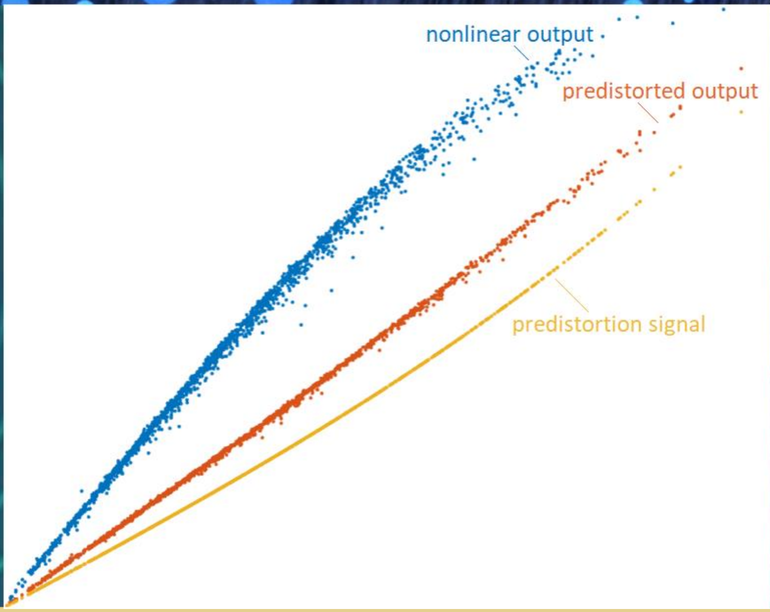
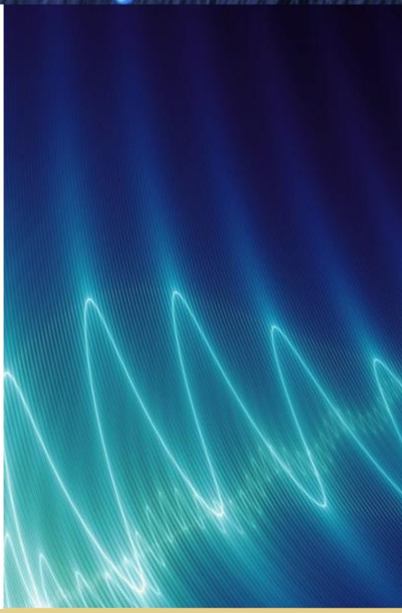




Beatriz Manata de Oliveira

Electro-Optic Interfaces for 60 GHz Networks

Dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering
September 2017

· U C ·

Universidade de Coimbra

FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Electrotécnica e de Computadores

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Interfaces Eletro-Óticas para Redes de 60 GHz

Beatriz Manata de Oliveira

Júri:

Presidente: Prof. Doutor Henrique José Almeida da Silva

Vogal: Prof. Doutor Luís Alberto da Silva Cruz

Orientadora: Prof. Doutora Maria do Carmo Raposo de Medeiros

Coimbra, Setembro de 2017

FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Electrotécnica e de Computadores

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Electro-Optic Interfaces for 60 GHz Networks

Beatriz Manata de Oliveira

Jury:

President: Professor Henrique José Almeida da Silva, PhD

Member: Professor Luís Alberto da Silva Cruz, PhD

Supervisor: Professor Maria do Carmo Raposo de Medeiros, PhD

Coimbra, September 2017

Agradecimentos

Em primeiro lugar, gostaria de agradecer à minha orientadora, Prof. Doutora Maria do Carmo Raposo de Medeiros, por entender que todos temos limitações e por me ensinar a ultrapassar as minhas, por ser a excelente professora que é e por me contagiar com o entusiasmo e gosto que tem pelas telecomunicações, por me inspirar e fazer querer saber sempre mais.

Agradeço ao Instituto de Telecomunicações – Pólo de Coimbra as condições de trabalho disponibilizadas e a atribuição de uma bolsa de investigação no âmbito do projecto interno “Photonic Technologies to support 5G/60GHz WLANS”.

Gostaria de agradecer também aos meus amigos e colegas pelo apoio que me deram ao longo do curso e pelos momentos de diversão. Em particular, gostaria de agradecer ao Daniel Costa por me apoiar nos momentos mais difíceis; ao André Silva, Bruno Fernandes, João Marques e Luís Parente pelas discussões que me ajudaram a compreender melhor várias matérias.

Agradeço também aos meus pais por me incentivarem a fazer as melhores escolhas, mesmo que seja o caminho mais difícil; por acreditarem que consigo ultrapassar todos os obstáculos e por me trazerem de volta à realidade nos objetivos irrealistas. Agradeço-lhes também pela independência e liberdade que me deram quanto aos estudos. Ao meu irmão, pela visão mais actual sobre os mundos da engenharia e tecnologia e pelas explicações ao longo de todo o meu percurso académico.

Resumo

Actualmente, as comunicações sem fios (wireless) estão a entrar numa nova fase, as redes de 5ª geração (5G), onde se espera o aumento de serviços que necessitam de grande largura de banda, nomeadamente, serviços interactivos e multimédia. Para suportar estas exigências, uma vertente da investigação em comunicações wireless estuda a operação na gama das microondas/ondas milimétricas para assim evitar o congestionamento espectral nas baixas frequências. Esta estratégia implica uma redução do tamanho da célula e, consequentemente, um aumento do número de células necessárias para cobrir uma determinada área. Tais redes exigem um número elevado de Estações de Base (BSs-Base Stations) para fazer a cobertura de uma área de serviço; assim, uma BS de baixo custo é a chave de sucesso no mercado. Esta exigência levou ao desenvolvimento de uma arquitectura de um sistema onde funções tais como o encaminhamento (routing)/processamento são tratadas numa Unidade Central (CU-Central Unit), mais do que na BS, tornando as BSs em unidade rádio remotas simples (RRHs-Remote Radio Head). A arquitectura de rede centralizada permite a equipamento sensível de ser localizado num ambiente mais seguro, para além de permitir a partilha dos componentes mais dispendiosos pelas várias RRHs. Uma estratégia atractiva para ligar uma CU com as RRHs é através de uma rede de fibra óptica, pois a fibra óptica tem baixas perdas e tem uma grande largura de banda. No entanto, os sinais de rádio são severamente distorcidos por estas ligações. A distorção, introduzida pelo sistema de transmissão ótico, será um fator limitante, uma vez que a potência dos sinais tem de ser reduzida para que os componentes operem na zona linear, diminuindo, assim, o alcance das ligações.

O espectro disponível torna-se uma limitação do aumento da taxa de transmissão, uma vez que a largura de banda não será suficiente para as redes de próxima geração. Uma banda que se encontra em estudo para responder a este desafio é a banda não licenciada nos 60 GHz, apesar das suas elevadas atenuações limitarem significativamente o alcance das ligações.

Nesta dissertação é implementado um modelo de simulação de um sistema rádio sobre fibra. As distorções sofridas pelos sinais rádio são identificadas, analisadas e compensadas com um bloco de pré-distorção digital baseado em polinómios de memória. A conversão eletro-ótica do sinal rádio multiplexado por divisão em frequências ortogonais (OFDM – Orthogonal Frequency Division Multiplexing) através do modulador Mach-Zehnder é destacada, uma vez que é a principal causa da distorção do sistema ótico.

Ao longo desta dissertação, os componentes e a arquitetura da rede são discutidos; modelos para o modulador Mach-Zehnder, fotodetector e fibra são apresentados. O ponto de polarização

DC do Mach-Zehnder é estudado, assim como a sua influência no desempenho do sistema; o número de símbolos e o débito binário são otimizados e o impacto do comprimento da fibra e a largura espetral do laser são também discutidos.

Palavras-Chave:

Fotónica de Microondas, Rádio sobre Fibra (RoF), OFDM, Compensação de distorção, Polinómios de Memória.

Abstract

Wireless communications are currently entering a new stage, 5th generation networks (5G), where an increase of services requiring high bandwidth is expected; namely, interactive services and multimedia. To support such demands, investigation in wireless communications has a strand studying operation at microwaves/millimeter waves to avoid spectral congestion in lower frequencies. This strategy implies a reduction in size of the cells, consequently leading to higher numbers of cells needed to provide coverage to a certain area. Such networks require a high number of Base Stations (BSs); thus, low cost BSs are the key to success in the market. This demand led to the development of an architecture of a system where functions such as routing/processing are dealt with at the Central Unit (CU), making BSs simple Remote Radio Heads (RRHs). This centralized network architecture allows sensible equipment to be localized at a safer environment while allowing resource sharing of the high cost components between RRHS. This strategy is attractive to connect a CU to RRHs trough optical fiber, since optical fiber has low losses and high bandwidth. However, radio signals are highly distorted by these connections. The distortion, introduced by the optical transmission system, will be a limiting factor, since the power of the signals must be reduced so the components operate at a linear zone, thus decreasing the range of the connections.

Available spectrum becomes a restraint in the increase of transmission rate, since the bandwidth will not be sufficient for next generation networks. The unlicensed 60 GHz band is being studied to meet this challenges, despite its high attenuation significantly limiting the range of the connections.

In this dissertation, a simulation model of a Radio over Fiber system is implemented. The distortions suffered by radio signals are identified, analyzed and compensated with a digital predistortion block based on memory polynomials. Electro-optic conversion of the Orthogonal Frequency Division Multiplexing (OFDM) radio signal is highlighted, since it is the main cause of distortion in the optical system.

Throughout this work, the components and architecture of the network are presented and discussed; the models for the Mach-Zehnder, optical fiber, photodetector and predistorter are shown. The impact on the performance of the bias point of the MZM is analyzed, the number of symbols and bit rate are optimized and the impact of the fiber's length and laser's linewidth are discussed.

Keywords:

Microwave Photonics, Radio over Fiber (RoF), Orthogonal Frequency-Division Multiplexing (OFDM), Distortion compensation, Memory polynomials.

Table of Contents

List of Figures

List of Tables

List of Acronyms

5G	5^{th} Generation
AM/AM	Amplitude/Amplitude
AM/PM	Amplitude/Phase
BBU	Baseband Unit
BER	Bit Error Rate
BS	Base Station
CO-OFDM	Coherent Orthogonal Frequency Division Multiplexing
CP	Cyclic Prefix
C-RAN	Cloud Radio Access Network
CU	Central Unit
CW	Continuous Wave
DC	Direct Current
DD-MZM	Dual Drive Mach-Zehnder Modulator
DD-OFDM	Direct Detection Orthogonal Frequency Division Multiplexing
DFT	Discrete Fourier Transform
E/O	Eletro-Optic
ED	Envelope Detector
ER	Extinction Ratio
EVM	Error Vector Magnitude
FEC	Forward Error Correction
FFT	Fast Fourier Transform
ICI	Inter-Carrier Interference
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IQ	In-phase/Quadrature
ISI	Inter-Symbol Interference
LMS	Least Mean Squares
LO	Local Oscillator
MIMO	Multiple Inputs, Multiple Outputs
MMSE	Minimum Mean Square Error
mm-Wave	Millimeter-Wave
MZM	Mach-Zehnder Modulator
O/E	Optic-Electric
OCS	Optical Carrier Supressed
ODSB	Optical Dual Side Band
OFDM	Orthogonal Frequency Division Multiplexing
OSSB	Optical Single Side Band
P/S	Parallel to Series
PA	Power Amplifier
PD	Photodetector
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
RLS	Root Least Squares
RMS	Root Mean Squares
RoF	Radio over Fiber
RRH	Remote Radio Heads
S/P	Series to Parallel

SMF	Single Mode Fiber

Chapter 1 - Introduction

This chapter discusses the topic and scope of this work. Section 1.1 contextualizes and sets the motivation of the dissertation. Section 1.2 presents the main objectives. Right after, Section 1.3 exposes the structure of this work and in last, Section 1.4 lists the major contributions.

1.1. Overview and Motivation

Consumer demands for high speed wireless networking video based interactive and multimedia applications are increasing. Today, such high bandwidth services are mostly supported by wired networks based on fiber optic technologies. However, users want to be able to use their mobile terminals and enjoy the same user experience as they do while connected at their fixed network. Wireless interfaces are already the dominant communication interface, it is foreseen that this trend will continue to increase as well as the traffic generated by them and consequently their interface transmission rate [1]. To provide the necessary capacity to wireless systems, Multiple Input Multiple Output (MIMO) and other technologies that result in network densification and high cost, have been developed [2], [3].

A solution to this bottleneck is seen in the development of wireless systems operating at much higher carrier frequencies in the millimeter-wave (mm-Wave) range where more bandwidth is available [4]. Present wireless technologies operate predominantly below the 10 GHz band; however, the abundant unlicensed spectrum around 60 GHz holds tremendous potential to upgrade wireless link throughput to Gbps level. There are already several standards defined for indoor wireless area networks (WLAN), for example IEEE 802.11ad, IEEE 802.15.3c and ECMA-387 enabling Gbps transmission. This advances also stimulate the interest of using the 60 GHz band for fifth generation (5G) mobile networks [5]. It is expected that 60 GHz technology will be a key player in 5G in the fronthaul and backhaul segments of the network. In some segments of the network, when the deployment of optical fiber infrastructures is not possible or it is very expensive, it is expected that high capacity point-to-point wireless links operating at millimeter wave including the 60 GHz band will be an essential part of the backhauling infrastructure [6], [7].

Nevertheless, the transmission of 60 GHz wireless signals is limited to a few meters (~10 m), which implies the deployment of multiple radio access points to cover a single house or building. Additionally, 60 GHz links are highly directional. The blockage effect and beam alignment are the main challenges for 60 GHz links which need multiple radio heads combined with cooperative beam steering strategies. During the last decade, much research work has been done on the suitability of radio-over-fiber (RoF) technology to support such distributed remote radio heads systems. RoF is an attractive technology to transport mm-Wave signals to remote places owing to the inherent advantages of optical link: reduced size, weight and power consumption, low attenuation, immunity to electromagnetic interference (EMI), and high bandwidth [8]. Additionally, photonics also provides a robust high quality approach to generate high bit rate 60 GHz signals [9].

A schematic of 60 GHz network supported by RoF technology is illustrated in Figure 1. Remote Radio Heads (RRHs) are connected by optical fibers to a Central Unit (CU). The RRH consist of just the optical to electrical (O/E), electrical to optical (E/O) transceivers (photo-receivers and lasers) amplification and antennas. All electronic equipment and radio processing devices will be centralized at the CU, thus leaving the architecture of the RRHs as simple as possible. This centralized architecture is compatible with the concept of Could Radio Access Network (C-RAN) also referred as Centralized-RAN proposed for 5th generation (5G) of broadband wireless networks [10]. C-RAN essentially designates a network architecture where several distributed remote radio heads (RRHs) with reduced complexity are linked to a base band unit (BBU) where the jointly radio signal processing and management is performed [2], [3].

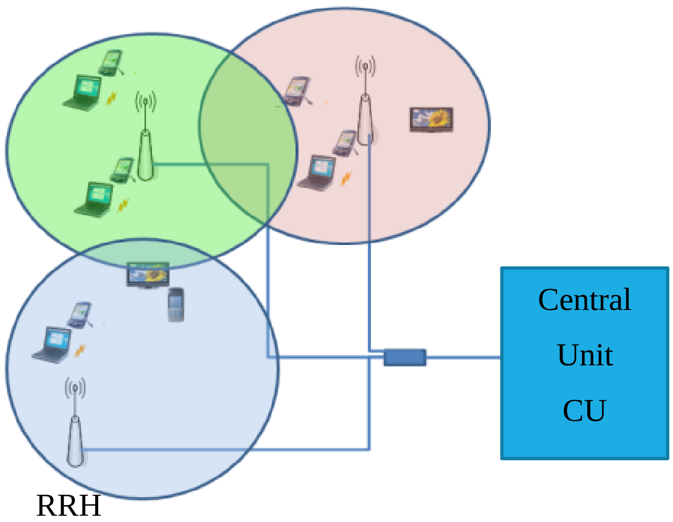


Figure 1 - 60 GHz network supported by RoF.

Although photonic technology adds flexibility to generate and distribute high bit rate 60 GHz signals, it also contributes to increase the nonlinear distortion of the overall wireless photonic link. Radio signals particularly Orthogonal Frequency Division Multiplexing (OFDM) signals with high bit data rate are very sensitive to the non-linear distortion of the RF amplifiers. When OFDM signals are transmitted through analog photonic links, these signals suffer additional nonlinear distortion due to nonlinear distortion of the analog photonic link.

1.2. Objectives

The objective of this dissertation is to study and compensate the nonlinear distortions added to an OFDM radio signal by the photonic link. The focus of the work is the downlink, i.e. signal transmission from the CU to the RRH. Photonic generation of the 60 GHz signal by externally modulated a continuous wave (CW) laser by means of a dual drive Mach-Zehnder modulator (DD-MZM) is considered.

The main goal of this dissertation is the development of pre-distortion digital schemes to overcome the distortion introduced by the E/O conversion of the OFDM radio signal, optical fiber transmission and O/E conversion. Thus, more power can be sent without distortion which can increase the range of the RoF links.

The impact of those impairments in the system performance as well as the efficiency of the pre-distortion strategies develop are also investigated.

1.3. Structure

Following this introduction, Chapter 2 will introduce the system's architecture that will be considered and introduces its main building blocks.

Chapter 3 describes the RF subsystem; namely, an overview on OFDM, the emitter and different receivers' architectures are presented.

In Chapter 4, predistortion compensation is discussed. Memory polynomials are analyzed and the predistorter's architecture is presented.

Chapter 5 shows the discussion of the results and analysis on the performance of the system when several parameters are changed.

Finally, in Chapter 6, the conclusion and future work suggestions are presented.

1.4. Contributions

This work was developed in Instituto de Telecomunicações in Coimbra, under a research grant from FCT (Fundação para a Ciência e Tecnologia) for the project Pho-Tech 5G/60GHz.

The contributions of this work can be divided in conference articles and models implemented using Matlab.

The results of this work were presented in the following conference articles:

[1] M.C.R. Medeiros, B.M. Oliveira ,P. Almeida, H.J.A. Silva and P.M. Monteiro, 'Ultra-high bit rate mm-wave photonic-wireless links employing digital equalization'. III International Conference on Applications of Optics and Photonics, (p. 1). Faro, Portugal, 2017.

[2] M.C.R. Medeiros, P. Almeida, B.M. Oliveira, P. Laurêncio and P.M. Monteiro, ‘Wireless-optical transceiver architectures for 60 GHz LANs’, *International Conf. on Transparent Optical Networks - ICTON,* Trento, Italy, Vol. 1, pp. We.D2.1 - We.D2.1, July, 2017. In Press

The models developed are presented below.

[1] Matlab function to store input and output data from VPI Photonics in ".mat" file

[2] Matlab function to load data from file and compute the optimum parameters for the coefficients of the memory polynomial to be used in the predistortion

[3] Matlab function that returns the predistortion signal computed from the input of the system in VPI Photonics

[4] Matlab function that computes the memory matrix from the input vector

[5] Matlab function to estimate the coefficients of the memory polynomial. Minimum Squares, Least Mean Squares and Root Least Squares algorithms can be used according to an input parameter.

[6] Optical filter implemented in Matlab, with bandwidth, center frequency and stopband gain as inputs. The filter is relative to an optical frequency of reference.

[7] Electrical filter implemented in Matlab, with bandwidth, center frequency and stopband gain as inputs. The filter is symmetric and does not include delay nor phase distortion.

[8] Matlab function that estimates FEC limit in EVM from the FEC limit in BER, with the number of bits per symbol as input and the EVM format (dB, percentage or linear) and the value of BER as parameters.

[9] Matlab function that estimates BER from EVM, with the value of EVM and the number of bits per symbol as inputs and the EVM format (dB, percentage or linear) as parameter.

Chapter 2 - 60 GHz Systems Supported by RoF

In this chapter, the elements that will be part of the optical system will be presented in detail.

Section 2.1 shows the limitations of operating at the 60 GHz bandwidth.

Sections 2.2 and 2.3 present different architectures for RoF: analog and digital, respectively.

Section 2.4 presents the E/O elements and possible modulation formats.

Sections 2.5 and 2.6 show the generation of optical carriers, without and with modulation of an OFDM signal, respectively.

Section 2.7 presents the model for optical transmission through fiber.

Finally, Section 2.8 presents O/E conversion through photodetector.

2.1. Limitations of 60 GHz

A major limitation of 60 GHz wireless networks is their short range due to the high-atmospheric propagation attenuation, which can reach around 14 dB/km, as shown in Figure 2. Additionally, 60 GHz radio signals cannot cross walls. Consequently, their indoor coverage is limited to a single room and to small areas, around 10 m radius [11].

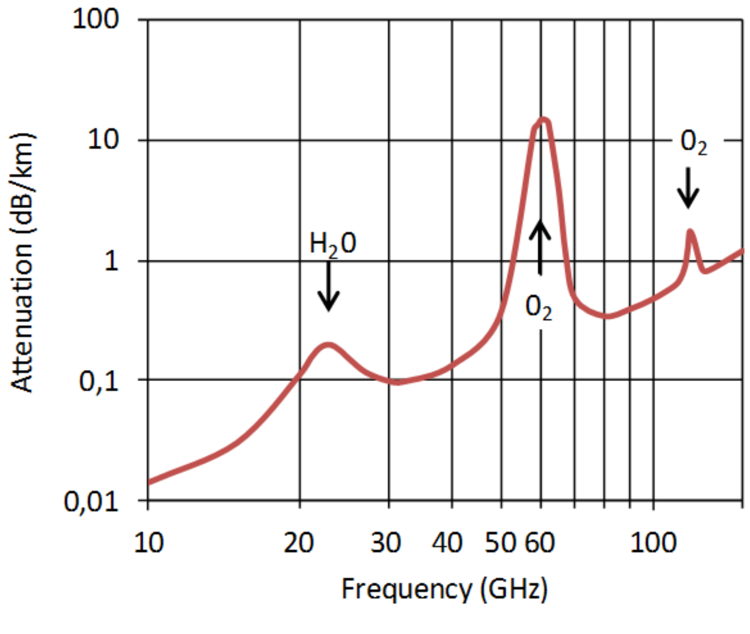


Figure 2 - Atmospheric attenuation at sea level with 20º C temperature, based on [12].

