derivates, may be expressed as $\beta_{order}$ (2.8, 2.9, 2.10).

$$\beta_0 = \beta(\omega_{ref}) = \frac{\omega_{ref}}{v_{phase}} \tag{2.8}$$

$$\beta_1 = \left.\frac{\partial\beta}{\partial\omega}\right|_{\omega_{ref}} = \frac{1}{v_{group}} = \tau_g \tag{2.9}$$

$$\beta_2 = \left.\frac{\partial^2\beta}{\partial\omega^2}\right|_{\omega_{ref}} = \frac{\partial}{\partial\omega}\tau_g \tag{2.10}$$

The $\beta_0$ phase constant is also known as wavenumber and it defines the relation between the frequency and the phase velocity. $\beta_1$ is the group delay and $\beta_2$ the group delay dispersion, GDD. This last parameter is also related to the total dispersion in the physical medium. Operating the expression (2.11):

$$\left.\frac{\partial}{\partial\lambda}\right|_{\lambda_{ref}} = \left.\frac{\partial}{\partial\omega}\right|_{\omega_{ref}} \left.\frac{\partial\omega}{\partial\lambda}\right|_{\omega_{ref}} = \frac{\partial}{\partial\omega}\frac{\partial}{\partial\lambda}\left.\left(\frac{2\pi c}{\lambda}\right)\right|_{\lambda_{ref}} = -\frac{2\pi c}{\lambda_{ref}^2} \tag{2.11}$$

The fiber's dispersion parameter can be obtained as (2.12):

$$D = \left.\frac{\partial\tau_g}{\partial\lambda}\right|_{\lambda_{ref}} = -\frac{2\pi c}{\lambda_{ref}^2}\beta_2 \tag{2.12}$$

Where $\lambda_{ref}$ is the wavelength and *c* the speed of light. And the total dispersion $D$, in the particular case of an optical fiber in third window (usual wavelength of 1.55$\mu m$) is *17ps·nm$^{-1}$·km$^{-1}$*.

## 2.4. VPI software

The VPIphotonics™ suite gathers a set of software applications designed for running photonic and optical simulations. From this suite, mainly VPItransmissionMaker™ Optical Systems is the application used to study the effects of adaptive algorithms in this thesis.

### 2.4.1. Graphical user interface, hierarchy and modular design

VPI software has a graphical user interface, GUI, that allows to design transmission systems intuitively by using interconnectable blocks which simulate devices (see Fig. 2.12) such as pseudo random bit generators, filters, optical fibers, spectrum analyzers, etc. Each one of these blocks has settings or parameters that can be adjusted for different behaviors like voltages, frequencies, roll-off factors, etc. The blocks are provided with explanatory documentation sheets which inform of their utility, the function of the related parameters and their range of values and, finally, some of the theory behind to understand how they were programmed, possibly including some of the equations used.

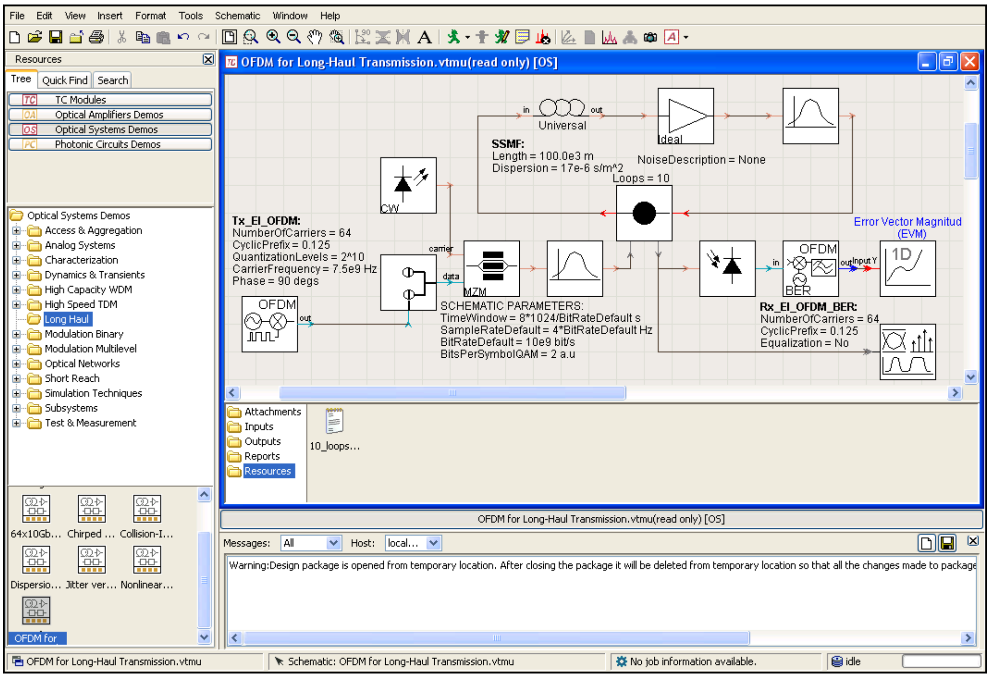


Figure 2.12: VPI main window

For more structured designs, VPI is provided with a hierarchical organization. It has up to three levels of elements or hierarchies: universes, galaxies and stars.

**Stars** Third level. An indivisible element with a specific functionality. It has either inputs or outputs, or both.

**Galaxies** Second level. It is composed of several stars and galaxies interconnected. The galaxies, if they are not encrypted, have a *"Look inside"* option from its drop-down menu (see Fig. 2.13) allowing to see its internal design. It has either inputs or outputs, or both, that are then used in a higher level galaxy or in the universe.

**Universes** First and top-most level. It is composed of several stars and galaxies interconnected. The lack of a complete and fully functional universe causes an error from VPI because the whole universe is simulated. It has neither inputs nor output.

With this hierarchy, it is far from difficult to have a well organized design having in the universe the most representative modules, for example the OFDM coder and the decoder

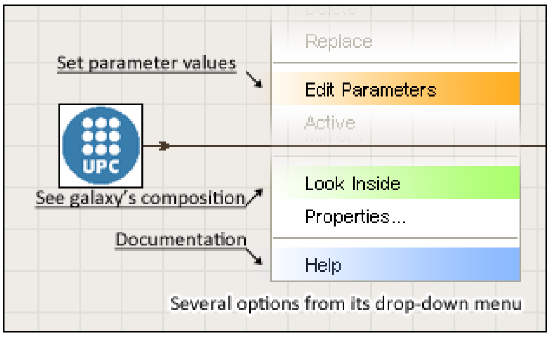


Figure 2.13: A galaxy block and its drop-down menu partially

are reduced to a couple of galaxies in the transmission system created for this thesis and that is what it is seen from the top level view. If modifications have to be done, just looking inside of them is enough to have access to all of their elements.

A galaxy itself may and usually has parameters. The parameters of the galaxy are added by the user conveniently. Stars and galaxies inside a galaxy are self-contained. This means that they can only access to the variables generated within the galaxy and the parameters of the galaxy that includes them. Therefore, external data that has to be operated, have to be included in the parameters of the galaxy.

### 2.4.2. Main parameters and conditions

Over all the parameters the blocks have, there are some intrinsic to the simulation itself, also known as schematic parameters, which have to be adequately configured for appropriate results. The most relevant are listed below.

**TIMEWINDOW** It is the length in time that is simulated. If the duration is increased, more data is generated and processed.

**SAMPLERATE** It determines the number of samples taken by second. Therefore if the sampling frequency is higher, the resolution in the time domain is better. This value also determines the maximum simulation frequency limited by the Nyquist rate.

**BITRATE** It specifies the rate at which bits are transmitted through the system, so the transmitting bandwidths depend on this value.

These parameters have to be set properly to fulfill certain conditions so VPI can run the simulations. Otherwise, problems may occur and the simulations cannot be carried out.

First, an integer number of symbols must be generated at the end of the simulation (2.13) because the modulating process fails if that does not happen; for the symbol rate computation it is mandatory to know the number of bits that are used per symbol (2.14), which is accessible through the BITSPERSYMBOL variable (explained in next chapters). An effective solution to make the total number of symbols simulated independent of the bits is the use of the following relationship (2.17).

$$N_{symbols}\,[symbols] = \textsc{TimeWindow} \cdot \textsc{SymbolRate} \in \mathbb{N} \tag{2.13}$$

$$\textsc{SymbolRate}\left[\frac{symbols}{second}\right] = \frac{\textsc{BitRate}}{\textsc{BitsPerSymbol}} \tag{2.14}$$

VPI performs the FFT and IFFT, transparently to the user, when simulating systems to make the corresponding calculations for some blocks (because they are programmed with such algorithms). By any means it is necessary to have powers of two as total amount of simulated samples (2.15). For this reason, SampleRate (2.16) and TimeWindow (2.17) must be chosen depeding on the number of bits per symbol. This way, the resulting number of samples can easily fulfill the condition since both parameters are already set to a power of 2. Then, in order to avoid problems in the simulations, the following relationships have been used:

$$\textsc{SampleRate} \cdot \textsc{TimeWindow} = N_{Samples}\,[samples] = 2^{n}, where\ n \in \mathbb{N} \tag{2.15}$$

$$\textsc{SampleRate}\,[Hz] = 2^{s}\,\frac{\textsc{BitRate}}{\textsc{BitsPerSymbol}} \tag{2.16}$$

$$\textsc{TimeWindow}\,[s] = 2^{t}\,\frac{\textsc{BitsPerSymbol}}{\textsc{BitRate}} \tag{2.17}$$

Where $s$ is 5 and $t$ is 14, hence $n$ is 19. As for the bit rate (2.18), the value directly determines the bandwidth which has to fit in the band of frequencies that is disposed of.

$$\textsc{BitRate}\,[bps] = 10^{b} \tag{2.18}$$

Where $b$ is 8. It has to be noted that all these values are the ones used for the simulations of this thesis, so they can be modified conveniently.

Finally, new parameters can be created in the schematic. A reason to do this is to share common variables that are repeatedly used within the universe. Like in the self-contained galaxies where parameters could be included, doing the same in the universe propagates them to all of its elements. Then, where the variables are used, if they are set to use the global parameters, just updating them once in the schematic menu is enough to have the variable, in all the blocks that use it, updated to the new value.

# CHAPTER 3. OPTICAL OFDM SYSTEMS

## 3.1. Introduction

The main goal of this thesis is to design an optical OFDM system in VPI that allows a straightforward and easy testing of adaptive loading techniques.

This design is based on the descriptions of OFDM systems and optical transmission systems in the previous chapters. From the many proposals of optical OFDM systems [17], IM/DD has been chosen and the one that this thesis focuses on consists in an electrical IQ up conversion prior to an intensity modulation, IM. After transmitting the optical signal, a direct detection is done followed by an electrical IQ down conversion. The RF frequency is chosen so that a spectral guard band equal to the OFDM signal bandwidth is allocated.

There is a VPI demo template of an optical OFDM system with this same scheme and it has been used as starting point for the VPI system that wants to be created. However, as this demo does not include the possibility to test adaptive techniques, the purpose is to use a *"CoSim"* interface with MATLAB functions to add the bit and power loading capabilities.

## 3.2. Programmed functions in VPI

When the available blocks are not enough to implement a feature, like the adaptive loading techniques, the cosimulation interface block can be used (see Fig. 3.1). This interface includes an external function, in a supported programming language, with variables from VPI passed as parameters. The outputs from the function are passed to the following interconnected VPI blocks. The function header has to be correctly defined in the *"RunCommand"* parameter and the programming language selected in *"InterfaceType"* (see Fig. 3.2).

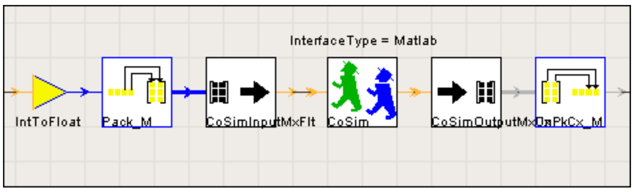


Figure 3.1: *"CoSim"* block to integrate external programmed functions in VPI

| Name | Value |
|---|---|
| General | |
| InterfaceType | Matlab |
| RunCommand | [y]=ofdm_coder(prbs,TimeWindow,BitRate,BpS,Nc,N_FFT,CP) |
| WrapupCommand | |

Figure 3.2: *"InterfaceType"* and *"RunCommand"* parameters from *"CoSim"* block

Once the simulation has been started, VPI opens a command window corresponding to the selected *"InterfaceType"* and executes the function with the variables generated in VPI. When the function ends, the resulting variables are available at the output port.

To calculate the BER in the decoder, it is necessary to know the original transmitted bit sequence. Since the PRBS is generated in the coder, which is a galaxy, and the decoder is a separate galaxy, to share the bit sequence a *"LogicalChannel"* port is used. This channel is connected to a block and stores its output in a variable which is accessible. This way, the information can be used in other galaxies too. In the mentioned case, the PRBS is another variable passed as a parameter to the MATLAB function to calculate the bit error rate.

## 3.3. VPI demo study

Before starting the design, the mentioned demo that comes preinstalled in VPI has been studied to understand how the software works and how to create the optical transmission system in which to implement bit and power loading. It is an OFDM system for long-haul transmissions (see Fig. 3.3) [18]. The selection of this demo is due to the interest in simulating and analyzing the impact of loading algorithms in long distance applications.

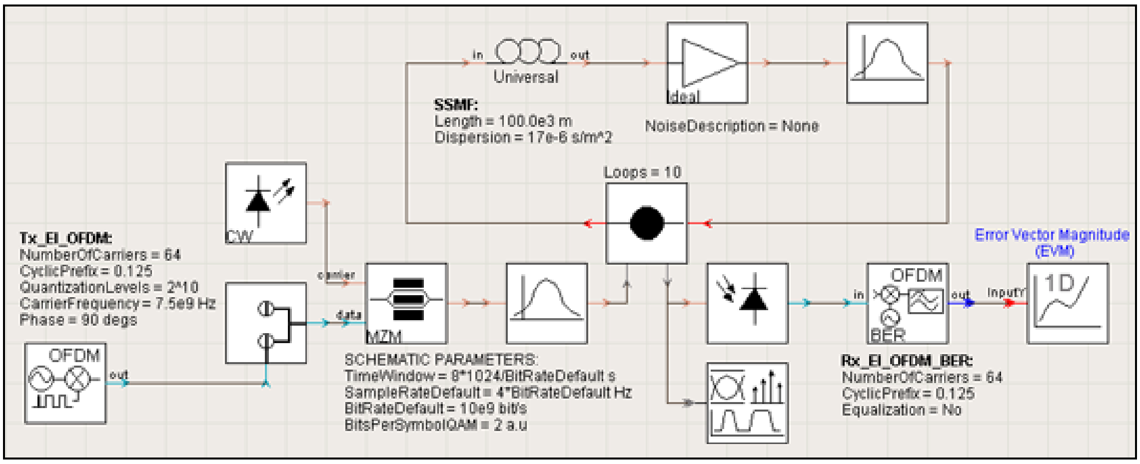


Figure 3.3: Universe from the VPI demo studied

The OFDM transmitter is a galaxy with an output for the analog OFDM signal. The signal is optically modulated by means of a Mach Zehnder modulator, which has a second input for the continuous wave laser to modulate its light. At the MZM output, an optical band-pass filter is used in order to suppress one band and avoid chromatic dispersion frequency fading when the two optical DSB bands are detected at the same frequency in the receiver. This SSB optical signal is transmitted through the optical fiber.

The configuration of the physical medium consists in using repeaters after several kilometers of fiber, so an amplifier and a filter are used to improve the quality of the signal as much as possible. VPI is able to loop this cycle as many times as indicated which, in this case, is 10 times with a 100km long fiber per loop.

After transmission, a photodetector is used for direct detection. Under this block, a spectrum analyzer is placed to show the optical spectrum before the detection. The electrical signal from the DD is the input for the OFDM receiver which is also a galaxy. Its output contains the error vector magnitude value that is shown with the simulation thanks to the last, right most, block.

With the *"Look inside"* option, the block diagram of the transmitter can be seen (see Fig. 3.4). First, a pseudo random bit sequence is generated as data source, which is duplicated: one branch is used to store the sequence in a *"LogicalChannel"*, whereas the other one is for the OFDM symbol modulation.

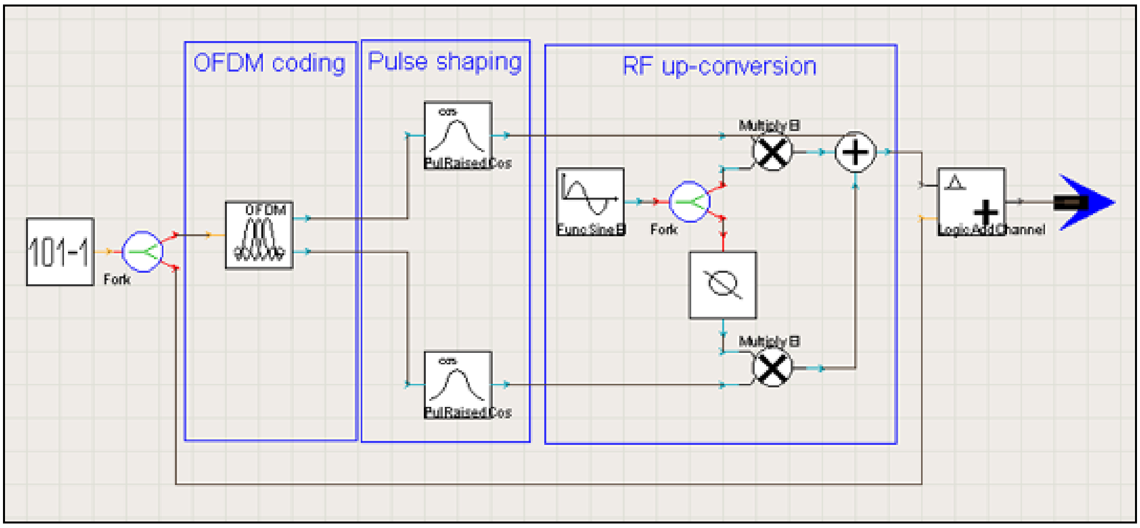


Figure 3.4: OFDM transmitter galaxy from the VPI demo studied

For this second branch, an OFDM coder block from VPI is used. It contains parameters such as the number of subcarriers, cyclic prefix and bits per QAM symbol. So using this information, two digital OFDM sequences are generated: the real and the imaginary parts. Both components are converted to analog signals with the pulse shaping block that uses a root raised cosine (its roll-off factor can be modified in the parameter list) as filter. Finally, an electrical IQ is used for the RF up conversion.

As for the receiver (see Fig. 3.5), the electrical signal in its input is amplified and duplicated for the electrical IQ for the RF down conversion. The in-phase and quadrature signals go to the pulse shaping block with the same configuration as in the OFDM transmitter. The resulting signals go to the OFDM decoder which has the two corresponding inputs. It outputs two digital sequences: one for the real components of the QAM symbols whereas the other is for the imaginary components. Both are used in a BER and EVM calculation block, where the QAM symbols are demodulated. This block is provided with a parameter to access the information from a *"LogicalChannel"* so it can compute the bit error rate and the error vector magnitude.

## 3.4. Optical OFDM transmission system designed

The optical transmission system created (see Fig. 3.6) is similar to the demo one. The IM/DD system is exactly the same, the only difference resides in the optical SSB filter at the output of the Mach Zehnder modulator. In this case, this is not used, so a DSB signal is

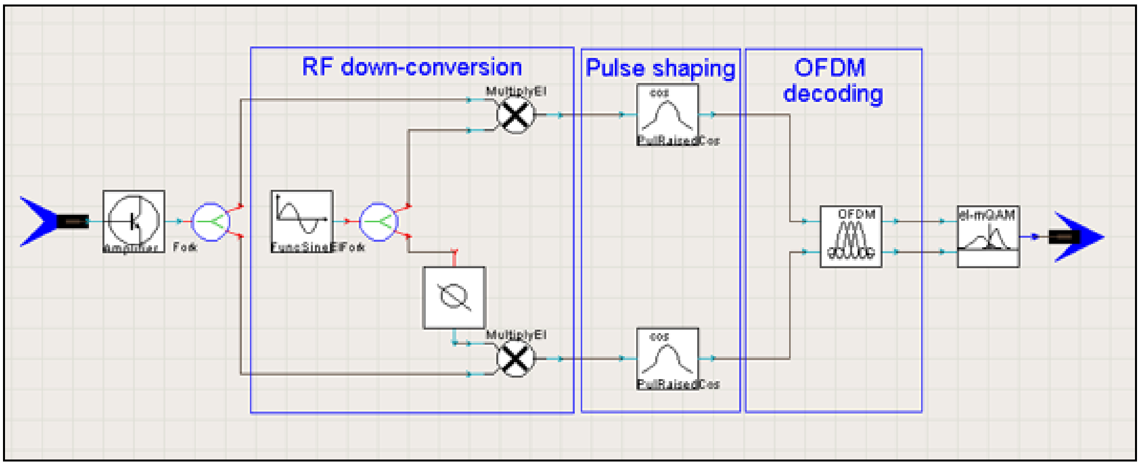


Figure 3.5: OFDM receiver galaxy from the VPI demo studied

transmitted through the optical fiber and the transmission will then be subject to chromatic dispersion amplitude loadings, which have proven not to be very problematic for the cases simulated. When thinking in optical network applications, the absence of filters allows for more flexible operation. There is, nevertheless, an optical filter block in the same place. It is a band-stop one used for simulating a frequency fading within one of the bands. So it is activated and deactivated conveniently when such a behavior wants to be simulated.

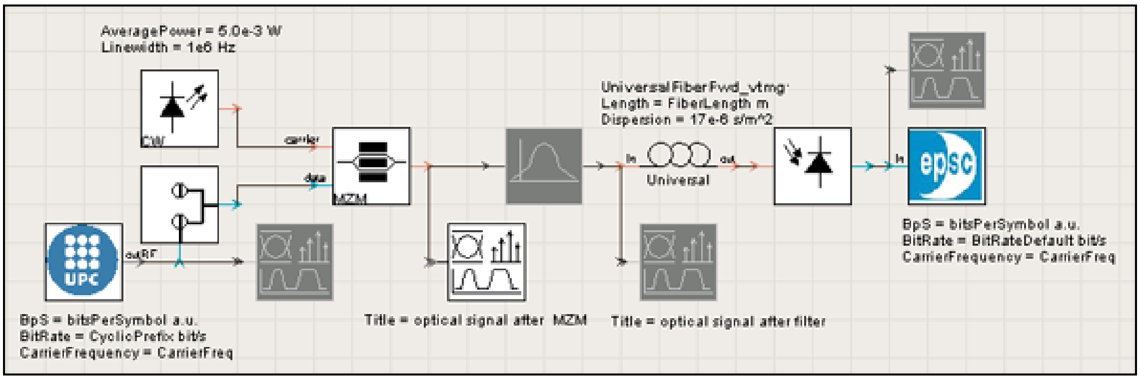


Figure 3.6: Universe from the design used

The composition of the physical medium is simplified. The repeaters and the loops from the demonstration are now removed to just have a 100km long optical fiber.