As a consequence, a large number of RRHs are necessary to cover an operational geographical area. Each RRH will need high bandwidth connections to the rest of the network which can be provided by radio over fiber technology. RoF consists in transporting the radio signals by means of an optical carrier from the remote site to network CU where it is processed.

2.2. Analog RoF

In analog RoF, an optical carrier is directly modulated by the wireless signals, which are transmitted in their native format. Therefore, no up/down RF conversion or digital signal processing is necessary at the RRHs which eliminates extra processing delays. All electronic equipment and radio processing devices can be centralized at the CU, thus leaving the architecture of the RRHs as simple as possible. However, analog RoF inherently suffers from intermodulation distortions arising from the nonlinearities of both RF and optical components. Furthermore, it has been shown that the dynamic range of the analog optical link depends on the optical modulation that is used and on optical fiber dispersion [13].

2.3. Digital RoF

Contrary to what happens in analog systems, by employing digitalization of wireless signals in optical transport schemes, it is possible to maintain dynamic range of the signal irrespective of the optical fiber length, provided that the received signal amplitude is above the sensitivity of the link [14]. However, the digitalization of wireless signals results in bit rates that are very high compared to the real data rate available to the user equipment. The total capacity requirement for a RRH backhaul is the sum of the total user data available at the cell site once all available channels are considered and including all sectors plus the control and signaling overhead. As an example, digitalization of a 100 MHz LTE-advanced channel, if all the features are enabled, may result in a backhaul traffic requirement up to 100 Gbit/s per RRH [15]. This dissertation will focus on analog RoF.

2.4. Electrical to optical conversion of the wireless signal

Electrical to Optical (E/O) conversion can be accomplished by external modulation of an optical carrier by means of lithium niobate (LiNbO3) MZM. This method has proven to be reliable due to the frequency range and excellent system stability. In this dissertation, a dual drive MZM (DD-MZM) is used for this conversion. Table 1 summarizes the configurations used in the DD-MZM to achieve different modulation formats that will be discussed below.

2.4.1. Dual Drive Mach-Zehnder Modulator

The basic principle of DD-MZM is very simple: an input optical field coupled to two waveguide branches fabricated from the same material and with the same length. An electrical signal can be applied on each arm of the D-MZM separately. When the electrical signal that is applied to the MZM arms changes, the effective refractive index of the waveguide will change; this will change the phase of the optical field. Thus, the phase delay of optical field, in each arm of D-MZM can be controlled by the external electrical signal field. The two optical fields at the output of each arm are then coherently added together by a coupler.

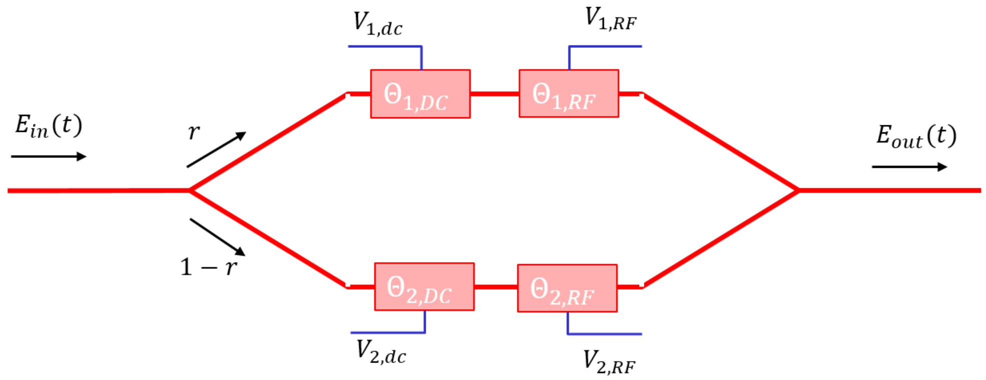


Figure 3 - Schematic of the Mach-Zehnder Modulator

Figure 3 shows a schematic of the Dual Drive Mach-Zehnder (DD-MZM), where r represents the splitting ratio, d_1 and d_2 represent the voltage applied to each arm and V_π is the voltage needed to introduce a phase shift of π (the switching voltage).

$$E_{out} = \frac{E_{in}}{2}\left(\mathrm{r}\times\exp\left(j\frac{\pi \mathrm{d}_1}{\mathrm{V}_\pi}\right) + (1-\mathrm{r})\times\exp\left(j\frac{\pi \mathrm{d}_2}{\mathrm{V}_\pi}\right)\right) \quad (1)$$
$$\mathrm{d}_i = \mathrm{V}_{i,DC} + \mathrm{V}_{i,RF}, \qquad i = 1,2$$

where d_1 and d_2 are the drive voltages applied to arms one and two, respectively. d_i is composed by a DC component $\mathrm{V}_{i,DC}$ and by the radio signal $\mathrm{V}_{i,RF}$. $E_{in}(t) = \sqrt{2P_o}e^{j\omega_c t}$ is the optical field at the input of DD-MZM with angular frequency ω_c and average power P_0. $\mathrm{V}_{i,DC}$ can be adjusted to generate different optical modulation formats: Optical Single Side Band (OSSB), Optical Double Side Band (ODSB) and Optical Carrier Suppressed (OCS).

To illustrate the generation of the different modulation formats in the following sections, we consider that the applied radio signal is a pure carrier at angular frequency ω_m, plus a DC component, $X_{RF}(t) = V_m sen(\omega_m t)$, *r=0.5*, and define the modulation depth defined as: $x_c = \frac{V_m}{V_\pi}$.

2.4.2. Optical Double Side Band Modulation

An ODSB signal can be obtained by applying the following electrical signals to the electrodes of the dual arm MZM:

$$d_1(t) = X_{RF}(t) - \frac{V_\pi}{4}$$
$$d_2(t) = -X_{RF}(t) + \frac{V_\pi}{4} \quad (2)$$

Being the optical field at modulator output, $E_{ODSB}(t)$:

$$E_{ODSB}(t) = \sqrt{2P_o}e^{j\omega_c t}cos\left(\frac{\pi X_{RF}(t)}{V_\pi} - \frac{\pi}{4}\right) \quad (3)$$

$E_{ODSB}(t)$ can be written as a series of Bessel functions, as Equation (4) shows:

$$E_{ODSB}(t) = \sqrt{2P_o}e^{j\omega_c t}\sum_{n=-\infty}^{\infty} cos\left(\frac{\pi}{4} + n\frac{\pi}{2}\right)J_n\left(m_I/\sqrt{2}\right)e^{jn\left(\omega_m t + \frac{\pi}{2}\right)} \quad (4)$$

where $J_n(.)$ is the n_{th} order Bessel function of the first kind and m_I is the modulation index defined as $m_I = \sqrt{2}\pi x_c$. In Figure 4-a, the DD-MZM is presented with the necessary configuration for generating an ODSB signal while Figure 4-b illustrates a representative optical spectrum at the output of the DD-MZM.

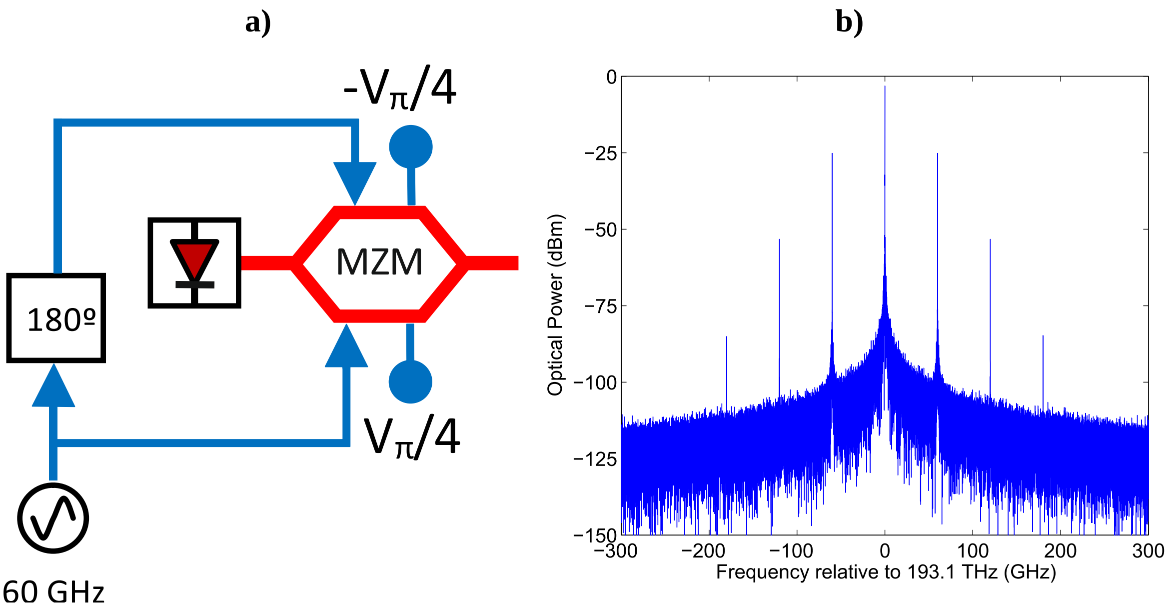


Figure 4 - a) MZM with ODSB configuration; b) ODSB spectrum at the MZM output.

2.4.3. Optical Single Side Band Modulation

In a similar fashion, an OSSB signal can be obtained by applying the following electrical signals to the electrodes of the dual arm MZM:

$$\begin{aligned} d_1(t) &= X_{RF}(t) - \frac{V_\pi}{4} \\ d_2(t) &= \widehat{X_{RF}}(t) + \frac{V_\pi}{4} \end{aligned} \tag{5}$$

where $\widehat{X_{RF}}(t)$ represents the signal $X_{RF}(t)$ with a 90º phase shift. The interaction between the RF modulation and the optical signals results in the suppression of one of the odd-harmonics modulation sidebands, as shown in Figure 5.

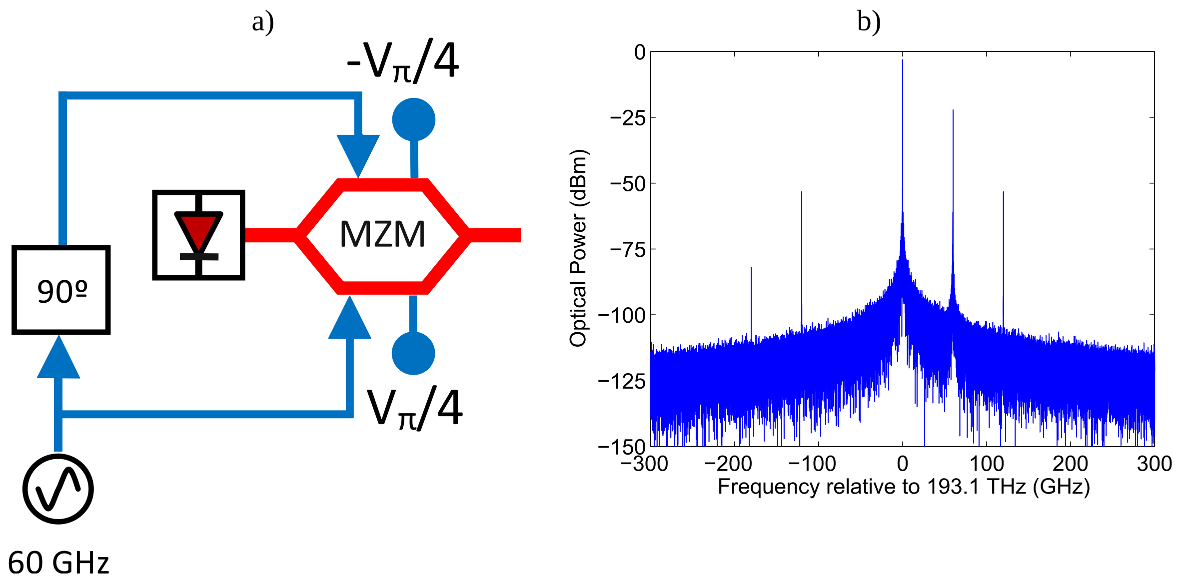


Figure 5 - a) MZM with OSSB configuration; b) OSSB signal at the MZM output.

2.4.4. Optical Carrier Suppression

An OCS signal can be obtained by applying the following electrical signals to the electrodes of the dual arm MZM:

$$\begin{aligned} d_1(t) &= X_{RF}(t) - \frac{V_\pi}{2} \\ d_2(t) &= -X_{RF}(t) + \frac{V_\pi}{2} \end{aligned} \tag{6}$$

At the output of the MZM, the optical carrier will be suppressed and a double-sideband-suppressed carrier optical signal will be generated as shown in Figure 6.

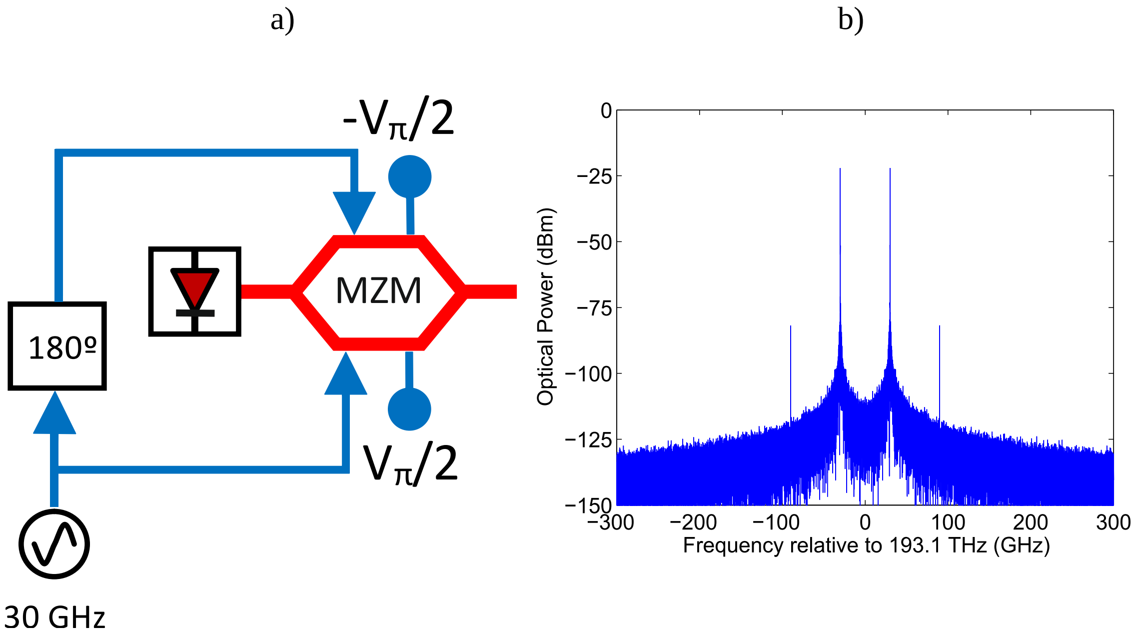


Figure 6 - a) MZM with OCS configuration; b) OCS signal at the MZM output

The DD-MZM is highly nonlinear when driven by a RF sinusoidal signal. The output optical spectrum consists of a number of harmonics spaced by the frequency of the RF frequency whose amplitudes depend on the modulation index. When the modulation index increases, several Bessel products become visible in the output spectrum for higher amplitudes. These components are undesirable and thus an optical filter should be applied at the output of the Mach-Zehnder.

2.5. Optical Generation of 60 GHz Carriers

Coherent heterodyning is a well-known technique to generate low phase noise mm-wave by beating of two coherent optical tones, $E_1(t) = \sqrt{2P_1}e^{j(2\pi v_1 t + \theta_1(t))}$ and $E_2(t) = \sqrt{2P_2}e^{j(2\pi v_2 t + \theta_2(t))}$, where P_1 and P_2 are the respective optical powers, v_1 and v_2 the optical frequencies, $\theta_1(t)$ and $\theta_2(t)$ the instantaneous phases [16]. Electrical signals can be generated by coherent mixing of the two optical tones upon photodetection by photodiode (PD) with responsivity $\mathcal{R}$, the generated photocurrent $I_{PD}(t)$ is given by,

$$I(t) = \mathcal{R}|E_2(t) + E_1(t)|^2 \propto e^{j(2\pi(v_2 - v_1)t + \theta_2(t) - \theta_1(t))} + \text{other terms} \tag{7}$$

The mixing process leads to the generation of new components at the sum and difference of the two input frequencies, whereby the difference in frequency is set to be at the required mm-Wave carrier frequency, as shown in Figure 2. However, this process is affected by the phase noise of the optical carriers and therefore only correlated optical sources upon photodetection produce spectrally pure mm-Wave carriers.

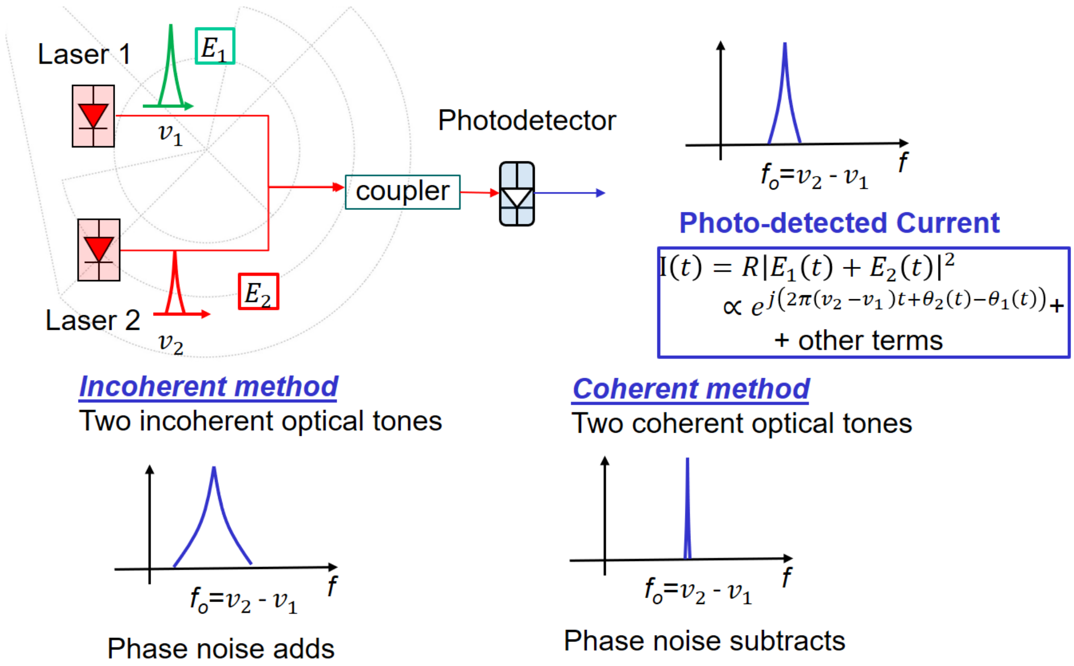


Figure 7 - Photonic mm-Wave generation. Source: [17]

2.6.Optical Generation of 60 GHz Carriers Modulated with Wireless Signal

To impose the wireless OFDM signal into the optical domain, one of the optical carriers is modulated with an OFDM signal located at the intermediate frequency, $f_{IF,OFDM}$, by means of a dual drive Mach-Zehnder modulator (DD-MZM) as illustrated in the schematic of Figure 8.

Similarly to Figure 7, CW Laser 1 generates an optical wave at the optical frequency ν_1 and CW Laser 2 generates another optical wave at optical frequency ν_2, different from ν_1.

The first optical tone is modulated with an OFDM signal in RF using a Dual Drive Mach-Zehnder Modulator (DD-MZM). The output of the DD-MZM is then filtered with an optical band pass filter. The aim of this filter is to remove the products of the Bessel function. The second optical tone is not modulated.

After photodetection, the electrical signal is centered at $f_{IF,Rx} = \nu_2 - \nu_1 + f_{IF,OFDM}$, where $f_{IF,OFDM}$ is the RF frequency of the OFDM signal.

Figure 8 also shows the resultant spectrums at several key points for OSSB. Figure 8-c is particularly important: the OFDM signal is in fact centered at $f_{IF,Rx}$, but there is a copy present at $f_{IF,OFDM}$ that will need filtering at the reception with Envelope Detection.