The transmitter's galaxy (see Fig. 3.7) has several variations if it is compared to the demo template. Instead of using a VPI's OFDM encoding block, a MATLAB function has been programmed to do the process with bit and power loading functionalities incorporated in. After separating the complex values in two channels, they are up-sampled (this is already done inside VPI's OFDM coder) for the pulse shaping. The last stage is the electrical IQ for the radiofrequency up conversion.

The up-sampling factor depends on the SampleRate. To have a power of 2 factor, the equation in the block (3.1) has to simplify the two other variables in the used SampleRate (2.16) expression, as follows:

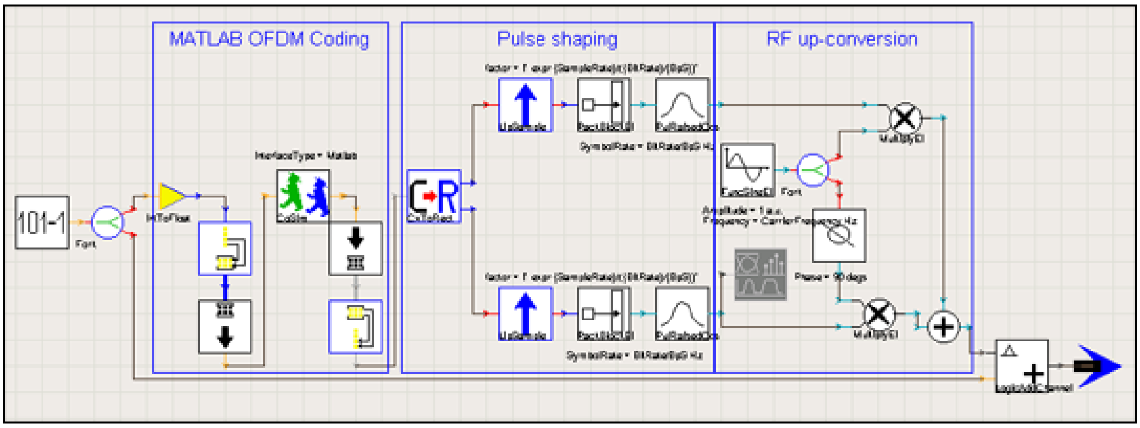


Figure 3.7: OFDM transmitter galaxy from the design used

$$n_{UpDown-Sample} = \text{SAMPLERATE}\frac{\text{BITSPERSYMBOL}}{\text{BITRATE}} = 2^5 = 32 \qquad (3.1)$$

The galaxy of the receiver (see Fig. 3.8), starts with the RF down conversion by means of an electrical IQ and the pulse shaping which are identically done if compared to the demo. A down-sampling is done after the pulse shaping, with the same factor than in the transmitter (3.1) to remove the repeated values and have more bearable arrays in the OFDM decoder also programmed in MATLAB (this is the step to go from the samples of the “analog signal” to symbols, which can be done in MATLAB too). A *“LogicalChannel”* is used as input to the *“CoSim”* interface to be able to compare the demodulated bit sequence with the transmitted one.

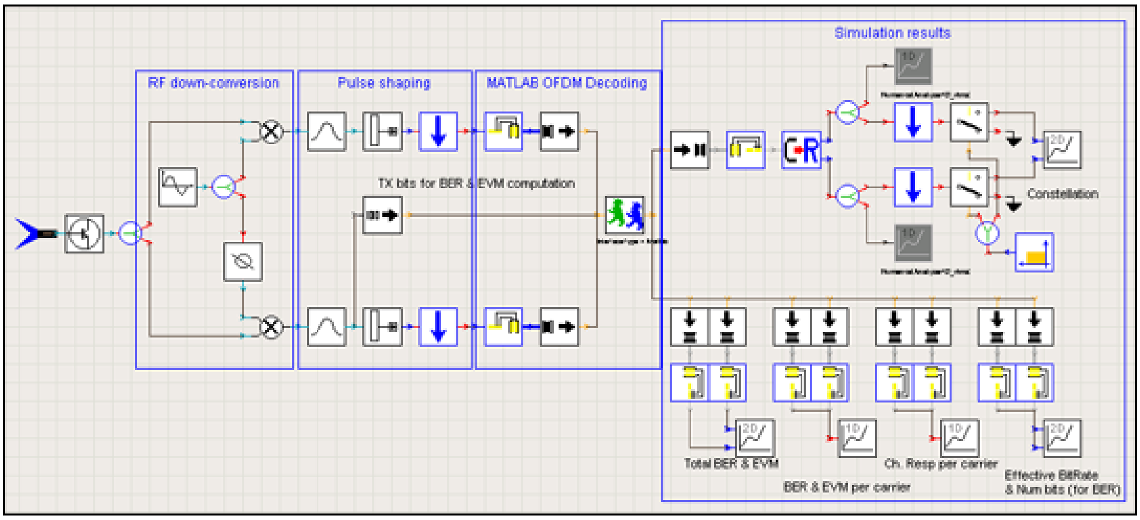


Figure 3.8: OFDM receiver galaxy from the design used

In the last place, the right most group of blocks is used to chart several data after simulating (see Fig. 2.11, Fig. 3.9): BER and EVM per subcarrier, total BER and EVM, an estimated channel response and the effective bit rate of the system. The arrays represented in the graphs are not generated with VPI blocks: it is impossible because they are not prepared to support multiple modulations at once; only the constellation is shown by means of VPI blocks with the symbols from all subcarriers superposed. For this reason, the rest has to be done in MATLAB and plotted in VPI.

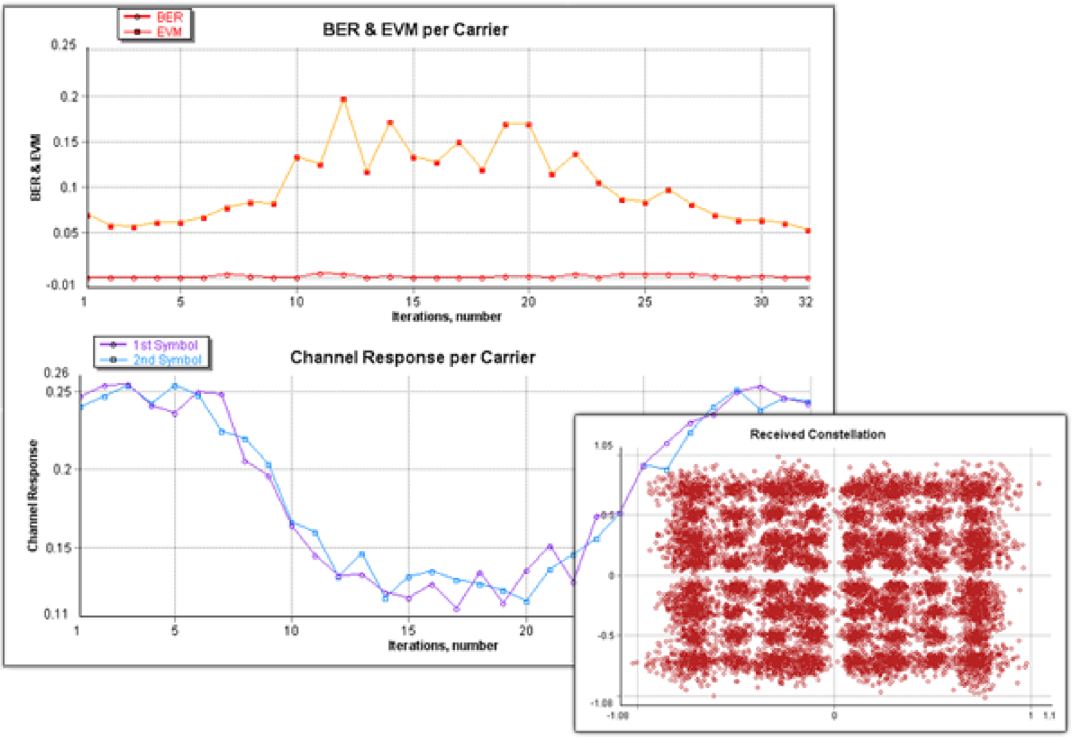


Figure 3.9: Some simulation results from VPI

### 3.4.1. Implementation and parameters of adaptive loading techniques

Thanks to the *"CoSim"* block, the adaptive loading techniques can be added in the OFDM coder and the decoder. The programmed code is partially shown and explained later (for the complete functions, see Appendix A).

First, the created parameters (see Fig. 3.10) are detailed for a better understanding on how to configure bit loading and power loading and how they are programmed.

| **DesignVars** | |
|---|---|
| useTraining | 1 |
| bitsPerSymbol | 6 |
| bitL | 3 3 4 4 5 5 5 6 6 6 5 5 4 4 3 2 |
| useWeight | 1 |
| weight | 0.8 0.9 1 1.1 1.1 1.2 1.2 1.25 1.25 1.2 1.2 1.1 1 1 1 0.9 |
| orderSubcarriers | 1 |
| Nc | 16 |
| N_FFT | 16 |
| CyclicPrefix | 0.2 |

Figure 3.10: Schematic parameters for bit and power loading

**USETRAINING [0: DISABLED, 1: ENABLED]** When enabled, two OFDM symbols are included at the beginning of the simulation for a channel estimation. The pseudo random bit sequence, PRBS, source is unaltered and used as information, therefore the training sequence is generated independently. The inverse of the channel response is used in the one-tap equalization (complex equalization coefficients are used per subcarrier). Both OFDM frames use two preestablished different random BPSK sequences known to the receiver.

**BITSPERSYMBOL [INTEGER]** This variable is related to the amount of bits the OFDM frame assigns to each subcarrier without taking into consideration how many of them are used for the modulated data. All subcarriers must have the same temporal duration in an OFDM frame and it is determined in BITSPERSYMBOL. For a specific bit load configuration, the most efficient value for this variable is the same than the largest number of bits used in one or several subcarriers. In the example (see Fig. 1.10), up to 16 bits could be transmitted if all four subcarriers used the 4 bits. Nevertheless, as bit loading is applied, only 9 are transmitted. It may be thought of as adding false zeros which are not used when modulating the real information. The effective bit rate is reduced. The system is using an effective data rate below its actual capabilities for improving BER.

**BITL [INTEGER ARRAY]** It is used to set the bit allocation. The number is the amount of bits assigned to each subcarrier so the position in the vector (index) is relevant. No number has to exceed BITSPERSYMBOL. If bit loading does not want to be applied, all the subcarriers should be assigned the same number of bits. The amount of subcarriers in the array (their bit allocations) has to correspond with the number of subcarriers set in the N_FFT variable, otherwise the simulation cannot be run. Some subchannels may be disabled using zeros in the vector but the amount of total enabled subcarriers must also match the NC variable.

**USEWEIGHT [0: DISABLED, 1: ENABLED]** When enabled, power loading is applied taking into account the factors put in the WEIGHT array. Otherwise all subcarriers are assigned the same power.