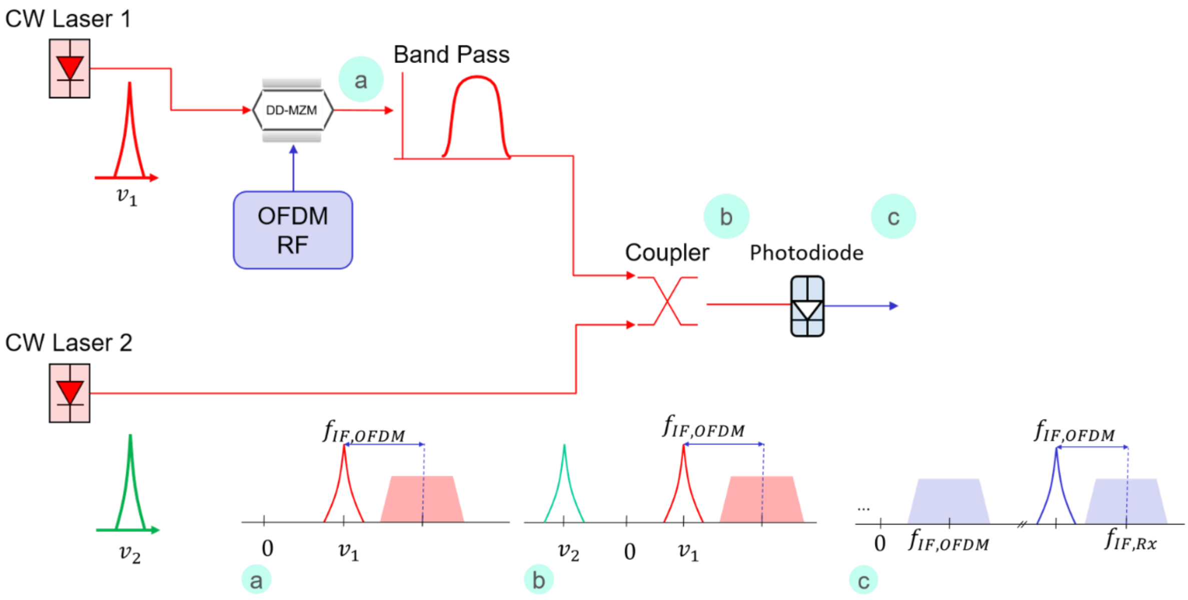


Figure 8 - Block Diagram of the system in the optical domain

As Table 1 shows, different modulation schemes can be used depending on the voltages applied to the electrodes of the DD-MZM. x_{RF} represents the OFDM signal at RF with data to modulate the optical input.

Figure 9 shows the curve of the intensity transfer function[1] for the Mach-Zehnder and how each modulation scheme is biased.

	Θ_1		Θ_2	
	DC	RF	DC	RF
OSSB Optical Single Side Band	$V_\pi/4$	x_{RF}	$-V_\pi/4$	$H\{x_{RF}\}$
ODSB Optical Dual Side Band	$V_\pi/2$	x_{RF}	$-V_\pi/2$	0
OCS Optical Carrier Suppressed	$V_\pi/2$	x_{RF}	$-V_\pi/2$	$-x_{RF}$

Table 1 - Modulation schemes for the Mach-Zehnder

[1] The intensity transfer function is showed in the appendix

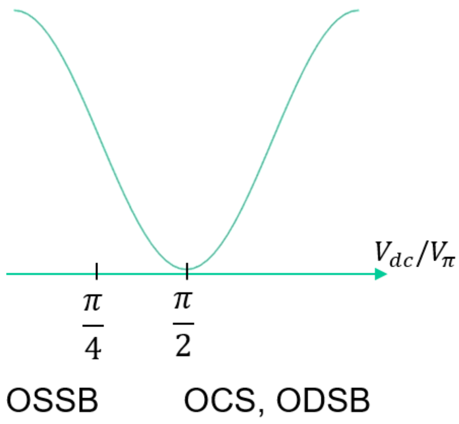


Figure 9 - Bias point for the modulation schemes. OSSB: Optical single Side Band, OCS: Optical Carrier Suppressed, ODSB: Optical Dual Side Band

2.7.Optical Transmission

Transmission through fiber can be modelled using a transfer function of the fiber.

In this work, a Single Mode Fiber (SMF) model was used. Equation (8) shows the implemented transfer function for such fiber, where G represents the fiber total gain in linear units and θ_{fiber} is the total phase shift introduced by the fiber. In equation (9), α is the fiber attenuation per unit of fiber length, in dB/Km and L is the fiber's length, in Km. In Equation (10), D represents the fiber dispersion in s/m^2, λ is the center wavelength in meters, c is the speed of light in vacuum in m/s and f is the frequency of the input signal in Hz.

$$H_{fiber} = G \times e^{\theta_{fiber}} \quad (8)$$

$$G = 10^{-\alpha \times \frac{L}{20}} \quad (9)$$

$$\theta_{fiber} = \frac{\pi D(\lambda \times f)^{2(L \times 1e3)}}{c} \quad (10)$$

2.8.Optical to electrical conversion of the wireless signal

Optical to Electrical conversion (O/E) is achieved with a photodetector. In this dissertation, a PIN Photodetector is used for direct detection of the signal.

The model of the photodetector was introduced in Section 2.5. In this section, we focus on the "other terms" in the photodetected current.

There are several sources of noise in photodetectors. Equation (11) shows the output current of a PIN photodetector.

For instance, the dark current, I_d, is the constant value of output current when there is no incident signal.

Another type of noise is the thermal noise which is always present in electronic devices.

Lastly, the shot noise is caused by random emission of photons.

The random currents produce White Gaussian Noise and thus are easily modelled.

$$I(t) = I_{ph}(t) + I_d + n_{th}(t) + n_{sh}(t) \tag{11}$$

Chapter 3 - RF subsystem: Operation principles and Modelling

In this chapter, we focus on the RF electrical signal. First, an introduction to OFDM is presented. Section 3.2 presents the block diagram of the RF subsystem, for both Coherent and Envelope Detection schemes.

3.1. Overview on OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a technique used to achieve higher data rates. It is a particular case for multi-carrier systems where the frequency bands overlap, thus allowing savings in spectrum usage close to 50% [18]. Figure 10 shows the main difference between conventional multi-carrier signals and OFDM.

The subcarrier spacing is defined so that the channels are mutually orthogonal. The orthogonality is the key to fight Inter-Carrier Interference (ICI), and it is achieved by defining the subcarrier spacing as $\Delta f = 1/T_{sym}$, where T_{sym} is the length of an OFDM symbol[2]. However, this is also the reason OFDM systems are highly sensitive to frequency offsets.

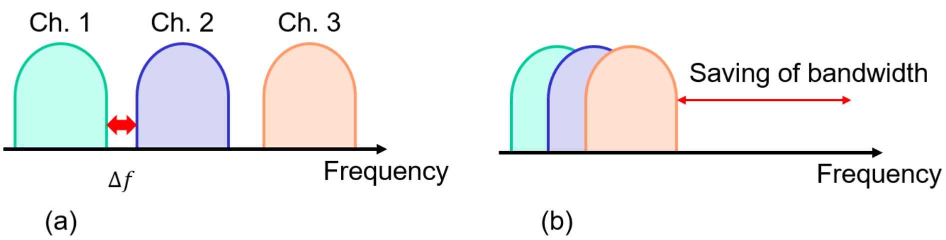


Figure 10 - Multiple Carrier Systems and bandwidth usage. a) conventional parallel system. b) OFDM. Source [18]

An OFDM system transmits several PSK or QAM symbols in parallel. For this reason, $T_{sym} = NT_s$ where T_s is the duration of a PSK or QAM symbol and N is the number of subcarriers. The l^{th} OFDM signal for the k^{th} subcarrier can be expressed as a function of time as Equation (12) shows. The subcarrier's frequency is given by $f_k = k/T_{sym}$, where $k = 0,1..N-1$ and $l =$

[2] Orthogonality in OFDM is studied in the appendix.

0,1..∞. The l^{th} symbol transmitted by that subcarrier is denoted by $X_l[k]$. Thus, the total bandpass signal is expressed in Equation (13) and the baseband signal in Equation (14).

$$\Psi_{l,k}(t) = \begin{cases} e^{j2\pi f_k(t-lT_{sym})}, 0 < t \leq T_{sym} \\ 0, \quad otherwise \end{cases} \tag{12}$$

$$x_l(t) = Re\left\{\frac{1}{T_{sym}}\sum_{l=0}^{\infty}\sum_{k=0}^{N-1} X_l[k].\Psi_{l,k}(t)\right\} \tag{13}$$

$$x_l(t) = \frac{1}{T_{sym}}\sum_{l=0}^{\infty}\sum_{k=0}^{N-1} X_l[k]e^{j2\pi f_k(t-lT_{sym})} \tag{14}$$

When sampling at $t = lT_{sym} + nT_s$, the discrete-time signal becomes the N-point IDFT of the OFDM symbol, as Equation (15) shows. Thus, modulation techniques can be applied using the IFFT.

$$x_l[n] = \sum_{k=0}^{N-1} X_l[k]e^{j2\pi kn/N} \tag{15}$$

Identically, the demodulation of an OFDM symbol is equivalent to the DFT and can be implemented using the FFT. Denoting the received symbol as $y_l[n]$, the received signal is given by Equation (16).

$$Y_l[k] = \sum_{n=0}^{N-1} y_l[n]e^{-j2\pi kn/N} \tag{16}$$

Thus, the OFDM modulation and demodulation can be implemented as Figure 11 shows.

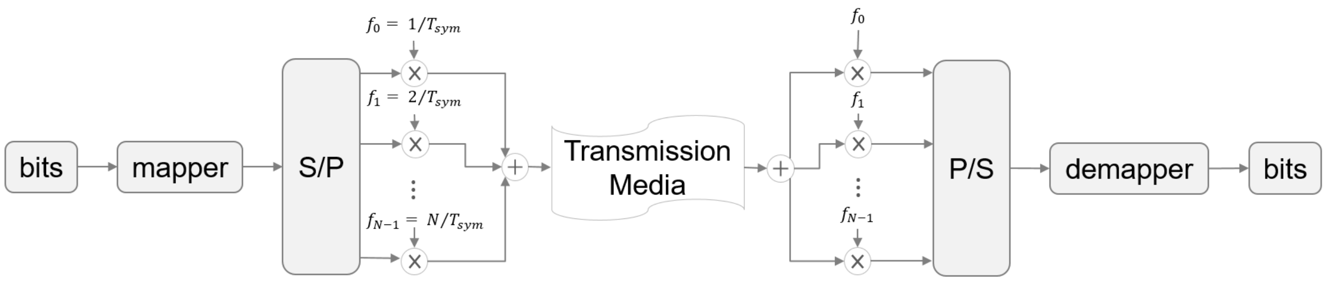


Figure 11 - OFDM Modulation/Demodulation. Source: [19]

3.2. OFDM Emitter and Receptor

The generation of an OFDM signal was discussed in the previous section. In this section, the generation of such signal is contextualized in the RF subsystem and the emitter and receiver schematics are presented.

3.2.1. OFDM Emitter

The previous section presented the necessary operations for an OFDM emitter. These operations are shown in Figure 13 and Figure 14.

The emitter is constituted by the OFDM generation, pulse shaping and RF conversion. Pulse shaping and RF conversion through IQ modulation are presented next.

Raised Cosine Filter

After the IFFT and Parallel to Series conversion (P/S), the OFDM signal is shaped: a square root raised cosine filter is applied to reduce Inter-Symbol Interference (ISI).

Equation (17) shows the frequency response of this filter, where T represents the symbol time, β is the roll-off factor ($0 \leq \beta \leq 1$) and f is the frequency vector. This equation also shows that the roll-off factor plays a big part in the shape of the filter as well as its bandwidth. Figure 12-a shows that as β increases, the bandwidth increases and the shape of the filter goes from rectangular ($\beta = 0$) to cosine ($\beta = 1$). The impulse response of the filter is presented in Figure 12-b. The response of the filter is null at multiples of T, thus eliminating ISI.

The bandwidth of the filter is given by Equation (18).

$$H(f) = \begin{cases} 1, & |f| \leq \frac{1-\beta}{2T} \\ \frac{1}{2}\left[1 + \cos\left(\frac{\pi T}{\beta}\left[|f| - \frac{1-\beta}{2T}\right]\right)\right], & \frac{1-\beta}{2T} < |f| \leq \frac{1+\beta}{2T} \\ 0, & otherwise \end{cases} \quad (17)$$

$$BW = \frac{\beta + 1}{T} \quad (18)$$

In real-life applications, the pulse-shaping filter used is $\sqrt{H(f)}$. The Square Root Raised Cosine is used immediately before and after transmission to reduce the impact of white noise. The overall filtering applied to the signal to be modulated is thus $\sqrt{H(f)}.\sqrt{H(f)} = H(f)$.

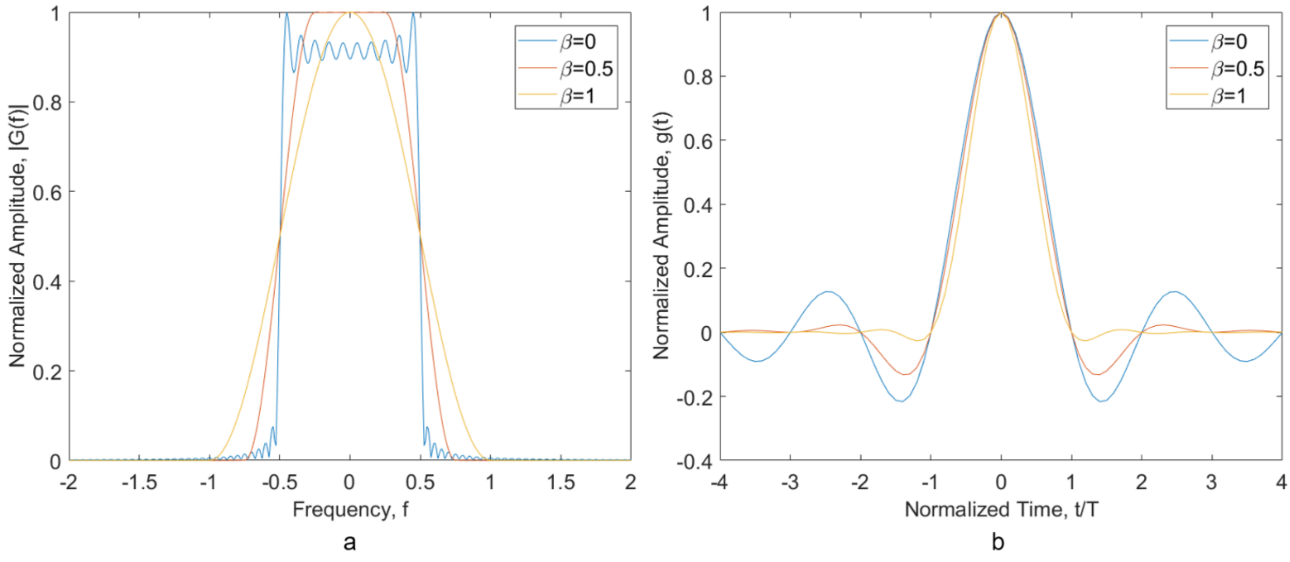


Figure 12 - Raised Cosine Filter. a) Frequency response. b) Impulse response.

I/Q Modulation

Figure 13 and Figure 14 show that after pulse-shaping, two copies of the signal are multiplied by different functions. The upper arm is multiplied by a cosine function at frequency f_{IF}, resulting in the in-phase component (I). The lower arm is multiplied by a sine function at frequency f_{IF}, resulting in the quadrature component (Q).

These components are then added and transmitted through the optical system.

3.2.2. OFDM Receiver

Figure 13 and Figure 14 present different receiver architectures, depending on the detector implemented. This section introduces OFDM detectors while comparing their performance.

Overview on OFDM Detectors

The detection of an optical OFDM signal can be made using Direct-Detection (DD-OFDM) [20]–[22], Coherent Detection (CO-OFDM) [20], [21] or Envelope Detection (ED) [23].

Direct-Detection of an optical OFDM signal is implemented by simply photodetecting the optical signal.

In CO-OFDM, baseband conversion is achieved by multiplying the photodetected signal with a Local Oscillator (LO), a cosine function oscillating at the center frequency of the OFDM photodetected signal.

In Envelope Detection, the absolute value of the photodetected signal is squared, which introduces several copies of the OFDM signal at different frequencies. The signal is brought to an intermediate frequency: the frequency of the original electrical RF OFDM signal. A LO is then used to convert this band to baseband.

CO-OFDM presents a better performance than DD-OFDM when signal-to-noise ratios and spectral efficiency are compared. However, DD-OFDM needs less resources because there is no need for a Local Oscillator (LO) at the receiver since the optical carrier must be transmitted together with the OFDM signal.

Both Coherent and Envelope Detection require a LO. Their performance is compared with detail below.

Coherent and Envelope Detection

Figure 13 shows the Coherent Detection scheme. After photodetection, the signal is demodulated with a local oscillator with frequency $f_{IF,Rx}$. In this scheme, the optical carrier can be suppressed and thus different modulation formats can be used (e.g. OCS, ODSB, OSSB). However, generating the local oscillator at frequency $f_{IF,Rx}$ is a difficult task, since such high frequencies are difficult to obtain with low noise levels and need to be optically generated for better results. This figure shows the electrical spectrum[3] after photodetection. The component centered at $f_{IF,Rx}$ is the desired one; however, as Figure 13 shows, other components are present. This is not a problem for this scheme as they do not introduce intermodulation products in the IQ demodulation.

Figure 14 presents the Envelope Detection scheme. The local oscillator needs lower frequencies as the envelope detection block shifts the OFDM spectrum to baseband. Thus, the oscillator's frequency $f_{IF,OFDM}$ is lower; generating this local oscillator becomes easier and cheaper. However, this detection scheme needs an optical carrier present in the spectrum, reducing the possibilities in the modulation formats. Furthermore, it requires electrical filtering before any operation to reduce intermodulation. Figure 14 shows the spectrum of the photodetected signal and the output of the envelope detector. The desired component in the spectrum identified as b is centered at $f_{IF,OFDM}$.

[3] In electrical spectrums, only the positive frequencies are shown. Negative frequencies are symmetric.

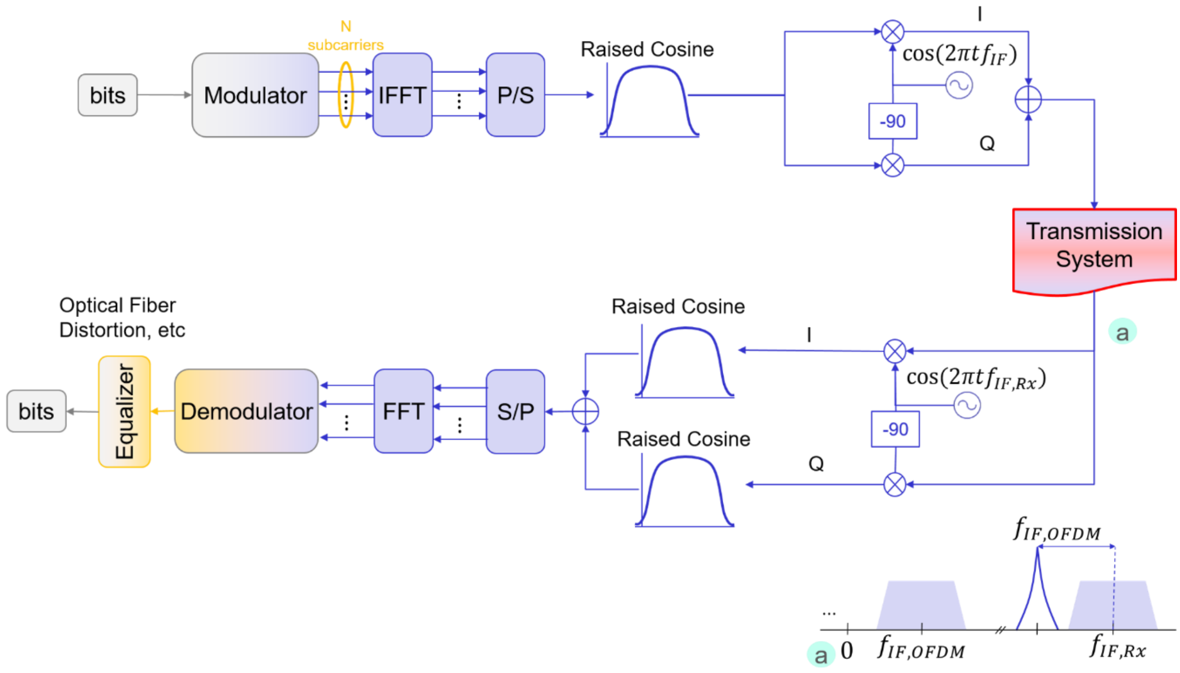


Figure 13 - Block Diagram of the system in the electrical domain, with Coherent Detection

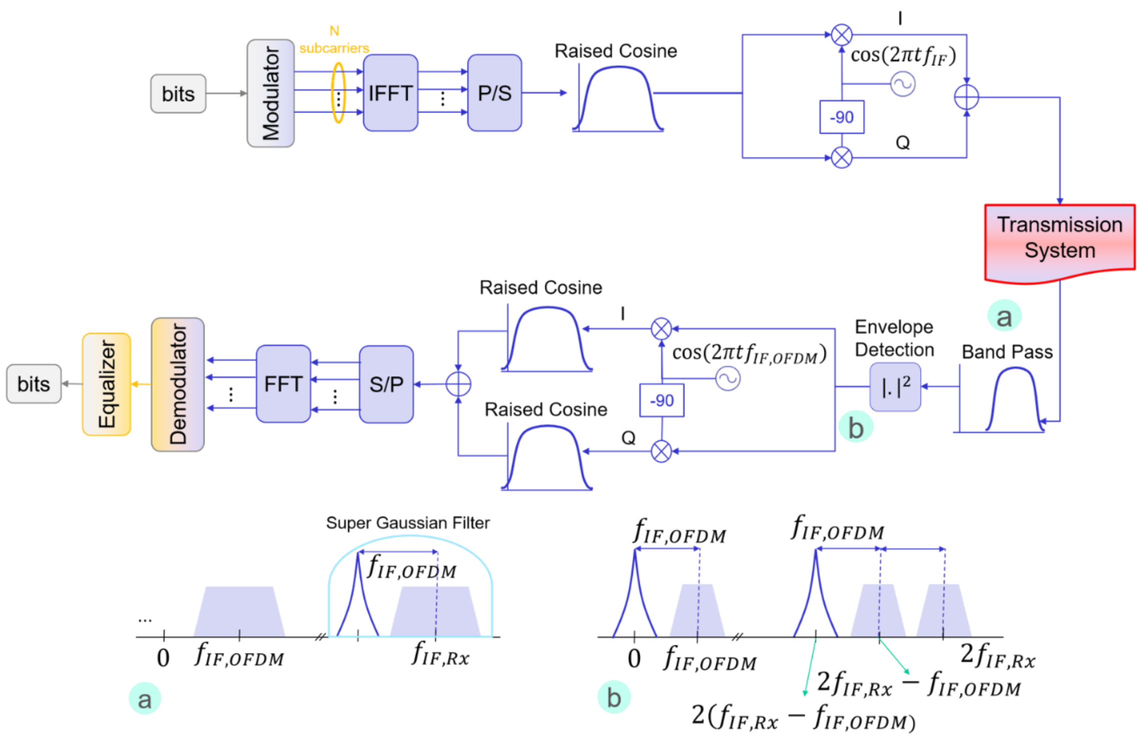


Figure 14 - Block Diagram of the system in the electrical domain, with Envelope Detection

In fact, the Envelope Detection introduces several undesired components.