**WEIGHT [FLOAT ARRAY]** It follows the same structure than BITL. When USEWEIGHT is enabled, roughly the same average power is used as the power is just redistributed. If the same number is assigned to all the subcarriers (no matter which number): there is no extra power and the carriers will be of the same amplitude. If the power of some subcarriers is diminished, the power of the rest will be raised and vice versa. After modulating the information, weighting is applied to the resulting symbols. The size of the array must match the N_FFT parameter. The weight from disabled subcarriers (set to zero in BITL is not considered for power loading.

**ORDERSUBCARRIERS [0: DISABLED, 1:ENABLED]** Due to the FFT properties in the OFDM modulation, positive carriers are at the beginning while negative subcarriers are at the end (see Fig. 3.11). When ORDERSUBCARRIERS is enabled, the resulting graphs have the subcarriers in order from lowest to highest frequencies. Otherwise, positive subcarriers are shown first.

**NC [INTEGER]** It sets the number of enabled subcarriers in agreement with BITL.

**N_FFT [INTEGER]** It sets the total number of subcarriers, whether they are enabled or disabled, in agreement with BITL. It must be a power of 2.

**CYCLICPREFIX [FLOAT]** It sets the amount of cyclic prefix used. Being 1 a guard time equal to $T_{OFDM}$, so 100%, and 0 no use of cyclic prefix.

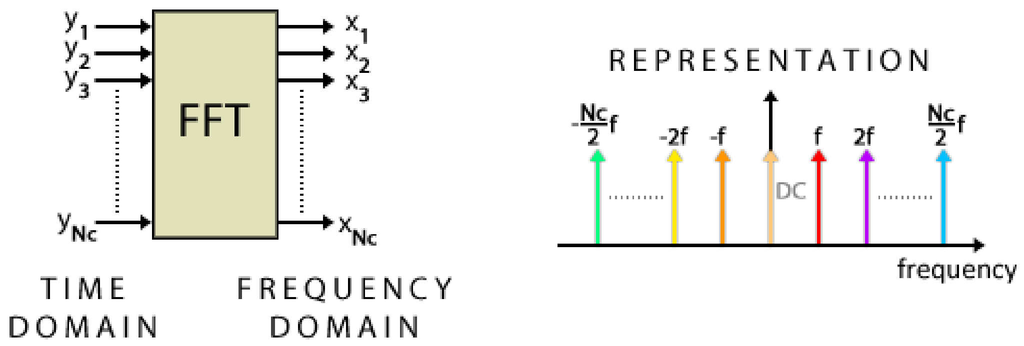


Figure 3.11: Subcarrier representation in VPI

MATLAB is a software characterized by its matrix oriented functions and operations. With such software, the purpose has always been to reduce, as much as possible, the number of loops to make it more efficient.

The main feature requiring loops is bit loading as MATLAB's QAM modulation function depends on the number of bits, which are different for each carrier. In consequence one loop is needed in the coder and, another one, in the decoder in which all the subchannels are covered. A [MxN] matrix is used to store all the OFDM symbols in rows, and the bits that have to be modulated in each OFDM symbol in columns (see Fig. 3.12).

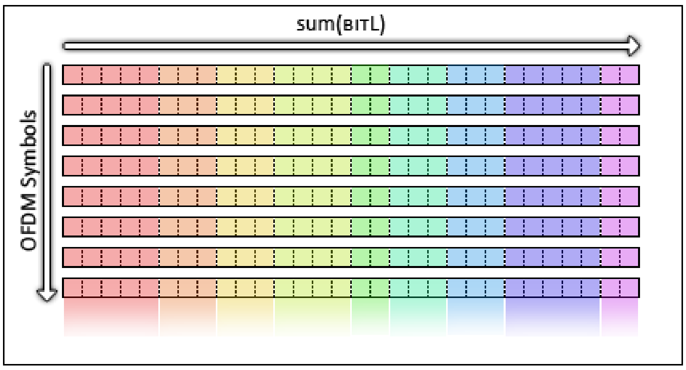


Figure 3.12: Matrix x2 for information (colors symbolize bits allocated to carriers)

In the coder loop, first the index of the subcarrier is stored in the `SC_Index` variable. The same is done with the amount of bits assigned to that subcarrier (`SC_bits`). The bits allocated to every one of the subcarriers are converted to decimal numbers, `bi2de`, and then, all of them are modulated at once when the pointer of the loop is updated to the corresponding subcarrier indexes (see Fig. 3.13).

When the QAM symbols are generated in `xx1_QAM` (a matrix storing the QAM symbols for each subcarrier in one dimension, for each OFDM symbol in the other dimension), a normalization factor (so the modulus from the constellation out most symbols is always 1) and power loading (`Weight_Factor`, which already had calculated the power redistribution so the average total power used remains equal) are applied to them. Prior to perform the IFFT, the modulated symbols have to be rearranged because the loop goes subcarrier per subcarrier, not OFDM symbol per OFDM symbol, thus the information is not in order (see Fig. 3.14).

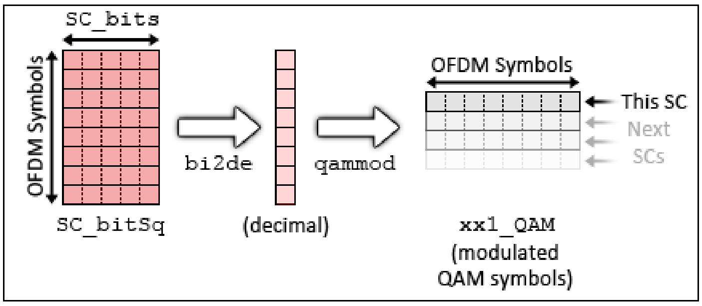


Figure 3.13: Modulation of a subcarrier

```
for n=1:Nc % Loop the enabled subcarriers (Nc)
    SC_Index = carriersON(n); % Get index of the current enabled subcarrier
    SC_bits = BitLON(n); % Get amount of bits for the current subcarrier
    % Get the whole bit sequence for a subcarrier
    SC_bitSq = x2(:, k:(k + BitL(SC_Index) - 1));
    % Generate QAM symbols modulating the decimal sequences
    xx1_QAM(SC_Index,:) = qammod(bi2de(SC_bitSq),2^SC_bits);
    % Subcarrier's factor to apply weight if it is enabled
    Weight_Factor = 1 + useWeight*(WeightON(n)-1);
    % Normalize and weight QAM symbols
    xx1_QAM(SC_Index,:) = xx1_QAM(SC_Index,:) * normfactor(SC_bits) * Weight_Factor;
    k = k + BitL(SC_Index); % Update bitload's pointer
end
```

Figure 3.14: MATLAB's code snippet: Loop for the coder

Finally, the bit load pointer `k` is moved to the next position of the bits to modulate in the following iteration of the loop.

As for the decoder, the steps are reversed (see Fig. 3.15). After doing the IFFT (see App. A.2), the matrix `yy1_QAM` is created with the symbols of the subcarriers in one dimension and the OFDM symbols in the other dimension. All the subcarriers are equalized with their own coefficients stored in the `AGC` array, which is computed using the training sequence at the beginning.

```
for n=1:Nc
    SC_Index = carriersON(n); % Index of the current enabled subcarrier
    SC_bits = BitLON(n); % Get amount of bits for the current subcarrier
    % Subcarrier's factor to undo the weight effect if it was applied
    Weight_Factor = 1 + useWeight*(WeightON(n) - 1);
    % Apply gain and undo weight to QAM symbols (these will be shown as constellation)
    yy1_QAM(n,:) = yy1_QAM(n,:) * AGC(n) / Weight_Factor;
    % Demodulate all normalized symbols from a subcarrier and store them in decimal
    yy1_decimal = qamdemod(yy1_QAM(n,:)/normfactor(SC_bits), 2^SC_bits);
    % Make the binary sequence, finally we will have a [NTS_OFDM, BpF] matrix (like coder)
    RX_bit_seq(:,k:(k + SC_bits - 1)) = de2bi(yy1_decimal', SC_bits);
    k = k + SC_bits; % Update bitload's pointer for the loop
end
```

Figure 3.15: MATLAB's code snippet: Loop for the decoder

The symbols are always multiplied by a factor: if training is enabled, the factor is a complex number from the training sequence and if it is not enabled, an averaged factor is calculated (a real number and the same one for all subcarriers) to amplify the symbols. This amplification is done because the QAM demodulation function in MATLAB, `qamdemod`, only considers the argument and the modulus of the symbols, not the spaces between symbols from an array. This means that a perfect but attenuated constellation would be demodulated incorrectly with high level QAM modulations.

Before demodulating, it is also mandatory to undo the power loading, if it was applied, and apply the appropriate unnormalization, `normfactor`. The `qamdemod` function generates decimal numbers, so a decimal to binary conversion (`de2bi`) is done to recompose the sequence `RX_bit_seq`. At the end of the loop, the bit error ratio and error vector magnitude are calculated for every subcarrier, `BERcarriers` and `EVMcarriers` respectively (see App. A.2).

# CHAPTER 4. SIMULATIONS AND ANALYSIS

## 4.1. Introduction

After designing a full-functional optical OFDM transmission system with bit and power loading, the last chapter is reserved for the simulations carried out to evaluate the impact of both techniques in an optical network.

A reference system is set up with some determined parameters which are maintained in all the simulations. Only the options that correspond to bit loading and power loading are modified to test their performance and compare them with the results obtained from the default scenario.

The presented setups are: constant bit allocation and power as reference, power loading, bit loading with two different approaches and, lastly, simultaneous bit and power loading.

## 4.2. Constant bit allocation and power

A reference optical transmission system is set up without any loading algorithms applied as a simple scenario. All configurations have most of the parameters in common: there are 32 subcarriers and a bit rate of 100Mbps is set; a cyclic prefix of 20% is established and as commented in previous chapters, training sequence is used and pilot tones are not. In average, the same power is always output from the transmitter. Other common parameters can be also reviewed (see Fig. 4.1).

| Schematic Param | | Units |
|---|---|---|
| TimeWindow | 16384*bitsPerSymbol/BitRateDefault | s |
| SampleRateDefault | 32*BitRateDefault/bitsPerSymbol | Hz |
| BitRateDefault | 1e9 | bps |
| DriverAmplitude | 0.17 | A,V |
| LaserBias | 0.5 | A,V |
| CarrierFreq | 1.5e9 | Hz |
| EmissionFreq | 193.1e12 | Hz |
| AveragePowerCW | 5e-3 | W |
| CWLaserLinewidth | 1e6 | Hz |
| FiberLength | 100e3 | m |
| rolloffFactor | 0 | |

Figure 4.1: Common schematic parameters for all simulations

In the reference system, all subcarriers are set to 5 bits with a 32-QAM modulation. The condition that wants to be met is a BER under $10^{-3}$ with the highest possible data rate. The signal transmitted through the optical fiber suffers from a frequency selective fading (see Fig. 4.2). It can be also checked noticing these carriers are the ones with higher BER and EVM levels too (see Fig. 4.3).

In this case, the total system's bit error rate is 0.48%. Without further algorithms, only the carriers with poor signal-to-noise ratio can be disabled to reach the BER threshold. Disabling nine of them (from the number 14 to the number 22), the bit error rate threshold is satisfied, 0.07%, still it would imply a 28.1% loss in effective bit rate. This loss is due to the

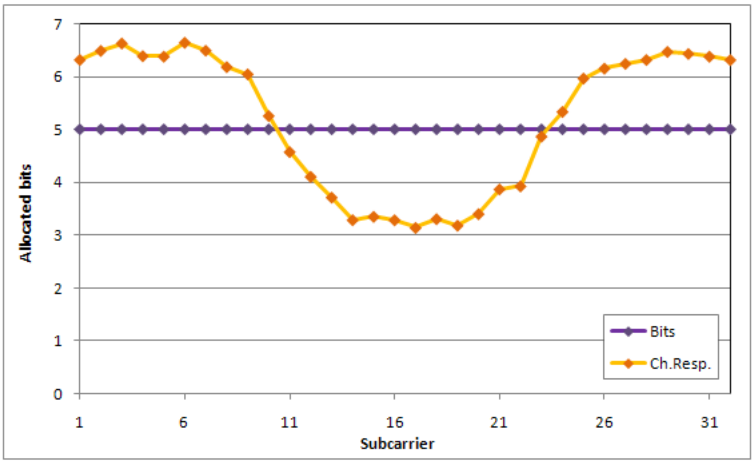


Figure 4.2: Bit allocation (without bit nor power loading) and channel response

lower number of bits modulated in the OFDM frame. If all the subcarriers remained enabled and one bit is increased in each subcarrier (with BITSPERSYMBOL = 6), the effective bit rate would be increased a 20% if the velocity of the system is also increased to output the same number of OFDM symbols per unit of time (4.1):

$$T_{OFDM} = T_b \cdot \text{BITSPERSYMBOL} \cdot \text{N\_FFT} \Rightarrow T_{OFDM} = T_b \cdot n_{BitsPerFrame} \tag{4.1}$$

Where $T_b$ is the time duration of the bit. So if the duration $T_b$ is shorter raising the clock frequency of the system, the same number of OFDM frames can be transmitted with more bits.

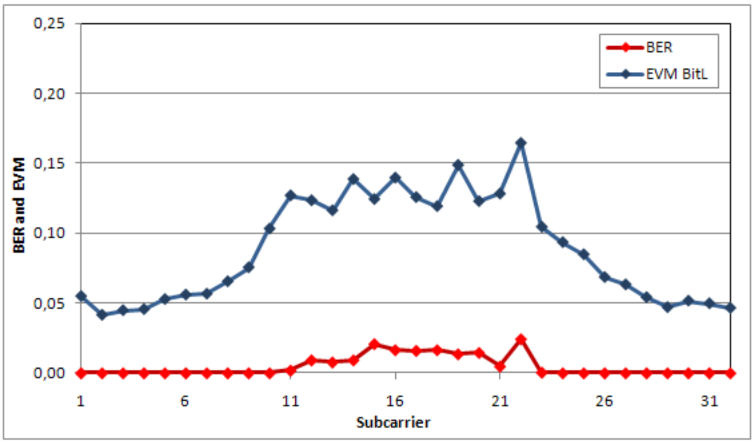


Figure 4.3: BER and EVM (without bit nor power loading)

## 4.3. Power loading

With a variable power distribution, the bit error rate can be decreased effectively. The water filling concept is used to establish a weighting relation that compensates the fading of the channel. The purpose is to have a more uniform signal-to-noise ratio in all subcarriers.

Therefore, the proposed WEIGHT array increases the amplitude of the subcarriers most affected by the fading (see Fig. 4.4) and subtracts some power from the subchannels with higher SNR.

With this configuration, the BER of the system has been reduced to 0.13% thanks to the redistribution of the error probability between all the subcarriers (see Fig. 4.5). It has to be noted that there is no sacrifice in bit rate as the amount of bits per OFDM symbol is the same than before.

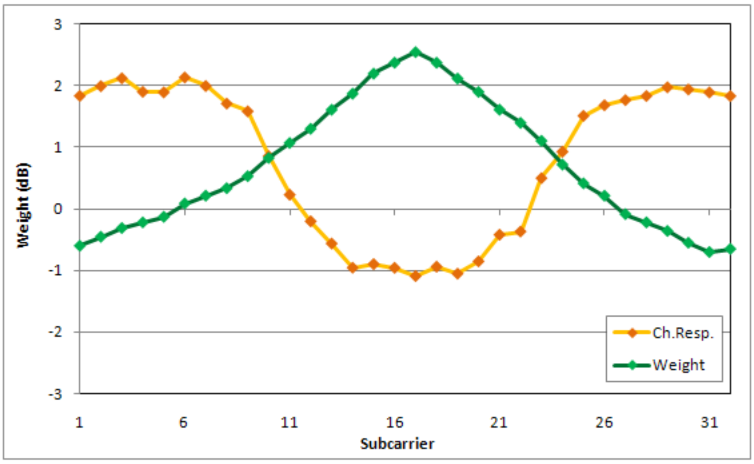


Figure 4.4: Assigned weights (with power loading) and channel response

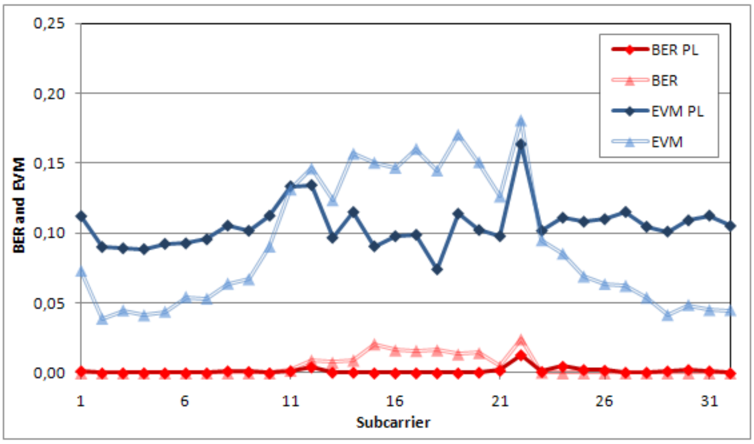


Figure 4.5: BER and EVM (with power loading)

If the optical fiber had about the same behavior in the whole band, with a constant bit allocation and power, more or less the expected bit error ratio and error vector magnitude would be almost the same for all carriers. As this is not the case, the EVM was not uniform, whereas using the proposed power loading scheme it can be seen that the EVM is approximately constant in all the subchannels. This means that the compensation is well made because the impact of the fading is almost the same in terms of how much dispersed the symbols per subchannel are.

When using less power in some subcarriers, their symbols tend to deviate more from their original position in the constellation, thus increasing their error vector magnitude, and the inverse process happens with more powerful subcarriers. The EVM calculation is done applying the normalization factors and removing the weight factors in the receiver, so the EVM is always computed under the same conditions.

The applied weight distribution implies that some subcarriers have now more amplitude and others less amplitude. This can be seen easily in the spectrum of the optical OFDM signal transmitted (see 4.6).

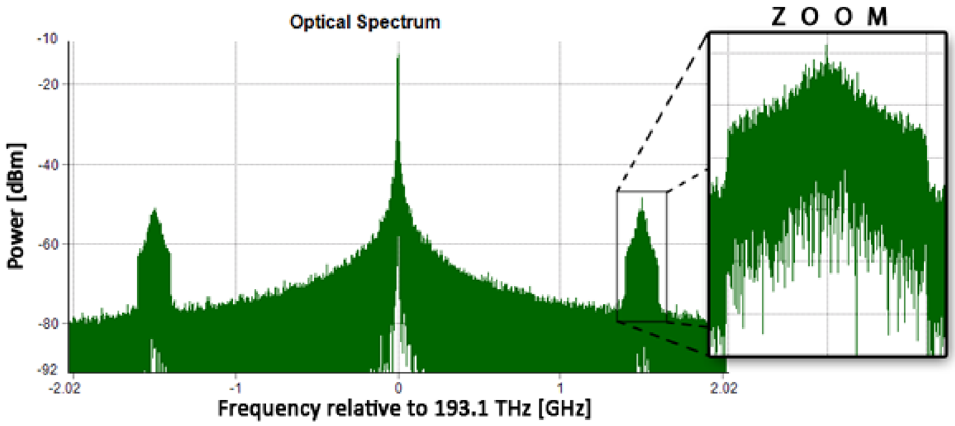


Figure 4.6: Spectrum of the optical OFDM signal transmitted with power loading (from VPI)

## 4.4. Bit loading

Using bit loading alone a consideration has been made to present two different cases. The first one, lowers the effective bit rate to achieve the target BER, so a comparison between this option and the deactivation of subcarriers can be made.

The second case also takes profit from the SNR gaps in some subcarriers to increase their bit allocation considering the channel state information. The main goal is to have the highest possible number of bits per OFDM symbol with a bit error rate below $10^{-3}$.

### 4.4.1. Decrease only

This setup is an enhancement over disabling subcarriers with poor SNR because it may be possible to transmit information in them with lower level modulations. The implementation of a bit loading system can be rather complex although the benefit of using most part of the band is important.

So instead of allocating zero bits to the channels within the range of frequencies affected by the fading, they use one or several bits for the QAM modulation. The best result achieved is an 8% loss in effective bit rate (see Fig. 4.7).

There is a big difference in terms of effective bit rate if it is compared with the 28% loss when the carriers are deactivated. The error vector magnitude does not change much as the dispersion still remains (see Fig. 4.8), nevertheless incorrect demodulations do not occur so frequently because the distance between symbols is wider, thus the total bit error rate is 0.08%.

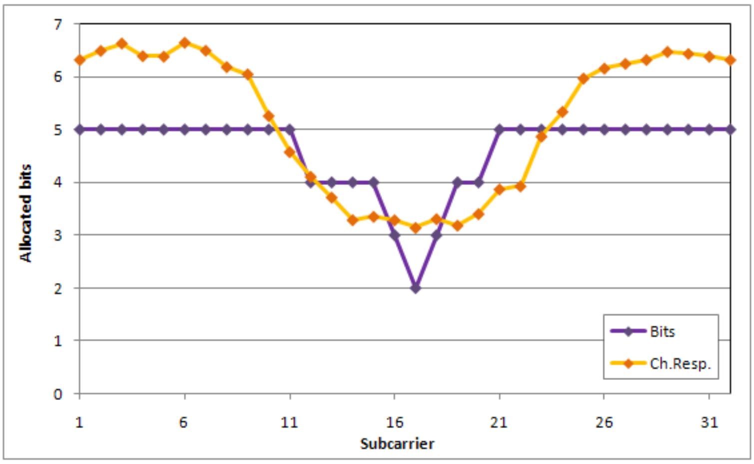


Figure 4.7: Bit allocation (with bit loading: decrease only) and channel response

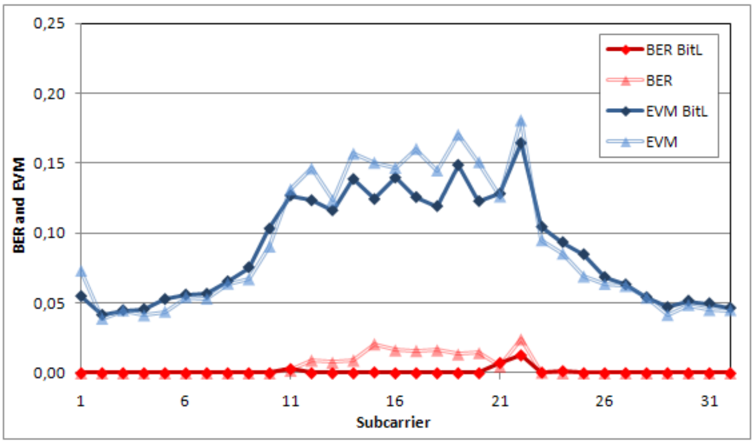


Figure 4.8: BER and EVM (with bit loading: decrease only)

### 4.4.2. Increase and decrease

This second option consists in increasing and decreasing the allocated bits per subchannel, with the initial objective of trying to maintain the number of bits per OFDM frame present in the reference and power loading setups, improving the reliability of the transmission. Thus the subchannels with SNR gap are pushed to higher modulation levels, and the ones with worse quality at reception are assigned a minor number of bits (see Fig. 4.9).