If we consider that the input of the envelope detector is defined by $x(t)$ in Equation (19), the predicted output is given by Equation (20), where the following trigonometric identities were applied: $\cos^2 a = \frac{1}{2}(1+\cos(2a))$, $\cos(a)\cos(b) = \frac{1}{2}(\cos(a+b)+\cos(a-b))$, $\cos(-a) = \cos(a)$. The absolute value of $x(t)$ was not applied since the frequency components remain the same without this operation; this way, analytical spectrum analysis becomes simple.

$$x(t) = \cos\big(2\pi\big(f_{IF,Rx} - f_{IF,OFDM}\big)t\big) + \cos\big(2\pi f_{IF,Rx} t\big) \tag{19}$$

$$\begin{aligned} x^2(t) = {} & \cos^2\big(2\pi\big(f_{IF,Rx} - f_{IF,OFDM}\big)t\big) \\ & + 2\cos\big(2\pi\big(f_{IF,Rx} - f_{IF,OFDM}\big)t\big)\cos\big(2\pi f_{IF,Rx} t\big) + \cos^2\big(2\pi f_{IF,Rx} t\big) \\ & = \\ = {} & 1 + \frac{1}{2}\cos\big(2\pi\times 2\big(f_{IF,Rx} - f_{IF,OFDM}\big)t\big) + \frac{1}{2}\cos\big(2\pi\times 2 f_{IF,Rx} t\big) \\ & + \cos\big(2\pi f_{IF,OFDM} t\big) + \cos\big(2\pi\big(2 f_{IF,Rx} - f_{IF,OFDM}\big)t \end{aligned} \tag{20}$$

Note that the output of the envelope detector in Figure 14 has components centered at the frequencies identified in Equation (20). The OFDM signal and the carrier were represented by cosine functions at the center frequencies.

This demonstration considers an ideal filter. The implemented filter is not ideal and thus other components arise. Equations (21) and (22) show where these other components are expected.

$$x'(t) = x(t) + \cos(2\pi f_{IF,OFDM} t) \tag{21}$$

$$\begin{aligned} \big(x'(t)\big)^2 = {} & x^2(t) + 2x(t)\cos\big(2\pi f_{IF,OFDM} t\big) + \cos^2\big(2\pi f_{IF,OFDM} t\big) = \\ = {} & x^2(t) + \big[\cos\big(2\pi f_{IF,Rx} t\big) + \cos\big(2\pi\big(f_{IF,Rx} - 2 f_{IF,OFDM}\big)t\big)\big] + \frac{1}{2} \\ & + \frac{1}{2}\cos\big(2\pi\times 2 f_{IF,OFDM} t\big) \end{aligned} \tag{22}$$

In this chapter, analytic analysis was made on the ideal and expected spectrums. In Chapter 5, the simulated spectrums and other results will be presented.

Chapter 4 - Pre-distortion compensation

In this chapter, system modelling and predistortion based on memory polynomials are presented.

In the first section, an analytic analysis on modeling a nonlinear system based on memory polynomials is presented together with the predistortion principles.

In section 4.2, a straightforward discussion on predistortion in simulated systems is presented.

Finally, section 4.3 shows the predistortion in the system and its implementation.

4.1. Overview on the predistorter

The implemented predistorter is based on memory polynomials [24]–[27]. Memory polynomials allow the expression of the output of a given nonlinear system as a function of the input. Equation (23) shows that we can model the system by calculating the coefficients a_{kq}, where K and Q represent the nonlinear order and memory length, respectively.

$$y(n) = \sum_{k=1}^{K} \sum_{q=1}^{Q} c_{kq} x(n-q) |x(n-q)|^{k-1} \tag{23}$$

Equation (23) can be rewritten in a matrix form. Equation (24) defines a new variable, X, as a matrix where a delay of q samples is introduced to the n^{th} sample of x, where N represents the total number of samples. Equation (25) expresses the estimated output, $\hat{y}$, as a function of the matrix X.

$$\begin{gathered} x_{kq} = x(n-q)\ |x(n-q)|^{k-1} \\ X_{kq} = \left[x_{kq}(n)\ , x_{kq}(n+1) \dots\ x_{kq}(n+N-1)\ \right]^T \\ X = \left[X_{10} \dots X_{K0}, X_{11} \dots X_{K1}, \dots X_{1Q} \dots X_{KQ}\right] \end{gathered} \tag{24}$$

$$\hat{y} = X \times c_{kq} \tag{25}$$

Modeling the system can be achieved by calculating the coefficients c_{kq}. Equation (26) shows the Minimum Square Method to determine such coefficients, where $(X^H X)^{-1} X^H$ represents the pseudo-inverse of X and $.^H$ represents the Hermitian transpose.

$$\begin{gathered} c_{kq} = (X^H X)^{-1} X^H y \\ c_{kq} = \left[c_{10} \dots c_{K0}, c_{11} \dots c_{1Q}\ c_{KQ}\right] \end{gathered} \tag{26}$$

4.2. Simulating systems with predistortion

When simulating a system, the model of that system does not need to be modelled. Instead, the inverse of the system should be estimated in order to calculate de predistortion signal. Literature on modeling the system is abundant [24]–[26], [28]–[33], although often unclear how the predistortion signal is computed. For this reason, this section will be highly straightforward.

Figure 15 shows two block diagrams for the predistorter. Figure 15-a represents the first step needed when implementing a predistorter. The input and output of the system must be collected, requiring feedback of the received signal, y, considered to be ideal. Figure 15-b represents the final step, when the inverse of the system is already modelled and the total system is now close to linear.

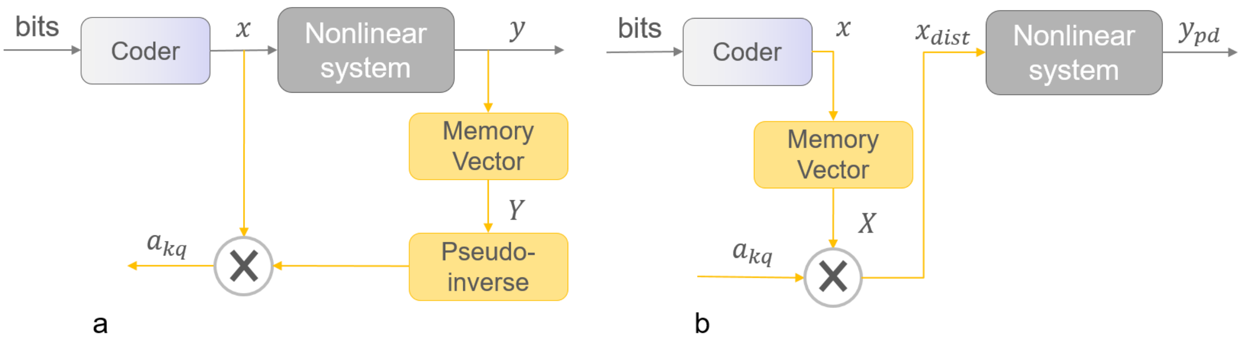


Figure 15 - Block Diagram of the Predistorter. a) Computing the coefficients from the input and output of the nonlinear system. b) Computing the predistortion signal from the loaded coefficients and coded signal

In Figure 15-b, the block "Memory Vector" applies Equation (24) to x and in Figure 15-a, Y is computed in a similar way, using the output of the system, y.

The coefficients for the predistorter, a_{kq}, are different from the coefficients that model the system.

There are several proposed algorithms in [24], [28] that increase the accuracy in calculating these coefficients; however, they are highly dependent on the parameters and thus they must be optimized. These algorithms were implemented and tested but they were not used in this dissertation due to the unpredictability in the results. Furthermore, these parameters needed optimization for each amplitude, being highly time consuming in simulations. The minimum square method[4] was used, producing steady and good results.

The a_{kq} coefficients are thus obtained using the minimum square method given by Equation (27), where $.^H$ represents the Hermitian transpose.

The block "Pseudo-inverse" calculates $(Y^HY)^{-1}Y^H$, present in Equation (27). The output of the predistorter is given by Equation (28), which is then applied to the non-linear system, producing a linear output.

$$a_{kq} = (Y^HY)^{-1}Y^Hx$$
$$a_{kq} = [a_{10} \dots a_{K0}, a_{11} \dots a_{1Q}\ a_{KQ}] \quad (27)$$
$$x_{dist} = X \times a_{kq} \quad (28)$$

Throughout this discussion, k is assumed to take values from 1 to K. However, in [24], [26], [28], only odd orders are used for k, as even values do not improve the predistortion significantly but increase the complexity. In this dissertation, k can take even and odd or only odd values.

4.3. Predistortion Subsystem

Figure 16 shows the system with focus on the predistorter.

First, the signals represented as x and y are stored. The block diagram in Figure 16 also shows other important blocks of the system so that the content of these signals is given. It is particularly important to note that x is stored before the raised cosine filtering and RF conversion. The signal y is thus analyzed after baseband conversion and raised cosine filtering, to "reset" the changes applied by RF conversion and raised cosine filtering.

The coefficients a_{kq} are calculated from the output and signal x, as it is said in Equation (27). Since the system is simulated using Matlab, only the inverse transfer function of the system is modelled.

In a second run, the coefficients are loaded and the predistortion signal x_{dist} is calculated from x and a_{kq}. The input of the transmitter's raised cosine block is now x_{dist} instead of x, making the output of the receiver's raised cosine block linear with the input.

[4] This method is presented with detail in the appendix.

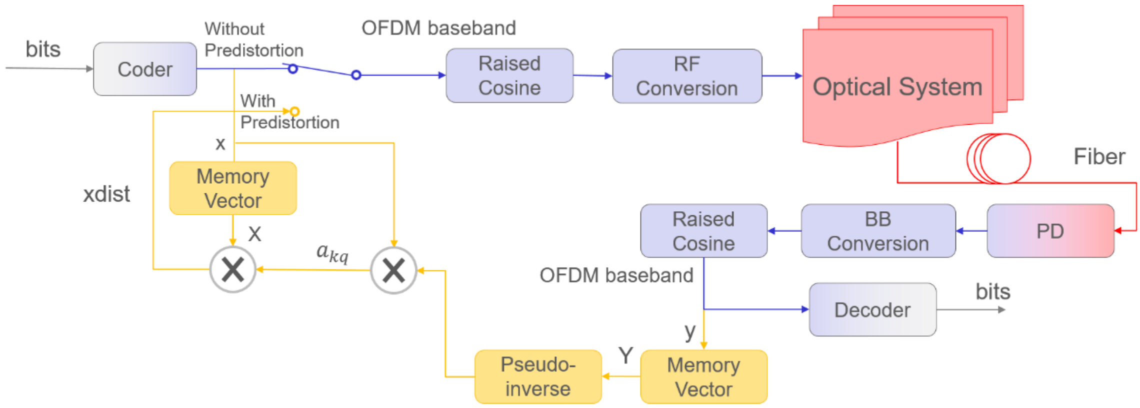


Figure 16 - The predistorter system

The transfer function of the system is considered to be constant and thus more runs of the system can be made without recalculating the coefficients. In mobile networks, however, this is not true: the transfer function can change over time, introducing a need for recalculating the coefficients. In this work, this will not be considered as recalculating these coefficients is a simple task.

Chapter 5 - Performance of the Optical System

In this chapter, the system is executed with different parameters and the performance is compared. The performance is analyzed through Error Vector Magnitude (EVM), Bit Error Rate (BER) and the transmitted optical and electrical spectrums.

First, the system's general parameters are presented. Simulation results with these parameters are presented, such as the received constellations with and without predistortion and AM-AM and AM-PM plots for the predistorter and the system and the input and output spectrums for the implemented band-pass filters.

In the third section, the performance of the modulation formats is analyzed to obtain the optimum modulating point (DC bias) of the Mach-Zehnder modulator. Several configurations are used, as the transmission method and extinction ratio's values vary.

With the optimal modulator, the impact of the number of bits per symbol and transmission rate is analyzed, in Sections 4 and 5, respectively.

After the optimal transmission system is obtained, the impact of the fiber's length is analyzed in Section 6.

Finally, in Section 7, the impact of the linewidth of the lasers on the performance is analyzed for all the systems.

Along each analysis, the performance of the system with predistortion is compared to that of the system without predistortion.

5.1. Simulation Results and Parameters

In this section, the parameters used for simulating the results are presented together with some general results.

System's Parameters

Table 2 presents the parameters. In the following sections, some of these parameters will be changed and will be specified then; unspecified parameters remain with the same value.

	Parameter and description		Value	Units
Physical Parameters	c_o Light speed in vacuum		2.99792458×10^{8}	m/s
Optical Parameters	v_o Optical center frequency, reference optical frequency		193.1×10^{12}	Hz
	λ_o Reference Wavelength		$\frac{c_o}{v_o}$	m
	v_1 Optical frequency (with reference to v_o) in the single tone generated by CW Laser 1		27.5×10^{9}	Hz
	v_2 Optical frequency (with reference to v_o) in the single tone generated by CW Laser 2		-27.5×10^{9}	Hz
	Laser's Linewidth		1	Hz
	V_π Voltage needed to cause a phase shift of π in the Mach-Zehnder Modulator		3.5	V
OFDM Parameters	m Bits per Symbol		4	-
	M Number of Symbols		2^{m}	-
	R Bit rate		5×10^{9}	bit/s
	Modulation Format		M-QAM	-
	Number of Carriers		32	-
	Cyclic Prefix Ratio of Cyclic Prefix's length with reference to the symbol's length		0.125	-
	Roll-off factor		0.25	-
	$f_{IF,OFDM}$ Center frequency for the OFDM signal		5×10^{9}	Hz
	$f_{IF,Rx}$ Frequency of the OFDM signal		60×10^{9}	Hz
Predistorter Parameters	K Order of nonlinearity		4	-
	Q Memory length		1	-
Fiber Parameters	α Attenuation in the fiber		0.22	dB/Km
	D Dispersion coefficient		16×10^{-6}	s/m^{2}
Optical Filter Parameters	F_c Center Frequency	OSSB	30×10^{9}	Hz
		ODSB	27.5×10^{9}	
		OCS		
	Bw Bandwidth	OSSB	10×10^{9}	Hz
		ODSB	15×10^{9}	
		OCS		
	Gain	OSSB	5	

		ODSB		Linear units
		OCS		
	SbA Stop band Attenuation	OSSB	50	dB
		ODSB		
		OCS		
	Super Gaussian Order	OSSB	8	-
		ODSB		
		OCS		
Electrical Filter Parameters Only for OSSB for Envelope Detection	F_c Center Frequency		59×10^9	Hz
	Bw Bandwidth		10×10^9	Hz
	Gain		5	Linear units
	SbA Stop band Attenuation		50	dB
	Super Gaussian Order		8	-
Other Parameters	f_s Sample Rate		$64\times R$	Hz

Table 2 - System's Parameters

Note that the input power of the OFDM signal will be changed frequently to study the nonlinearity of the Mach-Zehnder Modulator. In fact, the amplitude of the input signal is normalized but multiplied by a variable factor when entering the MZM. Throughout this work, the OFDM signal's power (referred to as RF power) is presented frequently together with the amplitude factor.

General Results

The following simulation results were acquired with Optical Carrier Suppressed in a Coherent Detection scheme for an amplitude factor of 21.7 ($\sim25\ dBm$).

Figure 17 shows the AM/AM and AM/PM for the predistorter. In an ideal predistorter, the predistortion signal, x_{dist}, presents a symmetry relative to $y = x$ of the non-distorted signal, y. From Figure 17-a, we see that x_{dist} is not far from this. Thus, as expected, the predistorted output, y_{pd}, presents a close-to-linear relation with the input signal, x. The AM/PM plot is also presented in Figure 17-b. This figure shows that for some samples, the difference between the angles of x and y is not null. With predistortion, however, the difference of the angles of x and y_{pd} is close to zero. Thus, phase distortion can also be compensated with the implemented predistorter.

The predistortion is expected to change the spectrum when there is intermodulation. In fact, Figure 18 shows the Optical spectrums without and with predistortion. From this figure, we can see that the side bands of the system with predistortion are narrower and $\sim 2dB$ below.

Because of the linearity of the output with the input, the performance of the system is expected to increase. In fact, Figure 19 shows how the predistortion improves the EVM by displaying the received constellations without and with predistortion. Without predistortion, the EVM was 7.6% (or $-11.1\ dB$); with predistortion, the EVM decreases to 2% (or $-16.8\ dB$).

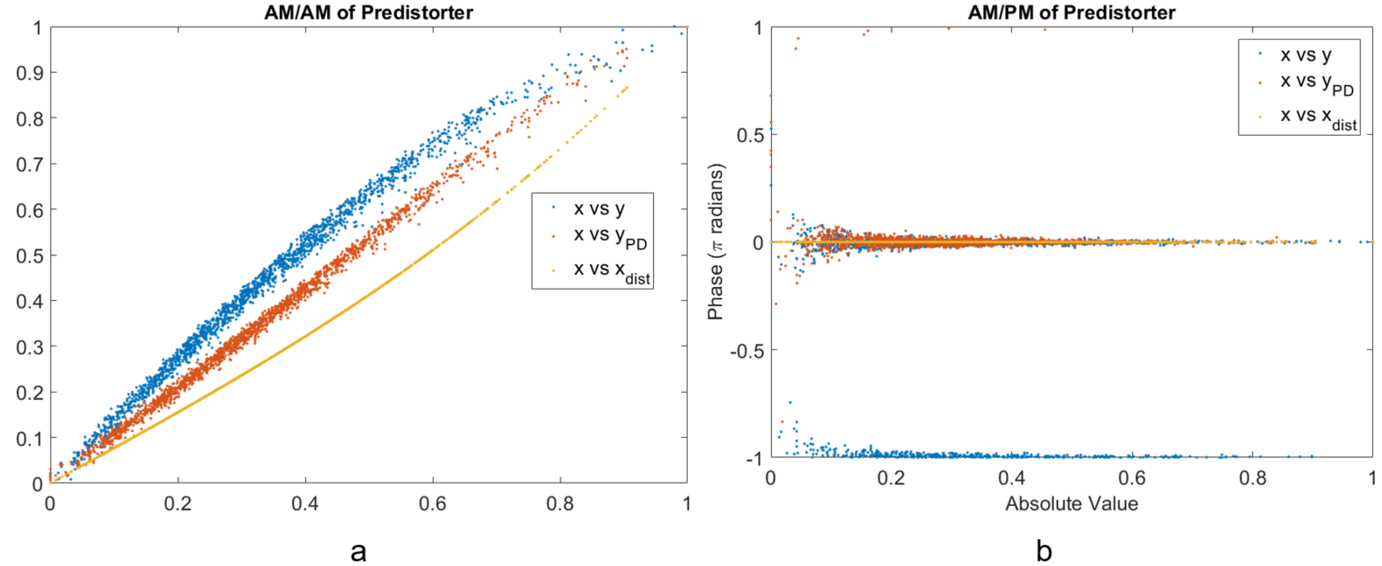


Figure 17 - AM/AM and AM/PM plots of the predistorter with OCS. a) AM/AM. b) AM/PM

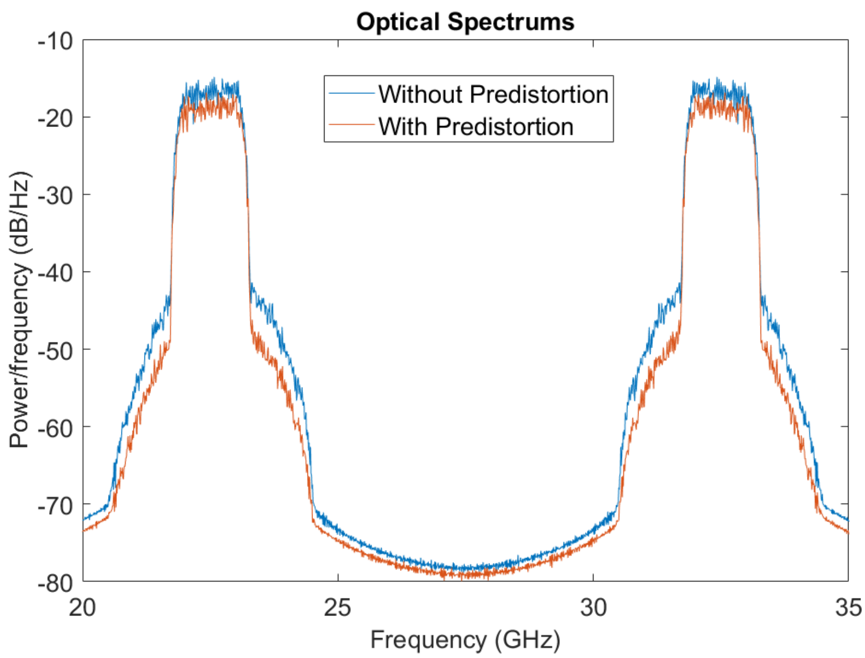


Figure 18 - Optical spectrum at the output of MZM without and with predistortion.