As some carriers are currently using up to six bits (see Fig. 4.10), the BITSPERSYMBOL parameter has to be set to 6, meaning that some spaces from the OFDM symbol matrix are empty (this also happens in the previous scenario even without changing the parameter). Having longer OFDM symbols make its processing slower, which leads to a lower amount of symbols transmitted per second. This can be solved by raising the clock frequency of the components in the transmitter and in the receiver, being this its main drawback. The

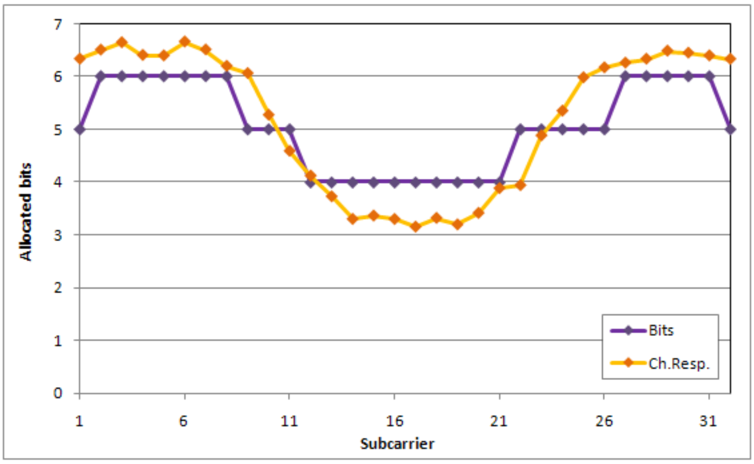


Figure 4.9: Bit allocation (with bit loading: increase and decrease) and channel response

devices are more complex, thus more expensive, and they have a higher power consumption.

In exchange to the clock frequency issue, the overhead due to cyclic prefix is lower and the total bit error rate achieved is 0.09%. Moreover, there is a 1% increase in effective bit rate. There is a significant improvement in the performance of the system.

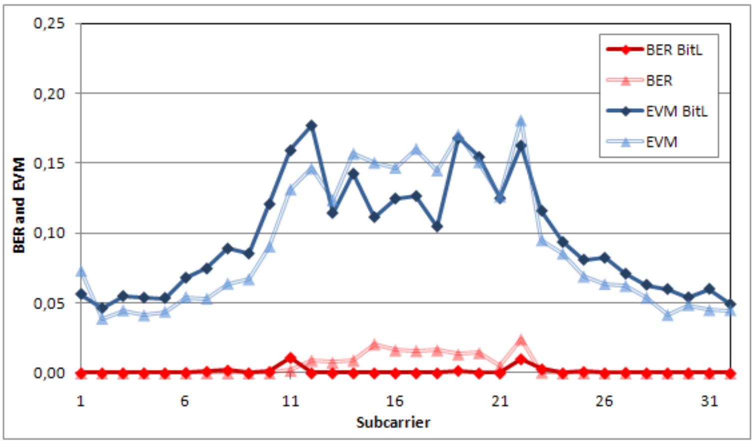


Figure 4.10: BER and EVM (with bit loading: increase and decrease)

## 4.5. Bit and power loading

Finally, the last system presented uses bit and power loading at once. In the previous simulations, power loading achieved great results in terms of a more reliable system just modifying the amplitudes of the carriers. On the other side, bit loading also improved the system much more noticeably. It is expected that the conjunction of both techniques is going to enhance even more the performance of the transmission.

First, the bit loading is set up maintaining the effective bit rate. To use the previous bit allocation (see Fig. 4.10) is a good starting point. Considering the dispersive nature of the channel as the CSI reveals, the allocations are slightly modified doing several iterations to this step. Then the power loading is applied trying to fulfill the BER constraint, doing some iterations too. When the bit error rate constraint is fulfilled, the bit allocation is increased and the loop restarted. Repeating the steps it is possible to push the transmission close to its limit (see Fig. 4.11).

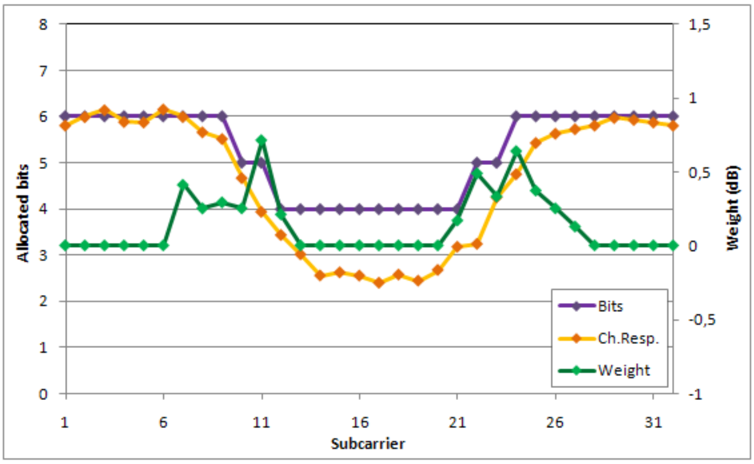


Figure 4.11: Bit allocation (with bit loading), assigned weights (with power loading) and channel response

This configuration does not add additional amplitude to distorted subcarriers as they have seen reduced their number of bits, reserving spare power in subchannels using the 64-QAM constellation (see Fig. 4.12).

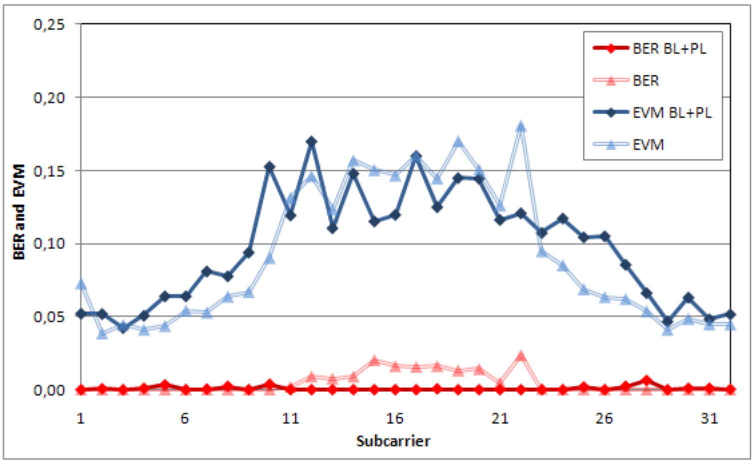


Figure 4.12: BER and EVM (bit and power loading)

The system bit error ratio is 0.09%. Although what is relevant is that the effective bit rate has increased 5% whereas with the second bit loading configuration this increase was of 1%.

It still requires devices which can respond faster as the number of bits per symbol and per carrier is six. But the resulting performance is really remarkable comparing with the first setup.

## 4.6. Comparison of the results

Finally, a table is presented showing the results obtained in the previous simulations for an easier comparison of the different scenarios (see Table 4.1).

| **Scenario** | **Bit rate variation** | **BER** | **BITSPERSYMBOL** |
|---|---|---|---|
| Constant bit and power | 0% | 0.48% | 5 |
| Constant bit and power (9 SC off) | –28% | 0.07% | 5 |
| Power loading | 0% | 0.13% | 5 |
| Bit loading (case 1) | –8% | 0.08% | 5 |
| Bit loading (case 2) | +1% | 0.09% | 6 |
| Bit and power loading | +5% | 0.09% | 6 |

Table 4.1: Comparison of the results obtained in the simulations

As it has been seen, the lack of adaptive loading techniques does not give too much margin to achieve the target BER. To disable nine subcarriers implies a 28% less bits per OFDM frame. Just using power loading, the bit error rate is reduced to almost meet the threshold without altering the amount of information modulated in each OFDM symbol.

The bit loading has a big impact on the performance of the system: losing an 8% of the effective bit rate against the 28% from disabling the subcarriers, and using the SNR gaps it has been possible to reach the same bit rate and even slightly increase it. And up to a 5% increase has been achieved when using power loading in conjunction with bit loading.

It has to be noted that all these results have been obtained without using algorithms that do many iterations. It has been done manually considering the channel state information extracted from VPI. This CSI is composed of the channel response per subchannel, the error vector magnitude per subchannel and the bit error rate per subchannel, programmed in MATLAB. So it is expected that algorithms could achieve even better results.

# CHAPTER 5. CONCLUSIONS AND FUTURE LINES OF WORK

In this thesis, a revision of the main principles of optical OFDM systems has been carried out with focus on IM/DD systems.

The basic characteristics of OFDM systems such as the subcarriers orthogonality, the cyclic guard time, pilot symbols and equalization have been reviewed. Special emphasis has been given to adaptive power and bit loading algorithms proven successful in fields like power line communications and DSL, and which are expected to provide gains in optical OFDM (OOFDM) systems as well.

The chromatic dispersion in single-mode fibers has been studied and characterized as the main limiting effect in optical OFDM systems. Spectral guard bands and SSB filters have been identified as solutions to avoid impairments related to chromatic dispersion.

From the various proposals for optical OFDM systems through optical fiber, the one based on electrical IQ up conversion prior to IM modulation and electrical down conversion after direct detection has been studied in more detail with the aid of a software for simulation of optical transmission systems, VPI.

A simulation setup that exploits the “cosimulation” capability offered by VPI has been developed. The OFDM coder and decoder functions have been programmed using MATLAB and integrated into VPI as blocks of the simulation. And the coder and decoder blocks have been provided with bit loading and power loading capabilities. The structure of the simulation setup, has been carefully designed to provide a user-friendly simulation platform. Universe variables have been defined that ease the data introduction and changing process, and the graphical results presentation devised allows for a quick and direct understanding of the problems simulated.

As test of use of the simulation platform, several examples of the improved transmission efficiencies offered by bit loading and power loading techniques have been presented. Even when the power and bit distribution has been made following intuitive approaches and no systematic optimization method has been employed, the improvements found are encouraging. A power loading example has shown a BER reduction from 0.48% to 0.13%. With bit loading, two different comparisons have been presented. One reduced the BER under 0.1% with an 8% loss of the effective bit rate instead of the 28% equivalent to disable the high error probability subcarriers with constant bit allocation. The other scenario took profit from the SNR gaps to decrease the BER from 0.48% to 0.09% and to have a 1% increase in effective bit rate. The last setup, with bit and power distribution, has shown an improvement in the system in terms of reliability with a bit error rate of 0.09%, and in terms of effective bit rate with a 5% increase. These results have been achieved without the use of automatized algorithms, so it is expected that they could obtain better results.

Finally, in the case of continuing this project, some future lines of work are suggested in form of possible enhancements to the current VPI design.

In this work preestablished random training sequences BPSK encoded have been used. It would be interesting to do some research on training sequences to find good symbol combinations that may estimate the channel more accurately. To achieve this, it is important to optimize the PAPR in the training sequence. It is also possible to add a moving average training [3] which is based on how the channel has changed over time as new training sequences have been received. Checking their evolution, between training frames, having an equalization that updates itself with the expected variation of the channel may improve the BER.

In spite of not using pilot tones because the phase noise is not very relevant in these simulations, to add them would make the system capable to detect the phase noise and compensate it. An additional information source for the channel estimation is also very positive to improve the effectiveness of the equalization. The BITL array parameter here defined could be used too indicating the position of pilot symbols with, for example, a -1 flag.

VPI is provided with scripting tools to run the simulations many times doing small modifications to the indicated parameters: it is originally intended to make optimizations and look for the best capable result under determined conditions in a universe. It uses a specific programming language and it is worth exploring how to adjust vectors like BITL and WEIGHT dynamically. With this option, an algorithm to find optimum bit and power distributions could be done in VPI.

# BIBLIOGRAPHY


[1] J. Armstrong, "OFDM for optical communications," *J. Lightwave Technol.*, vol. 27, pp. 189-204, Feb. 2009.

[2] S. L. Jansen, I. Morita, K. Forozesh, S. Randel, D. van den Borne, H. Tanaka, "Optical OFDM, a hype or is it for real?," *Optical Communication, 2008. ECOC 2008. 34th European Conference on*, pp.1-4, 21-25 Sept. 2008.

[3] S. L. Jansen, "SC341 OFDM for Optical Communications (Short Course OFC, 2010)."

[4] C. V. Arasu, P. Hyanki, H. L. Sharma, G. Lakshminarayanan, H. L. Moon, K. Seok-Bum, "PAPR reduction for improving performance of OFDM system," *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*, pp. 77-82, 7-9 Oct. 2010.

[5] T. Yucek, H. Arslan, "Time Dispersion and Delay Spread Estimation for Adaptive OFDM Systems," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 3, pp. 1715-1722, May 2008.

[6] X. Q. Jin, R. P. Giddings, J. M. Tang, "Experimental demonstration of adaptive bit and/or power loading for maximising real-time end-to-end optical OFDM transmission performance," *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, pp.1-3, 6-10 March 2011.

[7] M. F. Flanagan, "On Proving the Water Pouring Theorem for Information Rate Optimization."

[8] B. S. Krongold, K. Ramchandran; D. L. Jones, "Computationally Efficient Optimal Power Allocation Algorithms for Multicarrier Communication Systems," *Communications, IEEE Transactions on*, vol. 48, no. 1, pp. 23-27, Jan 2000.

[9] E. Giacoumidis, X. Q. Jin, A. Tsokanos, J. M. Tang, "Statistical Performance Comparisons of Optical OFDM Adaptive Loading Algorithms in Multimode Fiber-Based Transmission Systems," *Photonics Journal, IEEE*, vol. 2, no. 6, pp. 1051-1059, Dec. 2010.

[10] Q. Yang; W. Shieh; Y. Ma, "Bit and Power Loading for Coherent Optical OFDM," *Photonics Technology Letters, IEEE*, vol. 20, no. 15, pp. 1305-1307, Aug. 1, 2008.

[11] K. Hyeonmok, L. Kiseok, O. Seoungyoul, K. Cheeha, "Fast Optimal Discrete Bit-Loading Algorithms for OFDM-Based Systems," *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on,* pp. 1-6, 3-6 Aug. 2009.

[12] P. S. Chow, J. M. Cioffi, J. A. C. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," *Communications, IEEE Transactions on*, vol. 43, no. 234, pp. 773-775, Feb/Mar/Apr 1995.

[13] A. J. Goldsmith, S. G. Chua, “Variable-rate variable-power MQAM for fading channels,” *Communications, IEEE Transactions on*, vol. 45, no. 10, pp. 1218-1230, Oct 1997.

[14] T. N. Duong, N. Genay, M. Ouzzif, J. Le Masson, B. Charbonnier, P. Chanclou, J. C. Simon, “Adaptive Loading Algorithm Implemented in AMOOFDM for NG-PON System Integrating Cost-Effective and Low-Bandwidth Optical Devices,” *Photonics Technology Letters, IEEE*, vol. 21, no. 12, pp. 790-792, June 15, 2009.

[15] W. Deqiang, C. Yewen, Z. Laibo, “Efficient Two-Stage Discrete Bit-Loading Algorithms for OFDM Systems,” *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 7, pp. 3407-3416, Sept. 2010.

[16] R. F. H. Fischer, J. B. Huber, “A new loading algorithm for discrete multitone transmission,” *Global Telecommunications Conference, 1996. GLOBECOM ’96. ’Communications: The Key to Global Prosperity*, vol. 1, pp. 724-728 vol. 1, 18-22 Nov 1996.

[17] E. Heras Miguel, “Fiber-based orthogonal frequency division multiplexing transmission systems,” *Master Thesis*, October 27, 2010.

[18] A. Lowery, J. Armstrong, “Orthogonal-frequency-division multiplexing for dispersion compensation of long-haul optical systems,” *OPTICS EXPRESS*, vol. 14, no. 6, 2006.

# GLOSSARY

**ADC** Analog-to-digital converter

**BER** Bit error rate/ratio

**BL** Bit loading

**BPL** Bit and power loading

**BPSK** Binary phase-shift keying

**CP** Cyclic prefix

**CSI** Channel state information

**DAC** Digital-to-analog converter

**DD** Direct detected

**DML** Directly modulated laser

**DMT** Discrete multitone

**DSB** Double sideband

**DSP** Digital signal processor

**E-O** Electrical to optical

**EVM** Error vector magnitude

**FEC** Forward error correction

**FFT** Fast Fourier transform

**ICI** Intercarrier interference

**IFFT** Inverse fast Fourier transform

**IM** Intensity modulated

**IQ** In-phase / quadrature

**ISI** Intersymbol interference

**MMF** Multi-mode fiber

**MZM** Mach Zehnder modulator

**O-E** Optical to electrical

**OFDM** Orthogonal frequency division multiplexing

**OOFDM** Optical orthogonal frequency division multiplexing

**PAPR** Peak-to-average power ratio

**PL** Power loading

**PRBS** Pseudo random bit sequence

**PSK** Phase-shift keying

**QAM** Quadrature amplitude modulation

**QP** Quadrature point

**RF** Radiofrequency

**SMF** Single-mode fiber

**SNR** Signal-to-noise ratio

**SSB** Single sideband

**VPI** VPItransmissionMaker™ / VPIphotonics™

# APPENDIXES

# APPENDIX A. CODE USED IN THE DESIGNS

## A.1. Coder

```
function [y]=ofdm_coder_mcs(x1,TW,BR,BpS,Nc,N_FFT,CP,BitL,W,useWeight)

table_bpsk = [-1, 1, -1, 1, 1, -1, 1, -1, 1, -1, -1, 1, 1, -1, -1, 1,
    1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1];
table_bpsk2 = [-1, -1, 1, 1, -1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1,
    -1, 1, 1, -1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, -1];

% Generate normalize factors for peak power = 1
% a = zeros(1,9); for i=1:8 a(i) = modnorm(qammod((0:2^i-1), 2^i), 'peakpow', 1); end
% Normalization factors for modulations (position related to the level)
normfactor = [1.000000000000000, 0.707106781186547, 0.316227766016838, 0.235702260395516,
    0.141421356237310, 0.101015254455221, 0.064282434653323, 0.047140452079103];

%% START:    CARRIERS INFO
    % Reorder subcarriers (first the second half of the spectrum)
    BitL = [BitL(N_FFT/2 + 1:N_FFT) BitL(1:N_FFT/2)];
    % Reorder subcarriers (first the second half of the spectrum)
    W = [W(N_FFT/2 + 1:N_FFT) W(1:N_FFT/2)];

    carriersON = find(BitL ~= 0); % Index of enabled carriers
    BitLON = BitL(carriersON); % Get BpS of each enabled carrier without 0s from disabled carriers
    BpF = sum(BitL); % Bits per OFDM frame
    Weight = (W.^2)*Nc/sum(W(1,carriersON).^2); % Normalize weights (with weights of the enabled carriers)
    WeightON = Weight(1,carriersON); % Array of normalized weights for the enabled carriers
% # # # # # CARRIERS INFO

%% START:    OFDM INFO
    OFDM_LENGTH = ceil(N_FFT*(1+CP)); % Total QAM symbol length of OFDM symbol (including overheads)
    NTS_OFDM = floor(length(x1)/(BpS*OFDM_LENGTH)); % Total number of OFDM symbols in input sequence
    NTS_QAM = NTS_OFDM*OFDM_LENGTH; % Total number of QAM symbols, including overheads
    NTB_INFO = NTS_OFDM*BpF; % Total number of information bits [number of frames times bits per frame]
% # # # # # OFDM INFO

%% START:    BITS/SYMBOLS TO TX
    xx1=x1(1:NTB_INFO); % Vector containing the information bits to transmit

    x2 = reshape(xx1, BpF, NTS_OFDM)'; % Bits of information to TX with shape: [NTS_OFDM, BpF]
    xx1_QAM = zeros(N_FFT, NTS_OFDM); % Matrix of QAM symbols initialized to 0
    save C:\TMP\matlab_outputs\b1

    %% START:    Loop to make QAM symbols
        k = 1; % Bitload's pointer
        for n=1:Nc % Loop the enabled subcarriers (Nc)
            SC_Index = carriersON(n); % Get index of the current enabled subcarrier
            SC_bits = BitLON(n); % Get amount of bits for the current subcarrier
            % Get the whole bit sequence for a subcarrier
            SC_bitSq = x2(:, k:(k + BitL(SC_Index) - 1));
            % Generate QAM symbols modulating the decimal sequences
            xx1_QAM(SC_Index,:) = qammod(bi2de(SC_bitSq),2^SC_bits);
            % Subcarrier's factor to apply weight if it is enabled
            Weight_Factor = 1 + useWeight*(WeightON(n)-1);
            % Normalize and weight QAM symbols
            xx1_QAM(SC_Index,:) = xx1_QAM(SC_Index,:) * normfactor(SC_bits) * Weight_Factor;
            k = k + BitL(SC_Index); % Update bitload's pointer
        end
        save C:\TMP\matlab_outputs\b2
    % # # # # # Loop to make QAM symbols
```

```
    %% START:    Training sequence
        if N_FFT >= 32
            % Copy BPSK table N_FFT/16 times so it has N_FFT elements
            TSeq_Table = repmat(table_bpsk, 1, N_FFT/32);
            TSeq_Table2 = repmat(table_bpsk2, 1, N_FFT/32);
        else
            % Cut the sequence because there are less subcarriers
            TSeq_Table = table_bpsk(1:N_FFT);
            TSeq_Table2 = table_bpsk2(1:N_FFT);
        end

        table_training(1:2,:) = [TSeq_Table; TSeq_Table2]; % Create training sequence table
        % Training sequence (2 OFDM frames) at the beginning, QAM sequence
        % after cutting last 2 OFDM frames)
        xx1_QAM(1:end) = [reshape(table_training.', 1, 2*N_FFT) xx1_QAM(1:end-2*N_FFT)];
    % # # # # # Training sequence

    xx1_OFDM_INFO = reshape(xx1_QAM,N_FFT,NTS_OFDM); % Prepare the OFDM info
    xx1_IFFT = ifft(xx1_OFDM_INFO,N_FFT); % IFFT is applied
    % Cyclic prefix added to each OFDM symbol
    xx1_CP = [xx1_IFFT(1+N_FFT-ceil(N_FFT*CP):N_FFT,:);xx1_IFFT];
    xx1_serial = reshape(xx1_CP,1,NTS_QAM); % Serialize information
    y = zeros(1,TW*BR/BpS);
    y(1:NTS_QAM) = xx1_serial; % Output information

    save c:\TMP\matlab_outputs\txend
% # # # # # BITS/SYMBOLS TO TX
end
```