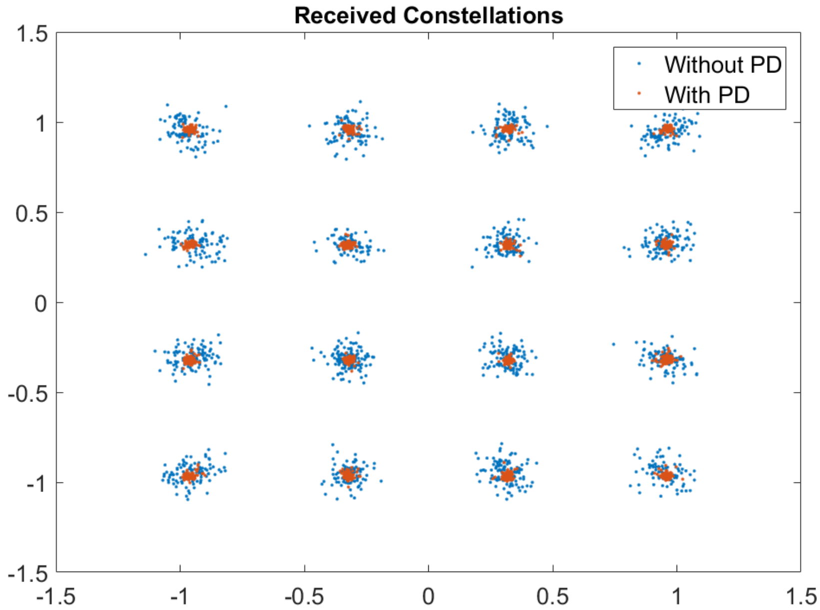


Figure 19 - Received Constellations without and with predistortion

With OSSB, as Table 1 shows, the Hilbert Transform of the electrical signal is applied to the lower arm. For this reason, a phase shift of π is applied in the modulation; thus, the AM/PM plot for OSSB should be different. Figure 20 shows the AM/AM and AM/PM plots for OSSB. Comparing Figure 17 and Figure 20, a significant difference is present in the AM/PM plot. As we can see in Figure 20-b, there is a phase difference in the y samples; however, the predistorter can also undo this phase shift and the predistorted output y_{pd} no longer presents a phase difference to the input x.

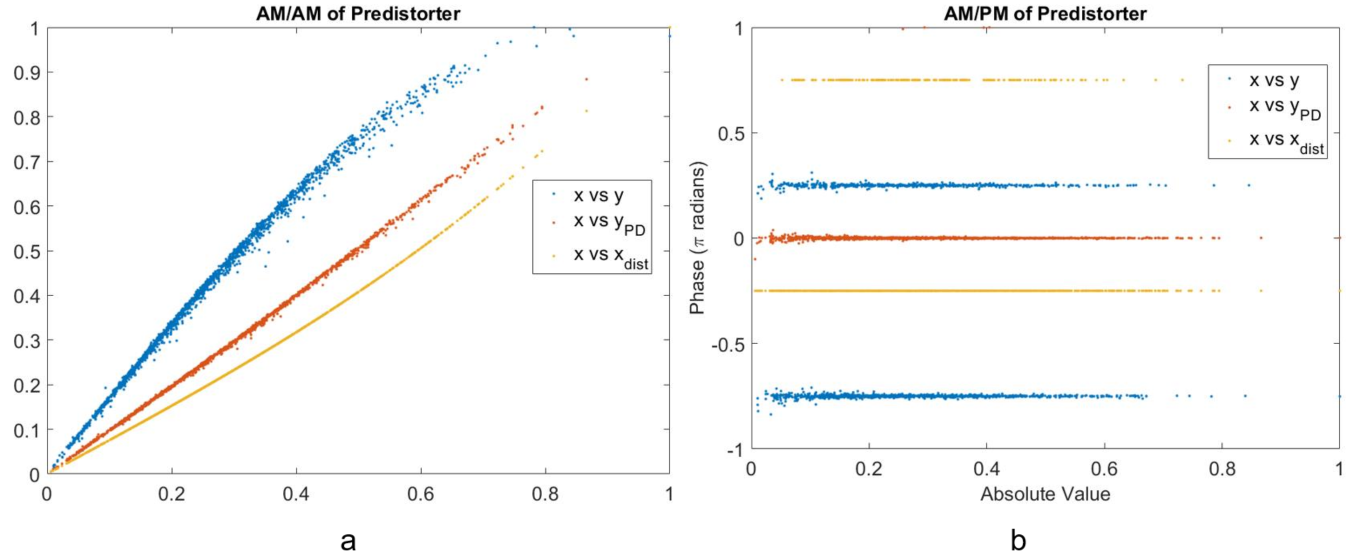


Figure 20 - AM/AM and AM/PM plots of the predistorter with OSSB. a) AM/AM. b) AM/PM

In both detection schemes, coherent and envelope detection, an optical filter is applied to remove possible Bessel products. The filter's parameters are presented in Table 2. Figure 21 shows the input and output of this filter for OCS for the same parameters. Since the filter was designed to be non-ideal, some undesired components are present at the output, although highly attenuated.

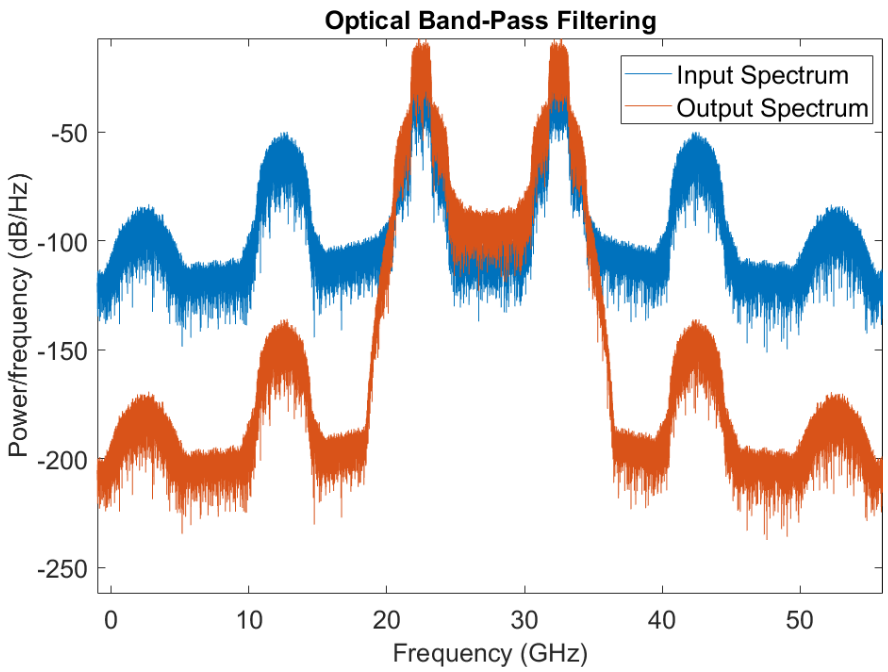


Figure 21 - Input and Output Optical Spectrums of the Optical Band-Pass Filter

In the Envelope Detection scheme, an electrical filter is applied immediately after photodetection. Table 2 shows the parameters of this filter and Figure 22 shows its input and output spectrums. The figure shows that the undesired components are attenuated.

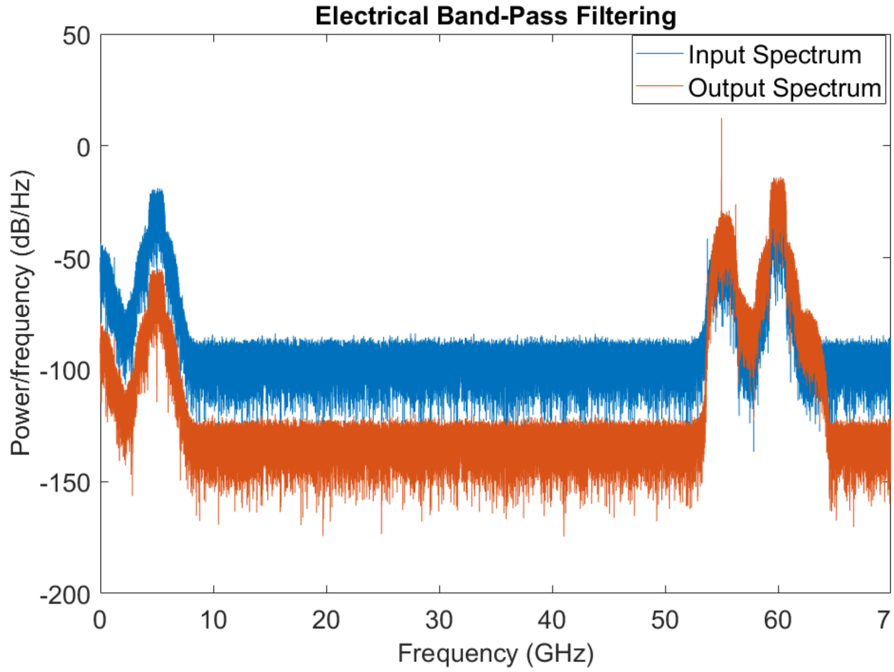


Figure 22 - Input and Output Electrical Spectrums of the Electrical Band-Pass Filter

In the following sections, the performance of the systems without and with fiber transmission will be compared.

Figure 23 presents the electrical spectrums for the systems without and with fiber and without and with predistortion, for systems with Coherent Detection. The predistorter can improve the performance of both systems, although it is more evident in Figure 23-a. Comparing the systems in Figure 23-b with those of Figure 23-a, we see that the systems with fiber transmission have higher attenuation ($\sim 4\ dB$) and wider sidebands ($\sim 50 MHz$). Table 3 shows the EVM for these systems. Note that for the system without predistortion and with fiber, the EVM is higher than 100%; this usually means that there is a phase shift in the in-phase and quadrature components. In the systems with predistortion, the system without fiber was expected to have better performance; however, this is not the case. In systems with fiber transmission, the coefficients are estimated according to that fiber's characteristics. This can be the reason behind the improvement in performance. Even though, the difference is not significant and can be caused by the modelling of the inverse of the system when computing the predistorter since it is highly dependent on the order of nonlinearity and memory length and these parameters are not optimized for each run.

Figure 24 shows the electrical spectrums for the systems without and with fiber and without and with predistortion, for systems with Envelope Detection. The EVM is presented in Table 3. We see that the performance of the system with predistortion improves: the laterals of the sideband are $\sim 8dB$ below when there is no fiber and $\sim 4dB$ when there is a fiber with 10 Km of length. The

EVM for the system without fiber and with predistortion has better performance because of this higher decrease.

Comparing Figure 23 and Figure 24 we see that the predistorter in the system with ED does not change the bandwidth, although it decreases the power of the intermodulations. From Table 3 we conclude that the system with Coherent Detection (with OCS) performs better than the system with Envelope Detection (with OSSB).

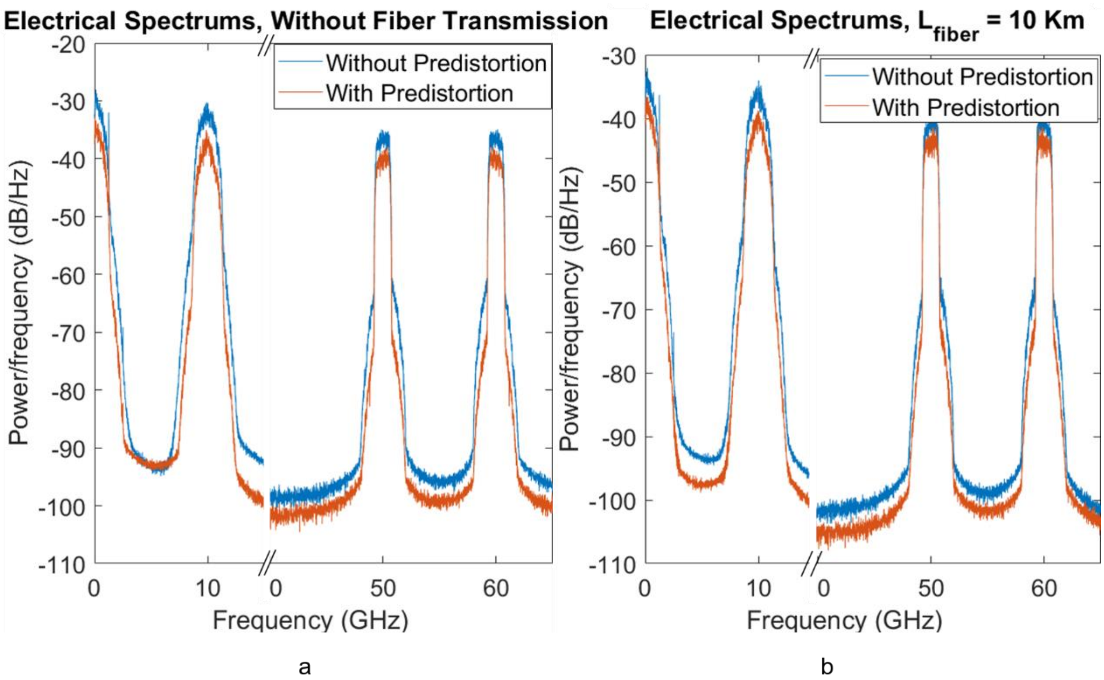


Figure 23 - Electrical Spectrums without and with predistortion, with Coherent Detection. a) Without fiber transmission. b) With fiber transmission

	Coherent Detection		Envelope Detection	
	Without Predistortion	With Predistortion	Without Predistortion	With Predistortion
$L = 0\ Km$	8.7%	2.5%	12.7%	4.6%
$L = 10\ Km$	113%	2.1%	12.5%	5.9%

Table 3 - EVM for the systems without and with predistortion, without and with fiber transmission, with Coherent Detection and Envelope Detection

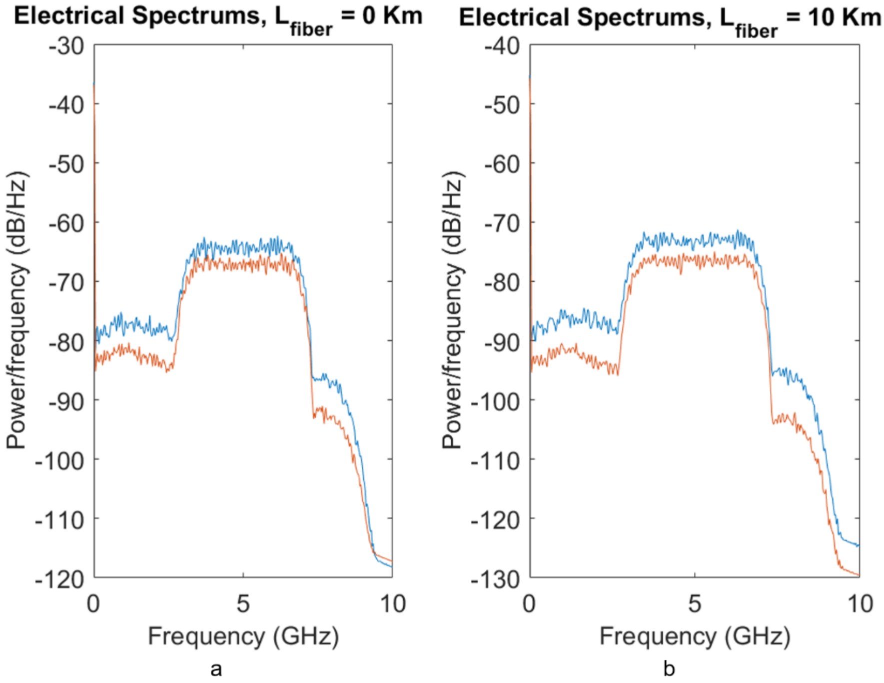


Figure 24 - Electrical Spectrums without and with predistortion, with Envelope Detection. a) Without fiber transmission. b) With fiber transmission

5.2.The impact of the bias point of the Mach-Zehnder

In this section, the DC bias point of the Mach-Zehnder is modified to obtain different modulation formats. This is achieved by sending different electrical signals to each arm of the Mach-Zehnder. Table 1 relates the electrical signals with the modulation formats and Figure 9 also shows the bias points for these modulation formats.

Figure 25 shows the optical spectrums at the output of the Mach-Zehnder Modulator with reference to the optical center frequency v_o. These spectrums were achieved with Coherent Detection by using unity amplitude factor, a laser linewidth of 10 kHz and $R = 15\ GHz$; the Mach-Zehnder was modeled to be ideal, with a splitting ratio of 0.5 ($ER \sim \infty$).

Optical Single Side Band (OSSB) presents smaller bandwidth use. Optical Dual Side Band (ODSB) and Optical Carrier Suppressed (OCS) present identic bandwidths; the main difference between these spectrums is the presence of the carrier between the sidebands.

The performance of each modulation format can be analyzed with Figure 26. In this figure, the Forward Error Correction (FEC) limit[5] line is represented in a green dotted line. FEC performs for values of EVM inferior to this line.

[5] The EVM limit value for FEC is derived in the appendix.

From Figure 26 we conclude that OCS has the best performance in average. In fact, when we analyze Figure 25 and Figure 26, we see that the power of the data in OSSB is the lowest; although it is not clear, OCS has the biggest power in each band. The output intensity of the Mach-Zehnder is limited and we see in Figure 25-a that the carrier "steals" power from the data. In Figure 25-b this is also the case, although it is not as significant. Thus, since in Figure 25-c the carrier is suppressed, the power in the sidebands of OCS and its performance is superior.

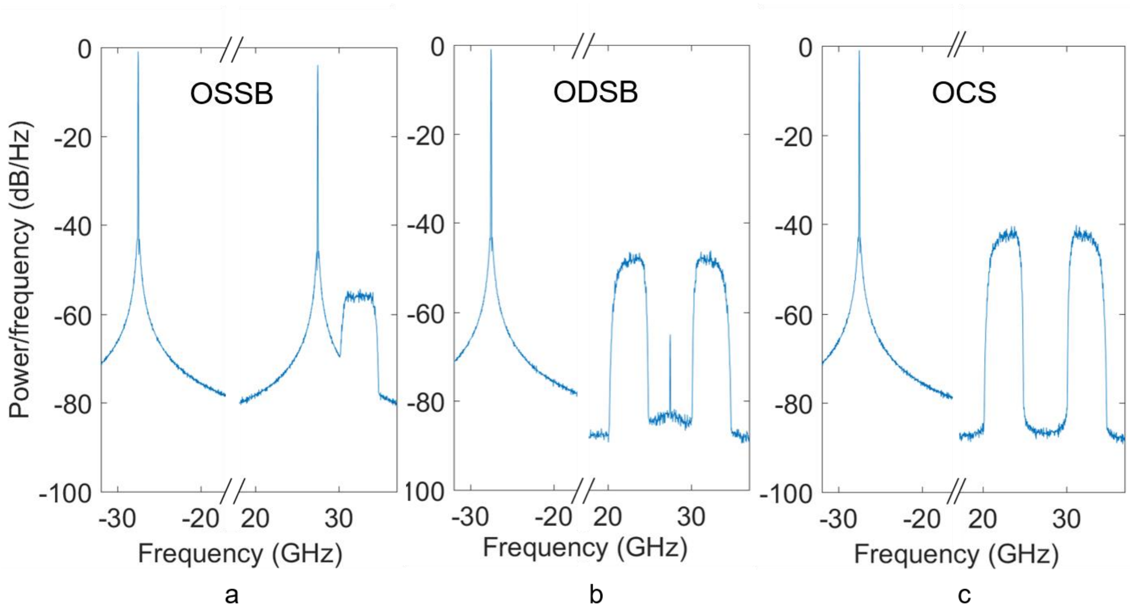


Figure 25 - Optical spectrums at the output of the coupler for an ideal MZM. a) Spectrum for OSSB. b) Spectrum for ODSB. c) Spectrum for OCS.

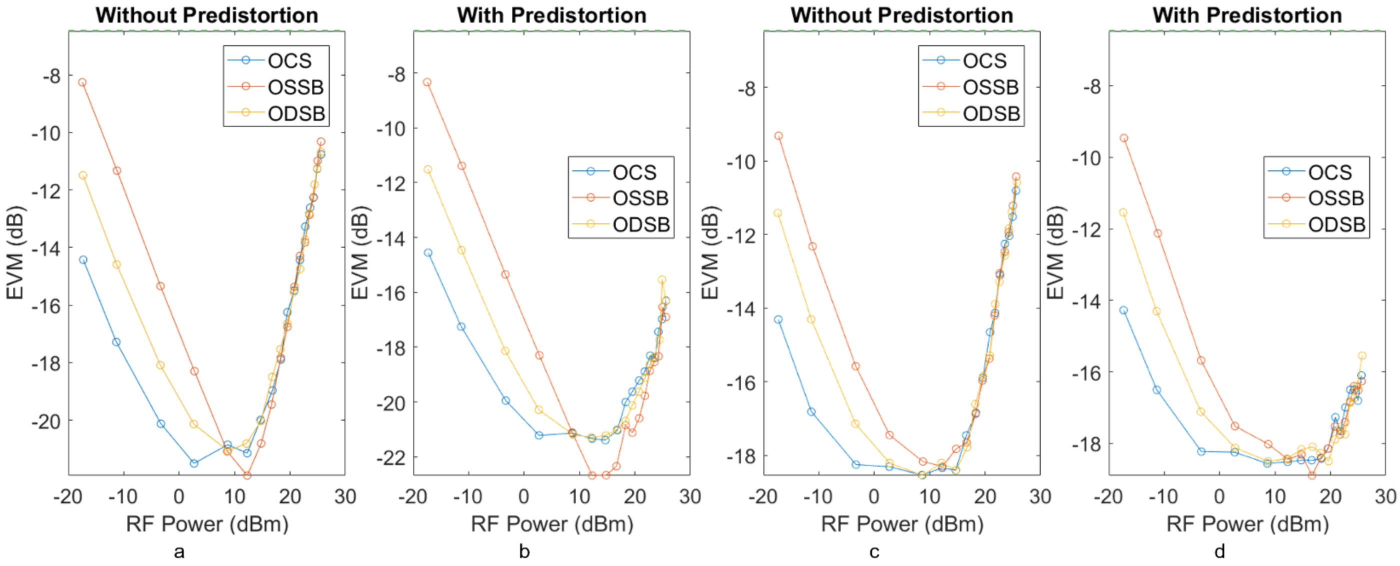


Figure 26 - Impact on EVM with the increase of input power for different modulation formats for an ideal MZM. a) Without predistortion, without fiber. b) With predistortion, without fiber. c) Without predistortion, with fiber. d) With predistortion, with fiber.