## A.2. Decoder

```
function [RX,RX_bits,BER,BERcarriers,EVM,EVMcarriers,CHresp,CHrespW,efBR,numBits]=
ofdm_decoder_mcs(yreal,yimag,TW,BR,SR,BpS,Nc,N_FFT,CP,BitL,TX_bits,Training,W,useWeight,orderSubcarriers)

table_bpsk = [-1, 1, -1, 1, 1, -1, 1, -1, 1, -1, -1, 1, 1, -1, -1, 1,
    1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1];
table_bpsk2 = [-1, -1, 1, 1, -1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1,
    -1, 1, 1, -1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, -1];

% Generate normalize factors for peak power = 1
% a = zeros(1,9); for i=1:8 a(i) = modnorm(qammod((0:2^i-1), 2^i), 'peakpow', 1); end

% Normalization factors for modulations (position related to the level)
normfactor = [1.000000000000000, 0.707106781186547, 0.316227766016838, 0.235702260395516,
    0.141421356237310, 0.101015254455221, 0.064282434653323, 0.047140452079103];

y = yreal + j*yimag; % Complex input symbol sequence. size: TW*BR/BpS
AGC = ones(1,Nc); % Automatic Gain Control per carrier

%% START:    CARRIERS INFO
    % Reorder subcarriers (first the second half of the spectrum)
    BitL = [BitL(N_FFT/2 + 1:N_FFT) BitL(1:N_FFT/2)];
    % Reorder subcarriers (first the second half of the spectrum)
    W = [W(N_FFT/2 + 1:N_FFT) W(1:N_FFT/2)];

    carriersON = find(BitL ~= 0); % Index of enabled carriers
    BitLON = BitL(carriersON); % Get BpS of each enabled carrier without 0s from disabled carriers
    BpF = sum(BitL); % Bits per OFDM frame
    % Normalize weights (with weights of the enabled carriers)
    Weight = (W.^2)*Nc/sum(W(1,carriersON).^2);
    WeightON = Weight(1,carriersON(1,:)); % Array of normalized weights for the enabled carriers
    BERcarriers = zeros(1, N_FFT); % Array of BER/carrier
    % Disabled carriers will be showing a BER of -1 to indicate its state
    BERcarriers(1, find(BitL == 0)) = -1;
    EVMcarriers = zeros(1, N_FFT); % Array of BER/carrier
    % Disabled carriers will be showing an EVM of -1 to indicate its state
    EVMcarriers(1, find(BitL == 0)) = -1;
    CHresp = zeros(1, N_FFT);
    CHrespW = zeros(1, N_FFT);
% # # # # # CARRIERS INFO

%% START:    OFDM INFO
    OFDM_LENGTH = ceil(N_FFT*(1+CP)); % Total QAM symbol length of OFDM symbol (including overheads)
    NTS_OFDM = floor(length(y)/(OFDM_LENGTH)); % Total number of OFDM symbols in input sequence
    NTS_QAM = NTS_OFDM*OFDM_LENGTH; % Total number of QAM symbols, including overheads
    numBits = BpF*NTS_OFDM; % Number of bits used for BER computation (bits simulated)
    efBR = numBits/TW; % Effective BitRate
% # # # # # OFDM INFO

%% START:    AUX VARS
    % Real information starts in the 3rd frame (2 first are training), and
    % last 2 are cut to have the same total number of OFDM frames
    FROM = 3;
    LAST = NTS_OFDM - 2;
% # # # # # AUX VARS
```

```
%% START:    BITS/SYMBOLS TO RX
    TX_bits_reshape = reshape(TX_bits(1:BpF*LAST),BpF,LAST); % Retrieve all the TX bits
    yy1 = y(1:NTS_QAM); % Remove single QAM symbols, consider only info+overheads

    % Move the last half CP at the start of string. This is related to fiber f_ref issues
    yy1_CP = [yy1(NTS_QAM-ceil(N_FFT*CP/2)+1:NTS_QAM),yy1(1:NTS_QAM-ceil(N_FFT*CP/2))];
    yy1_SP = reshape(yy1,OFDM_LENGTH,NTS_OFDM);
    yy1_CP_OUT = yy1_SP((ceil(N_FFT*CP)+1):size(yy1_SP,1),:); % Remove CP
    yy1_FFT = fft(yy1_CP_OUT,N_FFT); % FFT is applied

    %% START:    Automatic Gain Control (AGC) if Training Sequence
            % Set a vector with the expected received training sequence
            if N_FFT >= 32
                % Copy BPSK table N_FFT/16 times so it has N_FFT elements
                TSeq_Table = repmat(table_bpsk, 1, N_FFT/32);
                TSeq_Table2 = repmat(table_bpsk2, 1, N_FFT/32);
            else
                % Cut the sequence because there are less subcarriers
                TSeq_Table = table_bpsk(1:N_FFT);
                TSeq_Table2 = table_bpsk2(1:N_FFT);
            end

            table_training(1:2,:) = [TSeq_Table; TSeq_Table2]; % Create training sequence table

            save c:\TMP\matlab_outputs\y1

            % Get the channel response for the first and second OFDM symbols
            CHresp = ((yy1_FFT(:,1).') ./ table_training(1,:));
            CHrespW = ((yy1_FFT(:,2).') ./ table_training(2,:));

            if Training == 1
                % Gain per subcarrier (complex factors)
                % Average the gain from both training sequences (first two OFDM symbols)
                AGC = (1./CHrespW(carriersON))/2 + (1./CHresp(carriersON))/2;
            else % Training disabled, just a simple amplification for an appropriate qamdemod function
                % Average the gain from both training sequences (first two OFDM symbols)
                AGC = sum((1./abs(CHrespW(carriersON)))/2 + (1./abs(CHresp(carriersON)))/2)/Nc;
                AGC = repmat(AGC, 1, Nc);
            end
            CHresp = abs(CHresp); % Module of channel response because it is a complex value.
            CHrespW = abs(CHrespW); % Module of channel response because it is a complex value.
    % # # # # # Automatic Gain Control (AGC) if Training Sequence

    yy1_QAM = yy1_FFT(carriersON, :); % Get FFT data only for the active subcarriers

    %% START:    Loop to retrieve transmitted bits
        k = 1; % Bitload's pointer for the loop
        for n=1:Nc
            SC_Index = carriersON(n); % Index of the current enabled subcarrier
            SC_bits = BitLON(n); % Get amount of bits for the current subcarrier
            % Subcarrier's factor to undo the weight effect if it was applied
            Weight_Factor = 1 + useWeight*(WeightON(n) - 1);
            % Apply gain and undo weight to QAM symbols (these will be shown as constellation)
            yy1_QAM(n,:) = yy1_QAM(n,:) * AGC(n) / Weight_Factor;
            % Demodulate all normalized symbols from a subcarrier and store them in decimal
            yy1_decimal = qamdemod(yy1_QAM(n,:)/normfactor(SC_bits), 2^SC_bits);
            % Make the binary sequence, finally we will have a [NTS_OFDM, BpF] matrix (like coder)
            RX_bit_seq(:,k:(k + SC_bits - 1)) = de2bi(yy1_decimal', SC_bits);

            % START:    BER computation per carrier
            % All the TX bits by the carrier
            TX_bits_carrier = reshape(TX_bits_reshape(k:(k + SC_bits - 1),:), 1, SC_bits*LAST);
            % All the RX bits by the carrier
            RX_bits_carrier = reshape(RX_bit_seq(FROM:end, k:(k + SC_bits - 1))', 1, SC_bits*LAST);
            % XOR transmitted and received bits
            BER_carrier = xor(RX_bits_carrier, TX_bits_carrier);
            % Bit Error Ratio
            BERcarriers(SC_Index) = length(find(BER_carrier == 1)) / length(TX_bits_carrier);
            % # # # # # BER computation per carrier
```

```
            % START:    EVM computation per carrier
            % Make QAM transmitted symbols
            TX_symb_carrier = qammod(bi2de(TX_bits_reshape(k:(k + SC_bits - 1),:)'), 2^SC_bits).';
            % Normalize QAM transmitted symbols
            TX_symb_carrier = TX_symb_carrier * normfactor(SC_bits);
            % Error Vector Magnitude
            EVM_carrier = sqrt(sum(((real(TX_symb_carrier)-real(yy1_QAM(n,FROM:end))).^2+
                (imag(TX_symb_carrier)-imag(yy1_QAM(n,FROM:end))).^2))./
                sum((real(TX_symb_carrier).^2+imag(TX_symb_carrier).^2)));
            EVMcarriers(SC_Index) = EVM_carrier;
            % # # # # # EVM computation per carrier

            k = k + SC_bits; % Update bitload's pointer for the loop
        end
        save c:\TMP\matlab_outputs\y2
    % # # # # # Loop to retrieve transmitted bits

    % Serialize the bits excluding training sequence (if any)
    RX_bits = reshape(RX_bit_seq(FROM:end,:)',1,BpF*LAST);

    %% START:    Total BER & EVM computation
        BER = sum(BERcarriers(carriersON)) / Nc; % Bit Error Ratio
        EVM = sum(EVMcarriers(carriersON)) / Nc; % Error Vector Magnitude
    % # # # # # Total BER & EVM computation

    %% START:    Symbols for Constellation
        % Parallel to serial with/without training sequence
        yy1_QAM_serial = reshape(yy1_QAM(:,FROM:end),1,LAST*Nc);

        RX_symbol = zeros(1,TW*BR/BpS); % Preallocate space for the received symbols
        RX_symbol(1:LAST*Nc) = yy1_QAM_serial; % Get symbols' sequence

        % Fill the samples vector with repeated versions of the received symbol sequence
        SRfixed = floor(32*BR/BpS); % SR fixed because of the truncation of its tenths
        matrixSize = floor(TW*32*BR/BpS); % Avoid errors because of old SR (don't use SRfixed here)
        RX = reshape(ones(ceil(SRfixed/BR*BpS),1)*RX_symbol,1,matrixSize); % Symbols amplified by AGC
    % # # # # # Symbols for Constellation

    %% START: Order Subcarriers (if enabled)
        % Else the charts like BER and EVM per subcarrier or Channel Response
        % have the second half of the subcarriers at the beginning
        if orderSubcarriers == 1
            CHresp = [CHresp(N_FFT/2+1:N_FFT) CHresp(1:N_FFT/2)];
            CHrespW = [CHrespW(N_FFT/2+1:N_FFT) CHrespW(1:N_FFT/2)];
            BERcarriers = [BERcarriers(N_FFT/2+1:N_FFT) BERcarriers(1:N_FFT/2)];
            EVMcarriers = [EVMcarriers(N_FFT/2+1:N_FFT) EVMcarriers(1:N_FFT/2)];
        end
    % # # # # # Order Subcarriers (if enabled)
    save c:\TMP\matlab_outputs\rxend
% # # # # # BITS/SYMBOLS TO RX
end
```