Figure 26-a and Figure 26-b show the performance for the systems without and with predistortion, without using fiber. For values of power inferior to approximately 15 dBm, the predistorter cannot reduce the EVM; in these values, the system is degraded by noise. For values above, the distortions dominate the system. In these values, the predistorter can reduce the maximum EVM from $-10\ dB$ to $-15\ dB$.

Figure 26-c and Figure 26-d show the performance of the systems without and with predistortion, using a fiber with 10 Km of length. The performance is expected to decrease when comparing with the systems without fiber; however, it increases slightly. In the previous section, this was also observed and justified.

Real world Mach-Zehnder modulators do not have a splitting ratio of 0.5. Figure 3 shows that for an unideal splitting ratio the upper arm applies a factor of r while the lower arm applies $1 - r$. Using $r = 0.47$ $(ER \sim 30\ dB)$, we obtain Figure 27 and Figure 28.

The main differences between an ideal and unideal Mach-Zehnder Modulator are observed when comparing Figure 25 and Figure 27, namely in the optical carrier at the optical frequency v_1. In ODSB, the power of this carrier is now superior to the power of the two side bands. Figure 27-c is identified as OCS as the electrical signals applied to the upper and lower arms of the Mach-Zehnder Modulator maintain the relations identified in Table 1; however, the carrier is not suppressed.

The values of EVM do not change significantly and OCS remains the modulation format with the best performance. The systems with fiber transmission continue to have better performance.

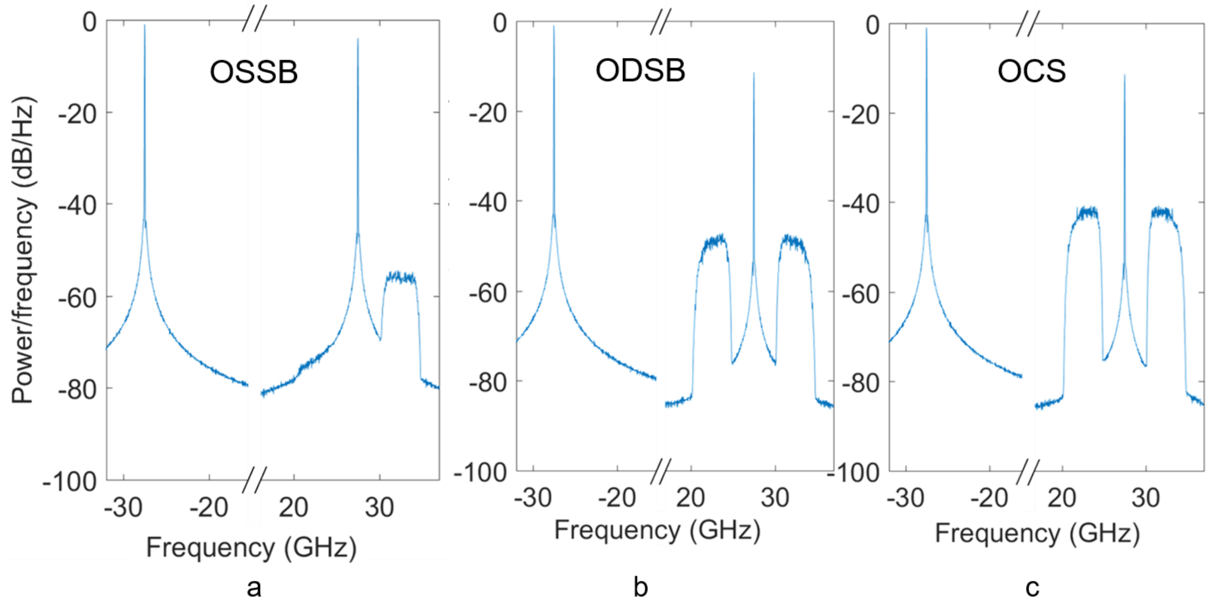


Figure 27 - Optical spectrums at the output of the coupler for an unideal MZM. a) Spectrum for OSSB. b) Spectrum for ODSB. c) Spectrum for OCS

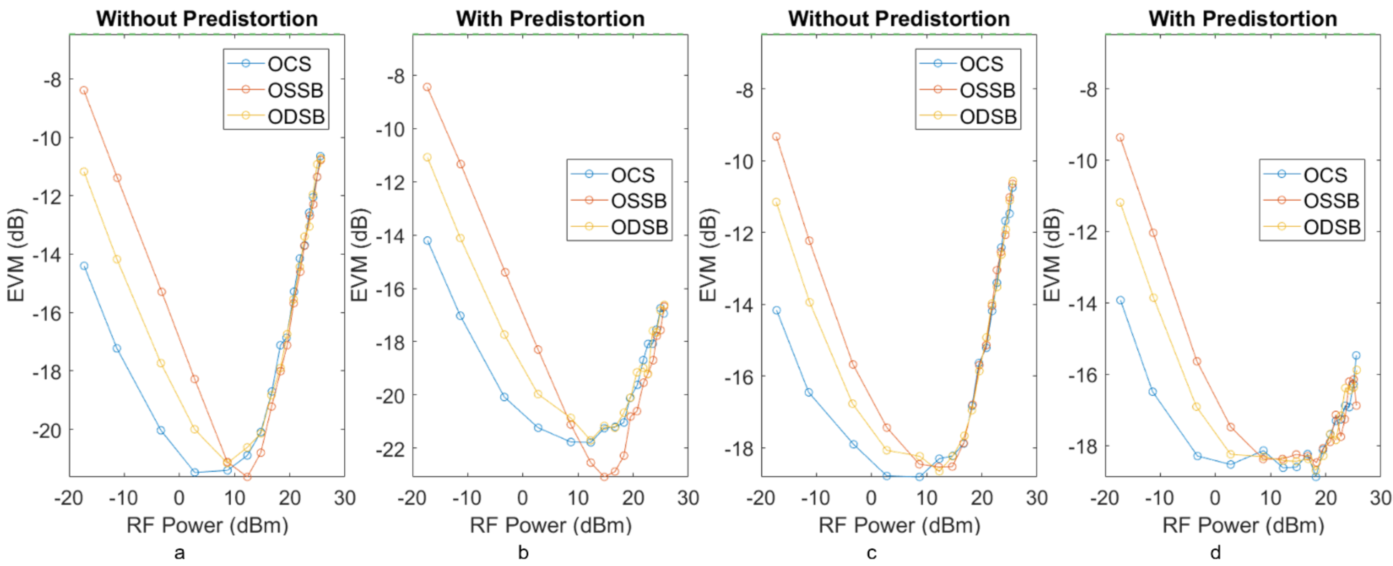


Figure 28 - Impact on EVM with the increase of input power for different modulation formats for an unideal MZM. a) Without predistortion, without fiber. b) With predistortion, without fiber. c) Without predistortion, with fiber. d) With predistortion, with fiber.

In average, the performance of the system is optimized for OCS. In the following sections, OCS is used with Coherent Detection and OSSB is used with Envelope Detection.

5.3. The impact of the Number of Bits per Symbol

The number of bits per symbol defines the number of symbols $M = 2^m$ in $M - QAM$. Spectral efficiency increases with M because the bandwidth decreases by a factor of $m = \log_2 M$.

In this section, three different values of M are compared: $M = 16, M = 64$ and $M = 256$ (or, equivalently, $m = 4, m = 6$ and $m = 8$). Different noise spectral density values are also studied.

Figure 29, Figure 30 and Figure 31 show the change in EVM when M increases. The mean EVM is minimum when $M = 256\ (m = 8)$. This derives from the spectrum use: with less bandwidth, less white noise affects the data; thus, better performance is achieved.

From these figures, we see that as the noise spectral density increases, the performance decreases, which is expected. It is interesting to analyze these figures without and with predistortion: the increase of noise degrades both systems equally; the predistorter cannot compensate the noise impact. In the previous section, we concluded that the predistorter can only increase the performance of the system when the power is superior to 15 dBm; this conclusion remains valid in this section.

With these figures, an analysis on BER is also possible. It is especially interesting to analyze the FEC limit: systems with values of BER above the green dotted line are not functional due to errors. We see that for higher values of η some values of RF power yield values of BER superior to the FEC limit.

With an increase of M, there is an increase on the power of the data at the output of the Mach-Zehnder. For this reason, in all aforementioned figures, the curve for $M = 256$ exceeds the FEC limit more rapidly (or is the only one exceeding). Figure 32 shows this phenomenon and the bandwidth narrowing with the increase of M.

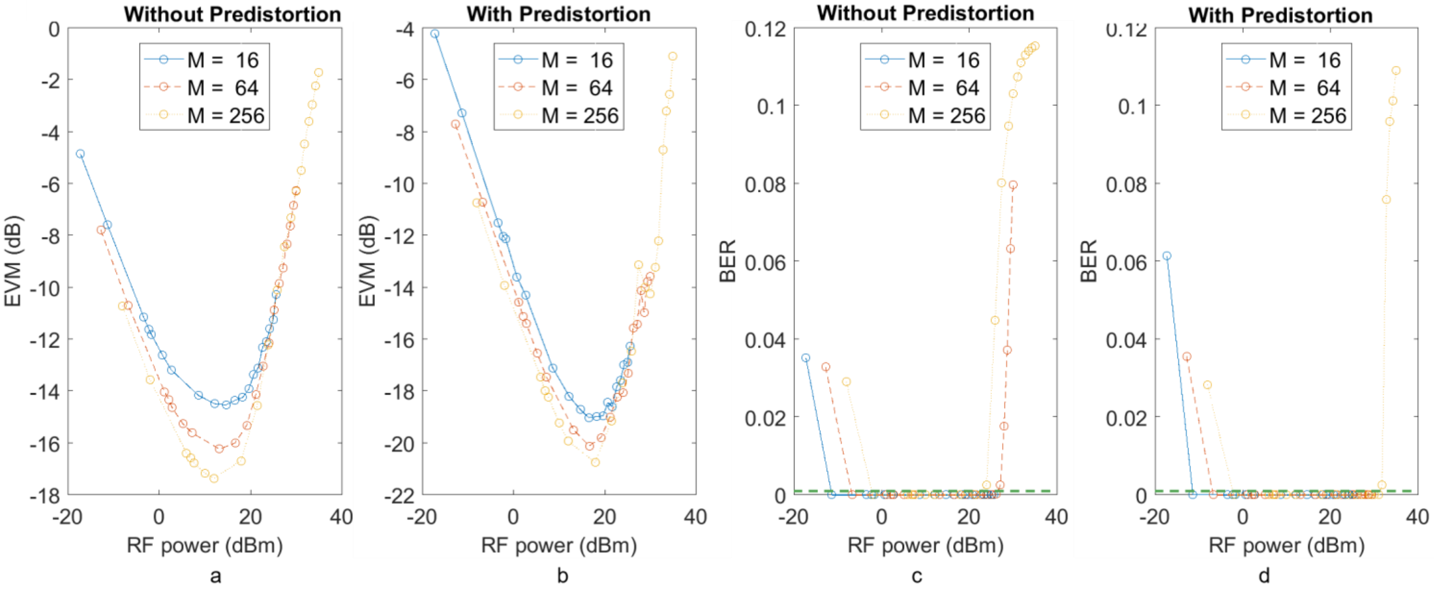


Figure 29 - Impact of M in the performance of the system for $\eta = -70\ dBm/Hz$, without fiber transmission, with Coherent Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER, with predistortion.

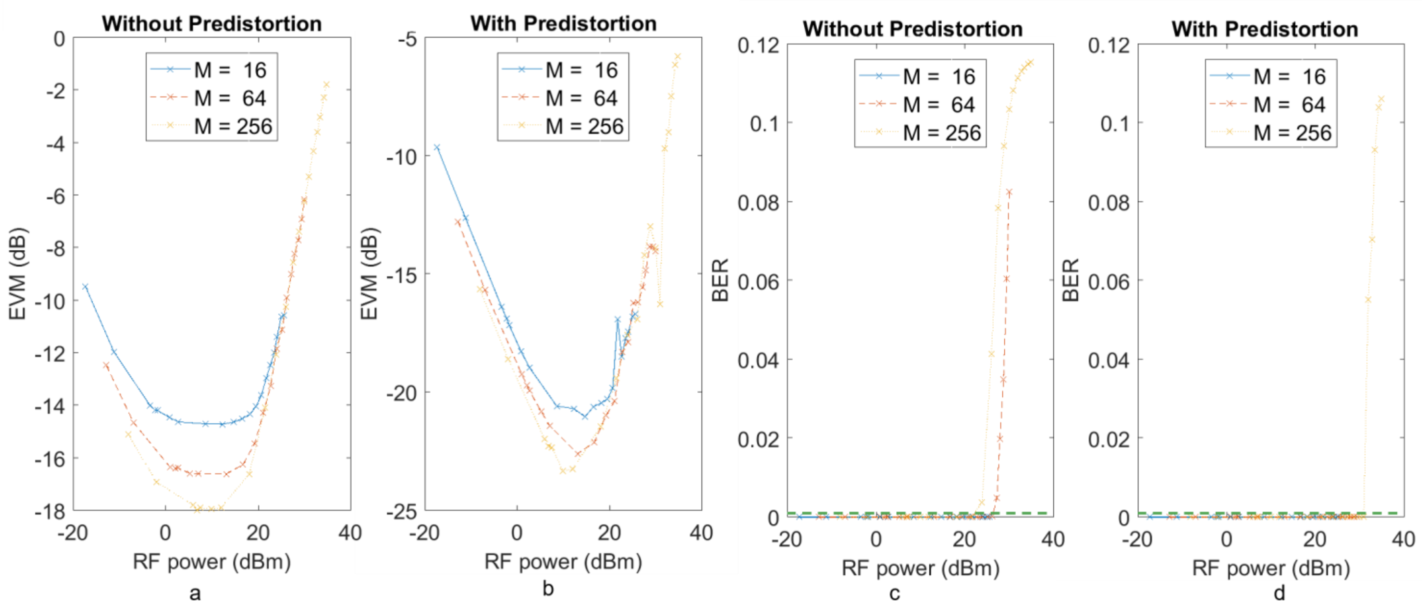


Figure 30 - Impact of M in the performance of the system for $\eta = -80\ dBm/Hz$, without fiber transmission, with Coherent Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER, with predistortion.

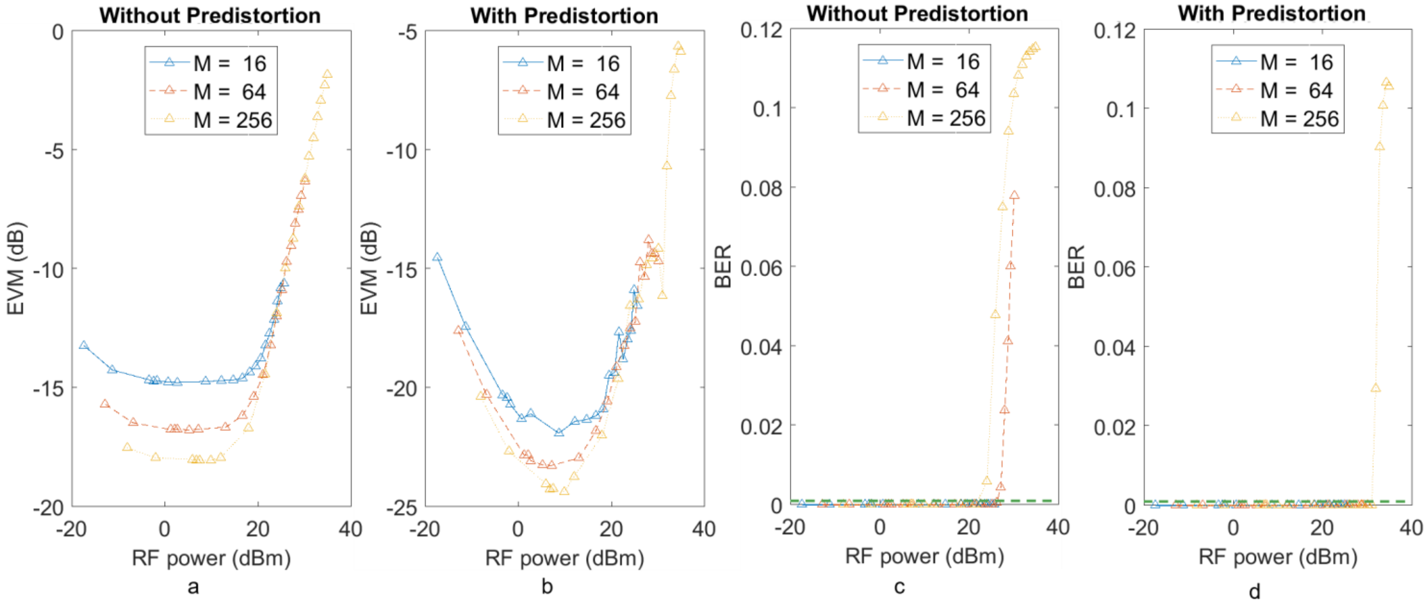


Figure 31 - Impact of M in the performance of the system for $\eta = -90\ dBm/Hz$, without fiber transmission, with Coherent Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER, with predistortion.

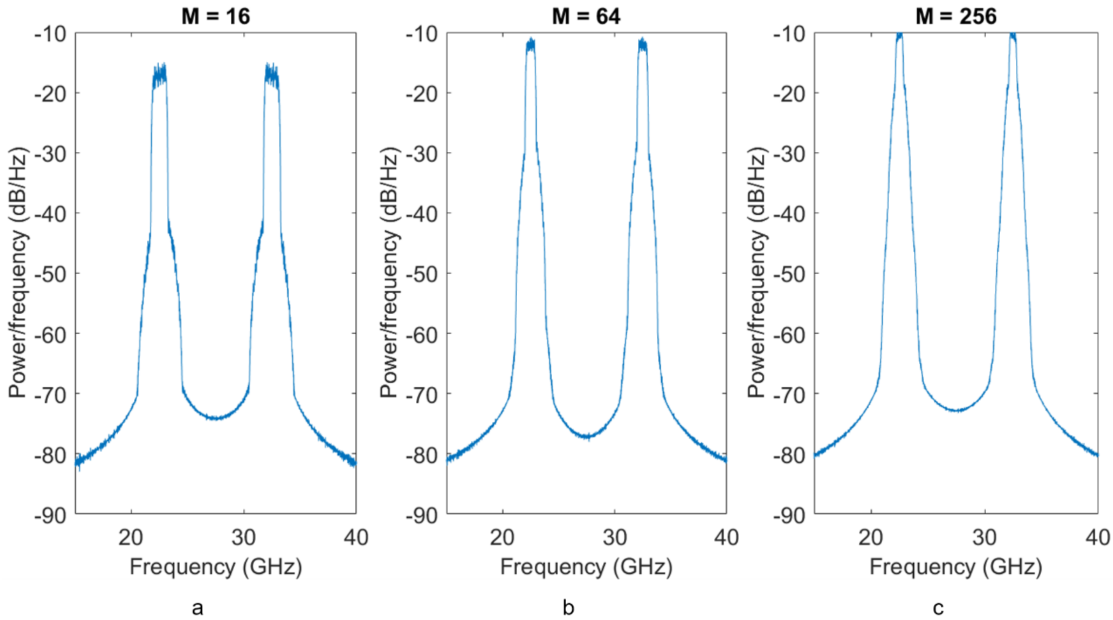


Figure 32 - Optical spectrum at the output of the MZM for different values of M. a) M = 16. b) M = 64. c) M = 256

The performance of the system with Envelope Detection was also analyzed. Figure 33, Figure 34 and Figure 35 show the EVM for systems with different values of η. The FEC limit in BER is represented as a green dotted line. The systems with Envelope Detection behave similarly to those with Coherent Detection: the optimum value of M is 256, only a few values of RF power are viable and the performance increases when the noise spectral density decreases.

Comparing the performance of the systems with Coherent Detection (Figure 29, Figure 30 and Figure 31) with the systems with Envelope Detection (Figure 33, Figure 34 and Figure 35), we see that Coherent Detection leads to better performances. In Envelope Detection, the absolute value of the signal is squared, increasing the power of both the signal and noise; furthermore, OSSB is used, which has been proved in the previous section to be inefficient when compared to OCS.

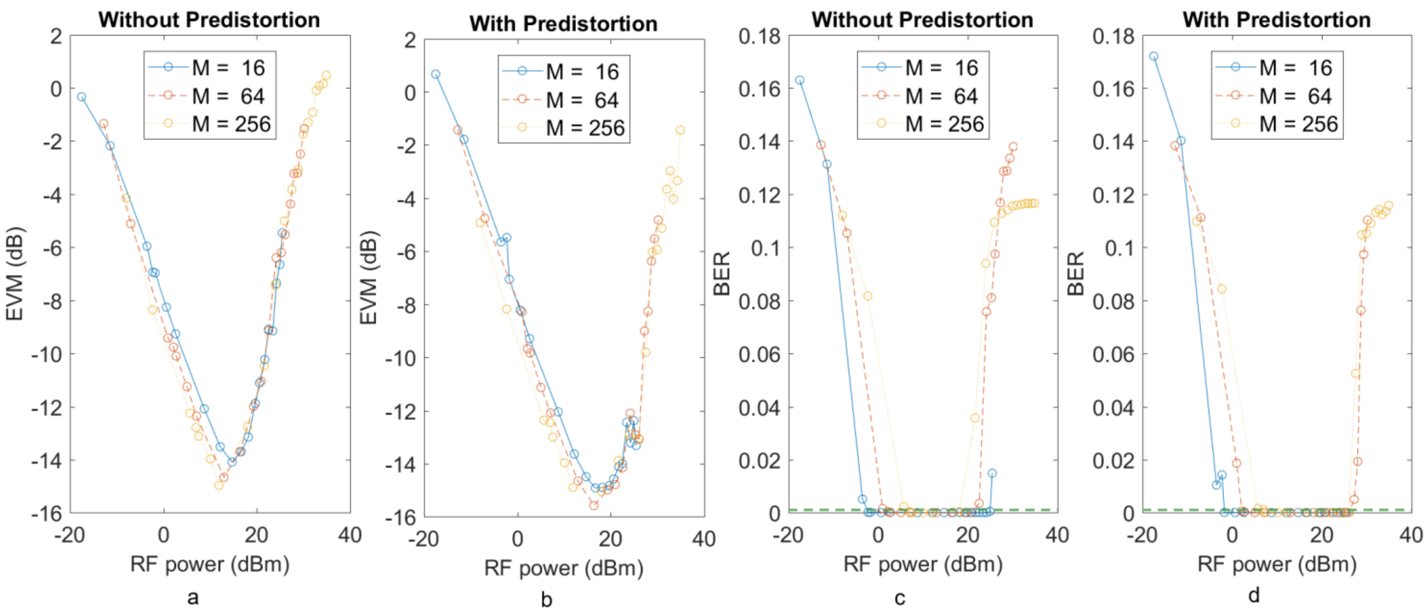


Figure 33 - Impact of M in the performance of the system for $\eta = -70\ dBm/Hz$, without fiber transmission, with Envelope Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER, with predistortion.

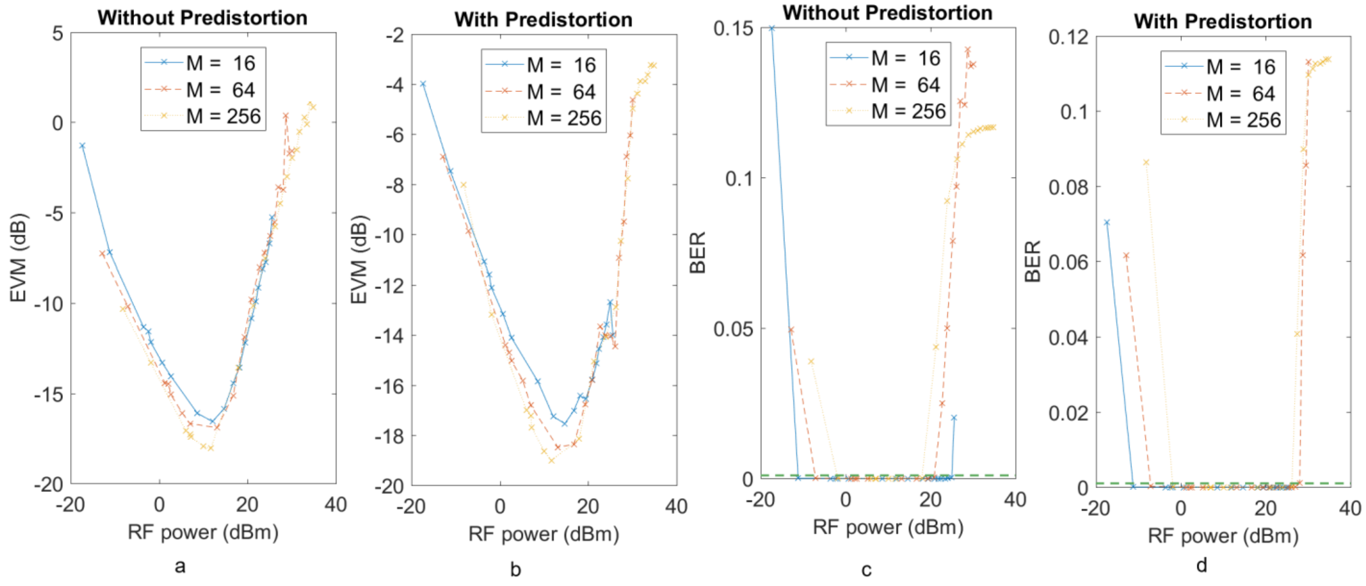


Figure 34 - Impact of M in the performance of the system for $\eta = -80$ dBm/Hz , without fiber transmission, with Envelope Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER, with predistortion.

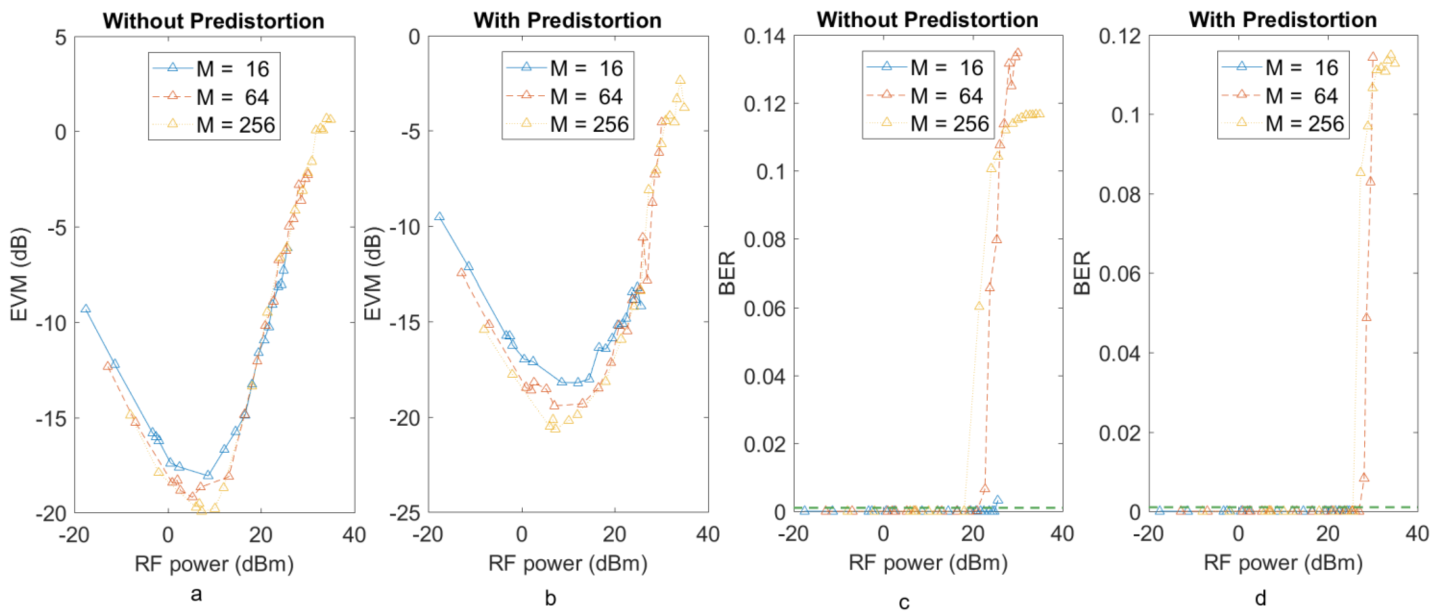


Figure 35 - Impact of M in the performance of the system for $\eta = -90\ dBm/Hz$, without fiber transmission, with Envelope Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER, with predistortion.

Figure 36 shows the performance of the systems with Coherent and Envelope Detection, when there is a fiber with 10 Km of length. Figure 36 is similar to Figure 31 and Figure 35: the performance without fiber is close to the performance with fiber; however, the performance of the system with fiber was expected to decrease. The reason behind the good performance of the system with fiber has been discussed in Section 1 of this chapter.

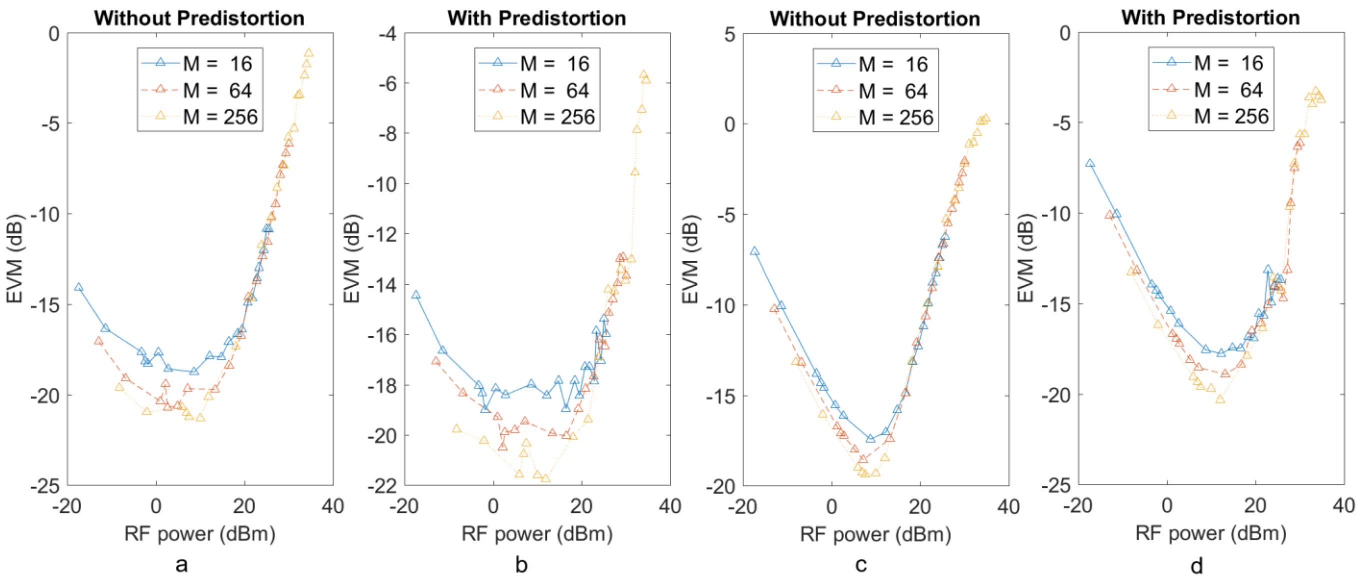


Figure 36 - Impact of M in EVM for $\eta = -90\ dBm/Hz$, with fiber transmission. a) Coherent Detection, without predistortion. b) Coherent Detection, with predistortion. c) Envelope Detection, without predistortion. d) Envelope Detection, with predistortion.

We conclude that the system can be optimized using $M = 256$. The following sections will use this value, unless it is explicitly stated otherwise.

5.4. The impact of the Bit Rate

The aim of this section is to study the impact on EVM of the bit rate, R, for different noise spectral densities.

In the previous section, we concluded that using less bandwidth optimizes the system. Since the bandwidth is directly proportional to R, the performance of the system is expected to decrease with an increase of R.

In fact, Figure 37, Figure 38 and Figure 39 show exactly what is expected: the performance of the system is optimized for $R = 5\ Gbit/s$. For higher values of R the system degrades as the bandwidth and the impact of noise in data increases. However, noise is not the only factor for the system's degradation: when the bandwidth increases, the side bands in OCS become closer and some intermodulation impacts start to appear. Figure 40 shows the optical spectrums for the studied values of bit rate. Figure 40-b shows some degree of intermodulation and in Figure 40-c, the intermodulation is more evident.

In these figures, the FEC limit is represented with a green dotted line. Without predistortion, only the system with a bit rate of 5 $Gbit/s$ is functional and only within some values of RF power. With predistortion, it is particularly interesting to see that the system with $R = 30\ Gbit/s$ becomes functional.

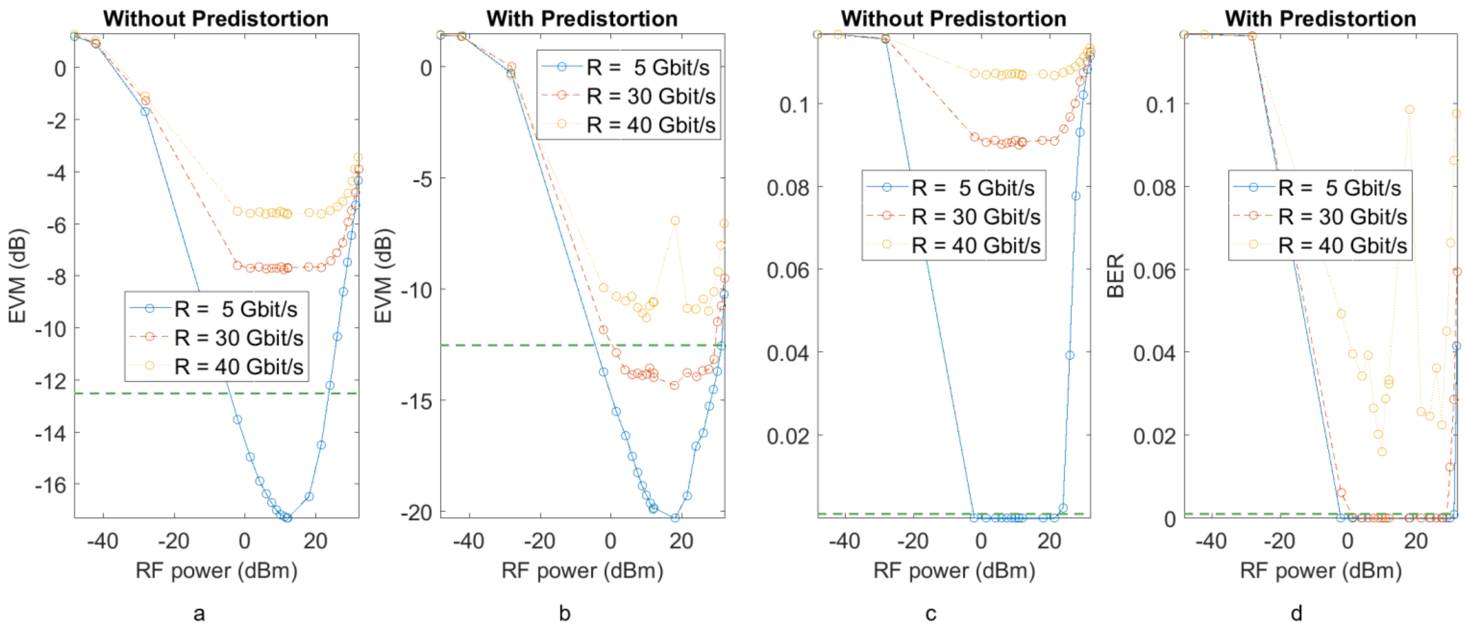


Figure 37 - Impact of R in the performance of the system for $\eta = -70\ dBm/Hz$, without fiber transmission, with Coherent Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER with predistortion.

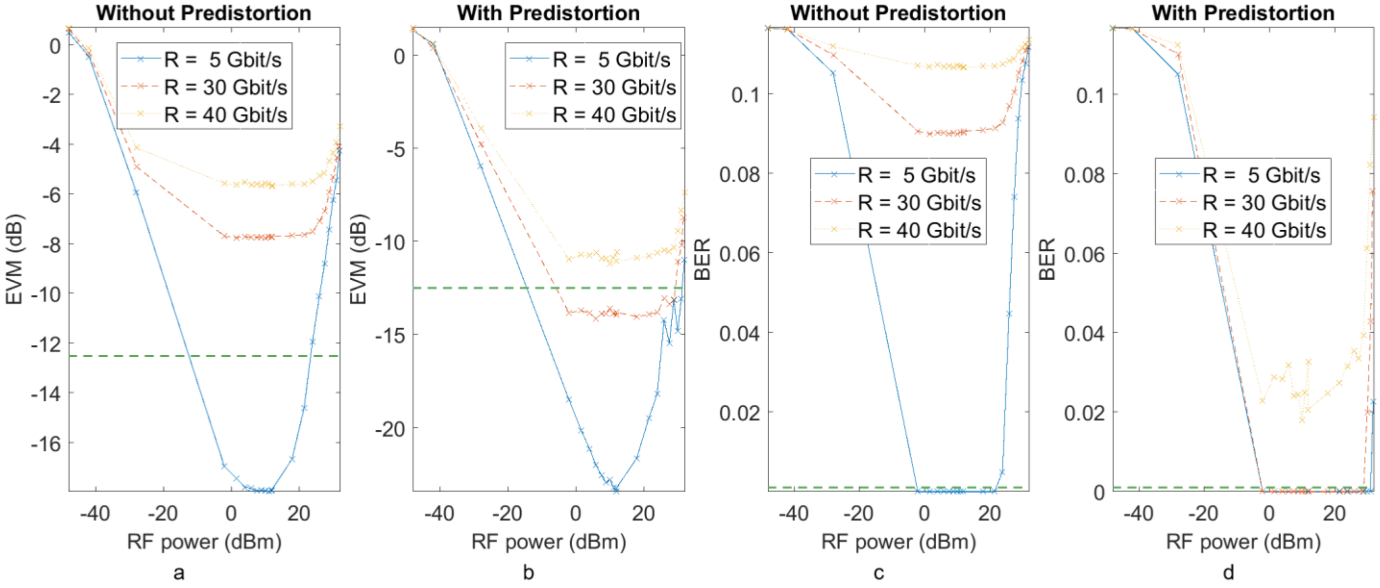


Figure 38 - Impact of R in the performance of the system for $\eta = -80\ dBm/Hz$, without fiber transmission, with Coherent Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER with predistortion.

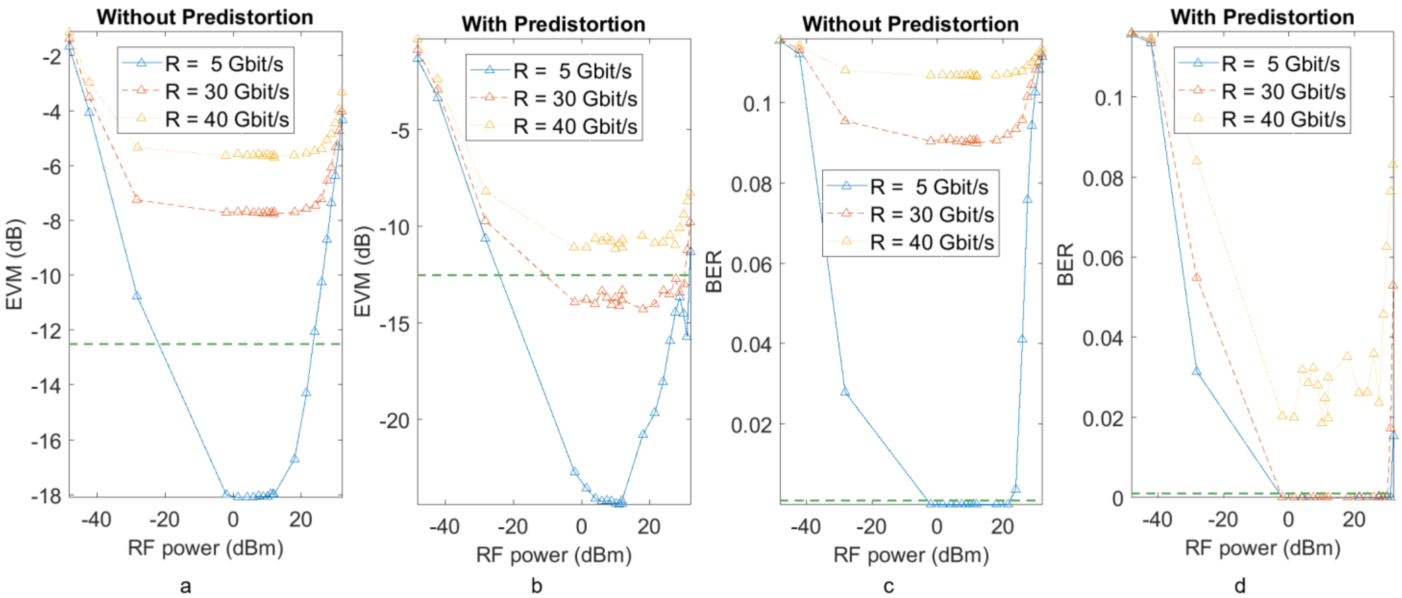


Figure 39 - Impact of R in the performance of the system for $\eta = -90\ dBm/Hz$, without fiber transmission, with Coherent Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER with predistortion.

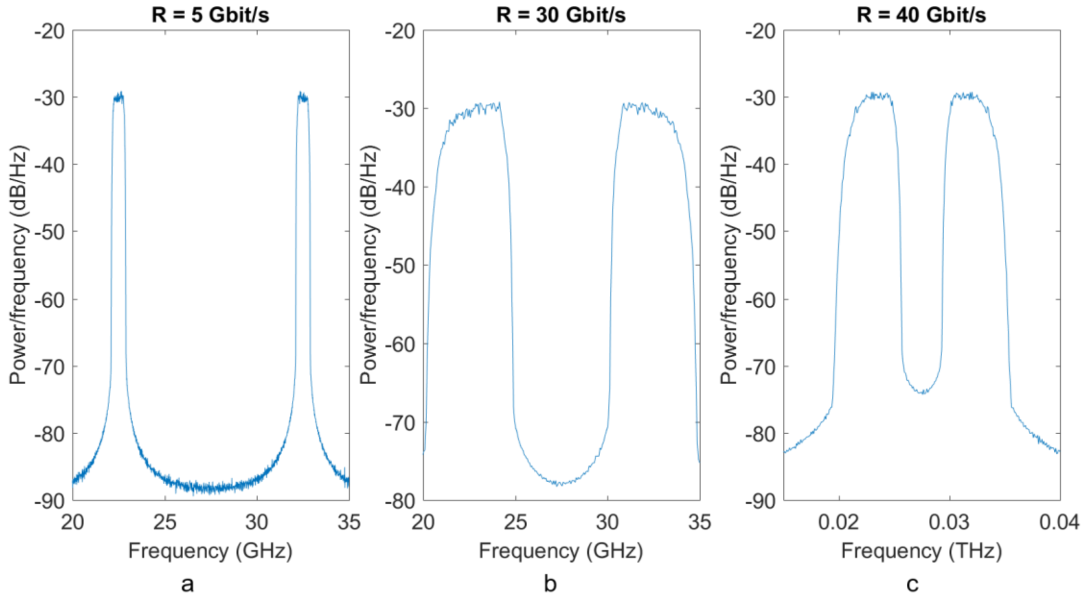


Figure 40 – Optical spectrum at the output of the MZM for different values of bit rate, with Coherent Detection. a) 5 Gbit/s. b) 30 Gbit/s. c) 40 Gbit/s.

Figure 41, Figure 42 and Figure 43 show the EVM and BER for the systems with Envelope Detection. In these systems, only a few values of RF power lead to EVM and BER below the FEC limit. The bit rate of 40 $Gbit/s$ is always above the limit.

Comparing the systems with Coherent Detection (Figure 37, Figure 38 and Figure 39) and Envelope Detection (Figure 41, Figure 42 and Figure 43), we see that the performance with 5 $Gbit/s$ is similar, while the performance for 30 $Gbit/s$ increases with ED. Figure 44 shows the optical spectrums for these systems. Comparing this figure with Figure 40, we see that the intermodulation for 30 $Gbit/s$ is less significant in the system with Envelope Detection and thus it has lower EVM.

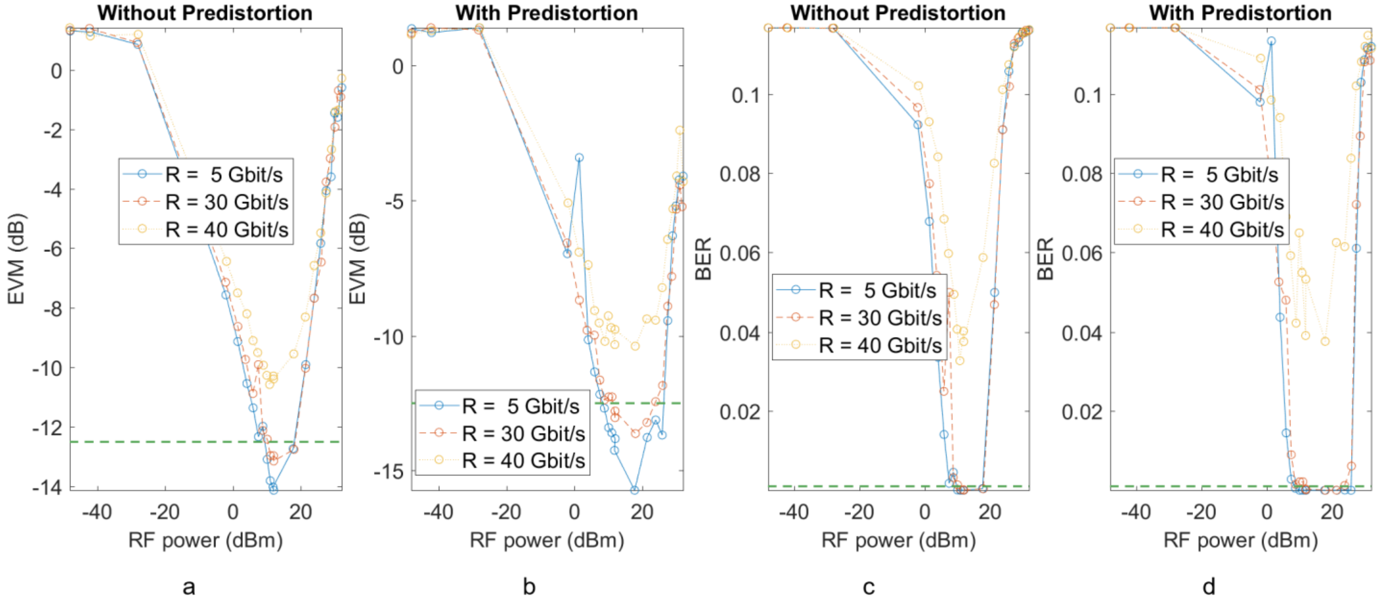


Figure 41 - Impact of R in the performance of the system for $\eta = -70\ dBm/Hz$, without fiber transmission, with Envelope Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER with predistortion.

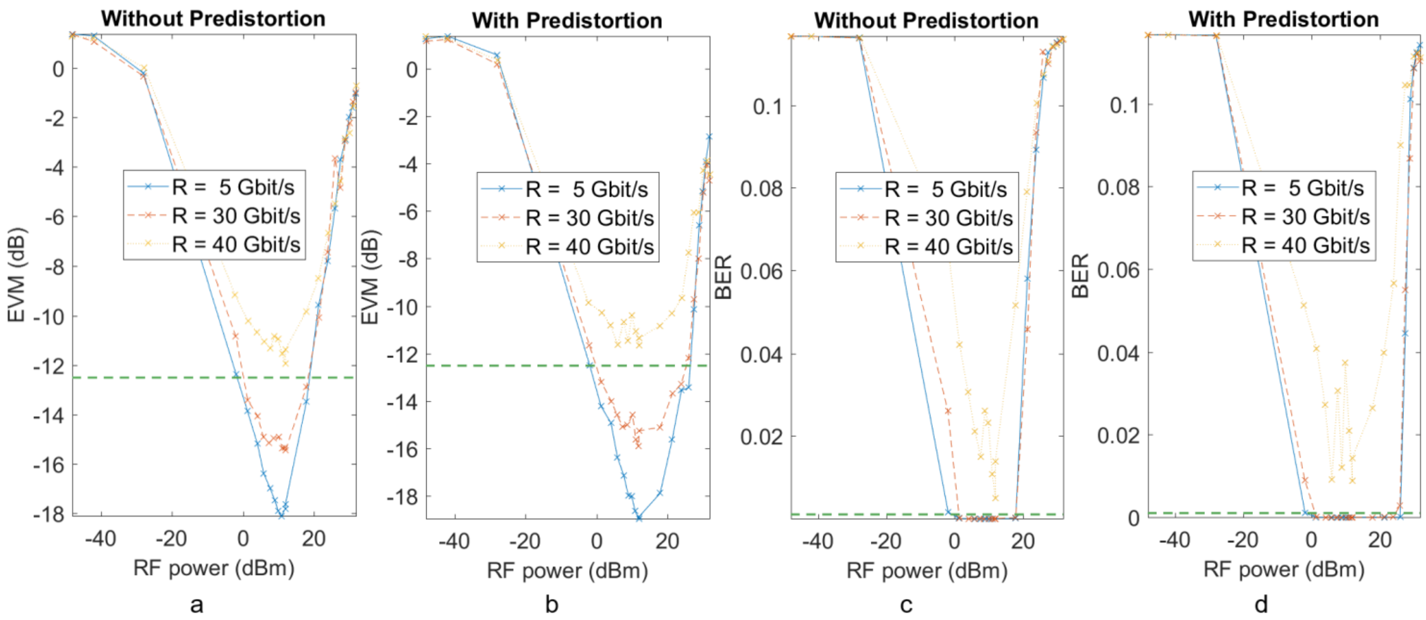


Figure 42 - Impact of R in the performance of the system for $\eta = -80\ dBm/Hz$, without fiber transmission, with Envelope Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER with predistortion.

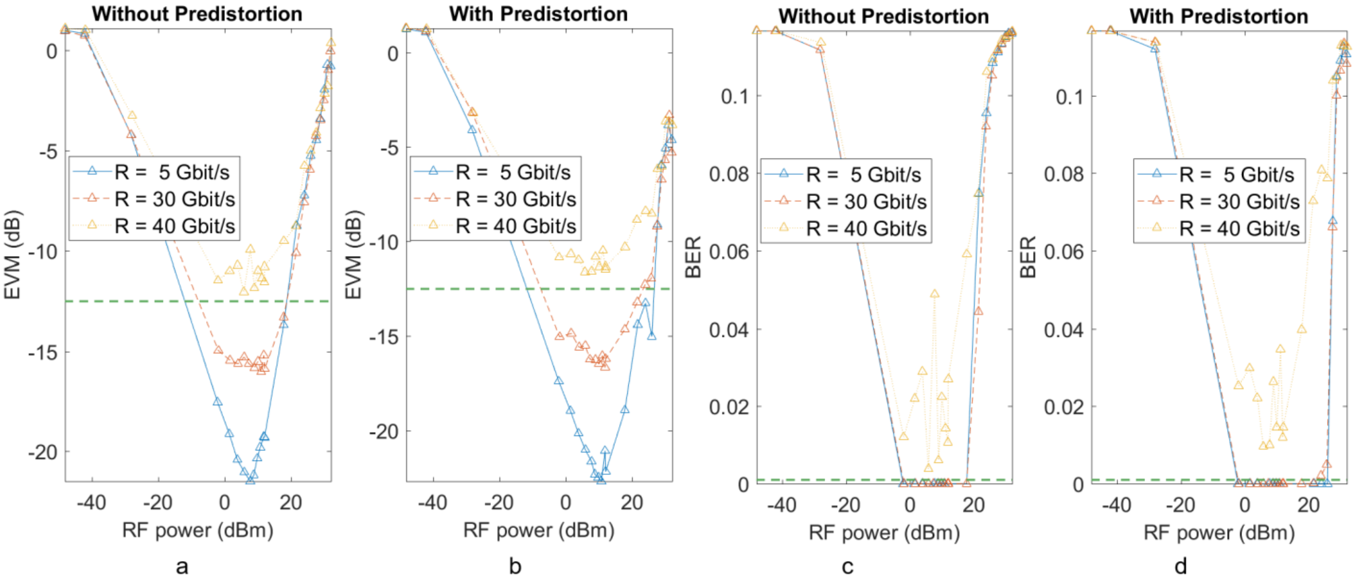


Figure 43 - Impact of R in the performance of the system for $\eta = -90\ dBm/Hz$, without fiber transmission, with Envelope Detection. a) Impact on EVM, without predistortion. b) Impact on EVM, with predistortion. c) Impact on BER, without predistortion. d) Impact on BER with predistortion.

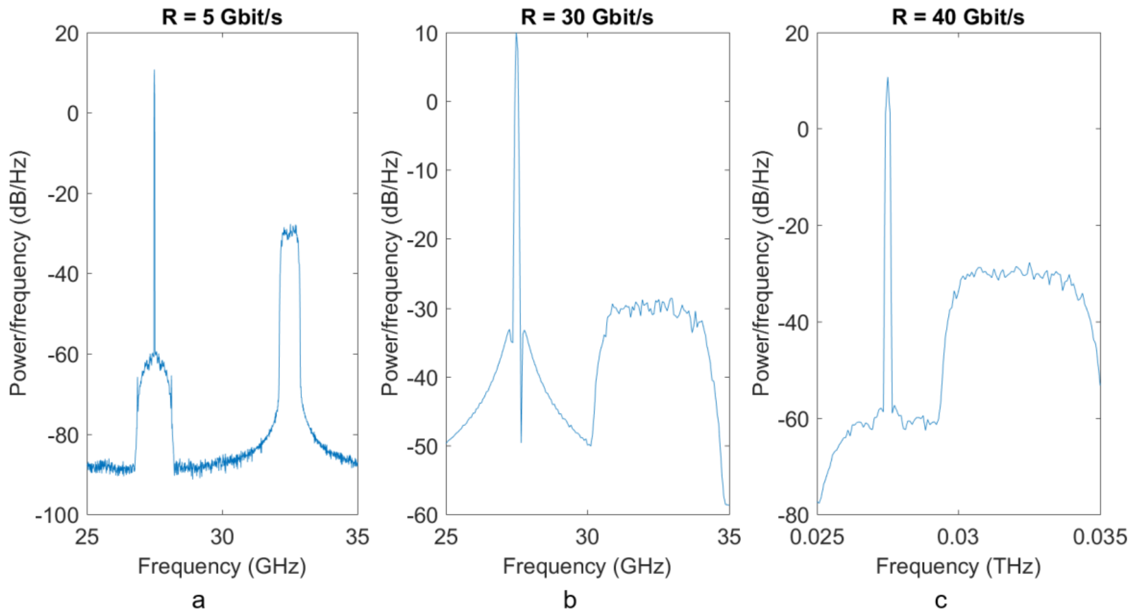


Figure 44 - Optical spectrum at the output of the MZM for different values of bit rate, with Envelope Detection. a) 5 Gbit/s. b) 30 Gbit/s. c) 40 Gbit/s.

The systems with Coherent Detection and Envelope Detection were also tested using a fiber with 10 Km of length; Figure 45 shows the EVM for these systems. Comparing Figure 45 to Figure 39 and Figure 43, we see that the performance for $R = 5\ Gbit/s$ is similar. The main differences are for higher values of R: the performance of the systems with fiber increases such that all values of bit rate are now operable for some values of RF power. Section 1 already presented an analysis on the improvement of the performance when there is transmission through fiber.

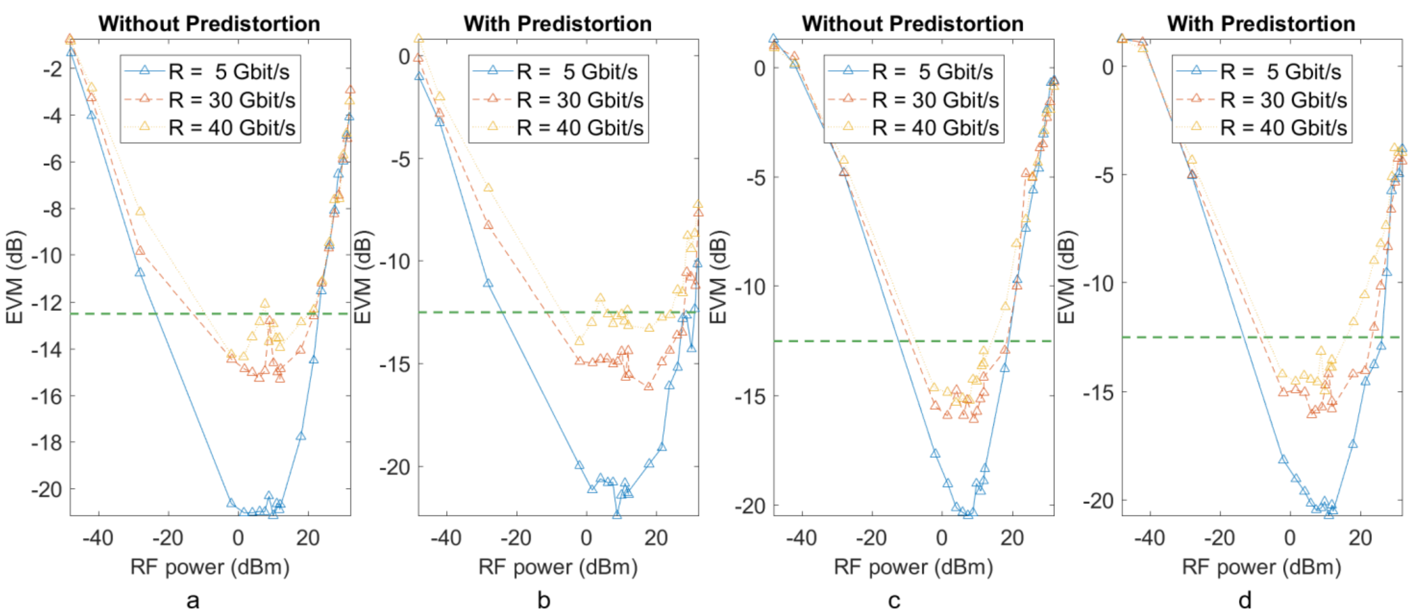


Figure 45 - Impact of R in EVM for $\eta = -90\ dBm/Hz$, with fiber transmission. a) Coherent Detection, without predistortion. b) Coherent Detection, with predistortion. c) Envelope Detection, without predistortion. d) Envelope Detection, with predistortion.

The bit rate that optimizes the system is 5 $Gbit/s$. However, due to the high requirements in data rate in future networks, a compromise between the bit rate and the performance must be made. Thus, in the following sections, a bit rate of 20 $Gbit/s$ will be used. Figure 46 shows the optical spectrum for this value of bit rate, and for an amplitude factor of 1; the FEC limit for such system is $-12.5\ dB$. The EVM in the system with Coherent Detection and without predistortion was close to 9% $(-10.5)\ dB$ while the EVM of the system with Coherent Detection and with predistortion was 2.22% $(-16.5\ dB)$. In Figure 46-b shows the optical spectrum for the system with Envelope Detection, for the same parameters. The EVM without predistortion was 0.998 % ($-20\ dB$) and 0.952% ($-20.2\ dB$) with predistortion.

The systems continue to be functional with some degree of intermodulation, except for the system with Coherent Detection without predistortion.

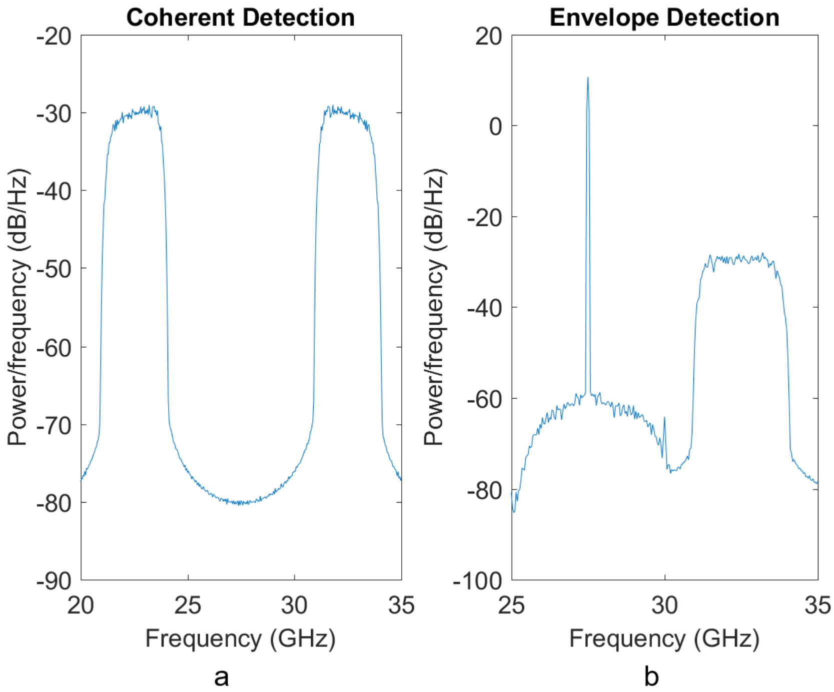


Figure 46 - Optical Spectrum at the output of the MZM for R = 20 Gbit/s. a) With Coherent Detection. b) With Envelope Detection.

5.5. The impact of the Fiber's Length[6]

The previous sections focus on analyzing parameters of the transmitter that optimize performance. This section focuses on the impact of the fiber's length in the performance, in the systems with Coherent and Envelope Detection, and without and with predistortion.

The systems had the following parameters: amplitude factor of 17.5, $M = 16$, $R = 20\ Gbit/s$ and $\eta = -90\ dBm/Hz$. The optimum number of symbols ($M = 256$) was not used because the distance between the symbols in the constellation is small and the system becomes less robust to distortions when higher lengths for the fiber are tested.

Figure 47 shows the EVM with the fiber's length for the systems with Coherent and Envelope Detection. The values for the fiber's length start at 0 and end in 50 Km, with a step of 2 Km.

[6] AM/AM plots and electrical spectrum for different values of fiber's length are presented in the appendix

Figure 47-a shows that the predistorter can improve the performance for smaller values of L. When the length of the fiber increases, the predistorter cannot compensate. For some values of length, the predistorter decreases the performance. Table 4 shows that for higher values of length, the shape of the AM/AM curve is lost and thus the inverse of the system is not accurately calculated. Even though, there is only one length that leads to a poor performance that might cause an amount of errors that FEC might not be able to correct.

Figure 47-b presents the EVM for the systems with Envelope Detection. Previous sections already discussed the poorer performance of Envelope Detection when compared to Coherent Detection with OCS. Thus, the higher values of EVM are expected. From this figure, we see that the predistorter can improve the performance until higher values of L. For values of L superior to 40 Km, the predistorter once again decreases the performance of the system. Table 4 shows the AM/AM plot and the spectrum for the system without predistortion. As we can see, these figures show that ED is not as affected by the fiber as Coherent Detection, as it continues to be linear (although with high levels of noise).

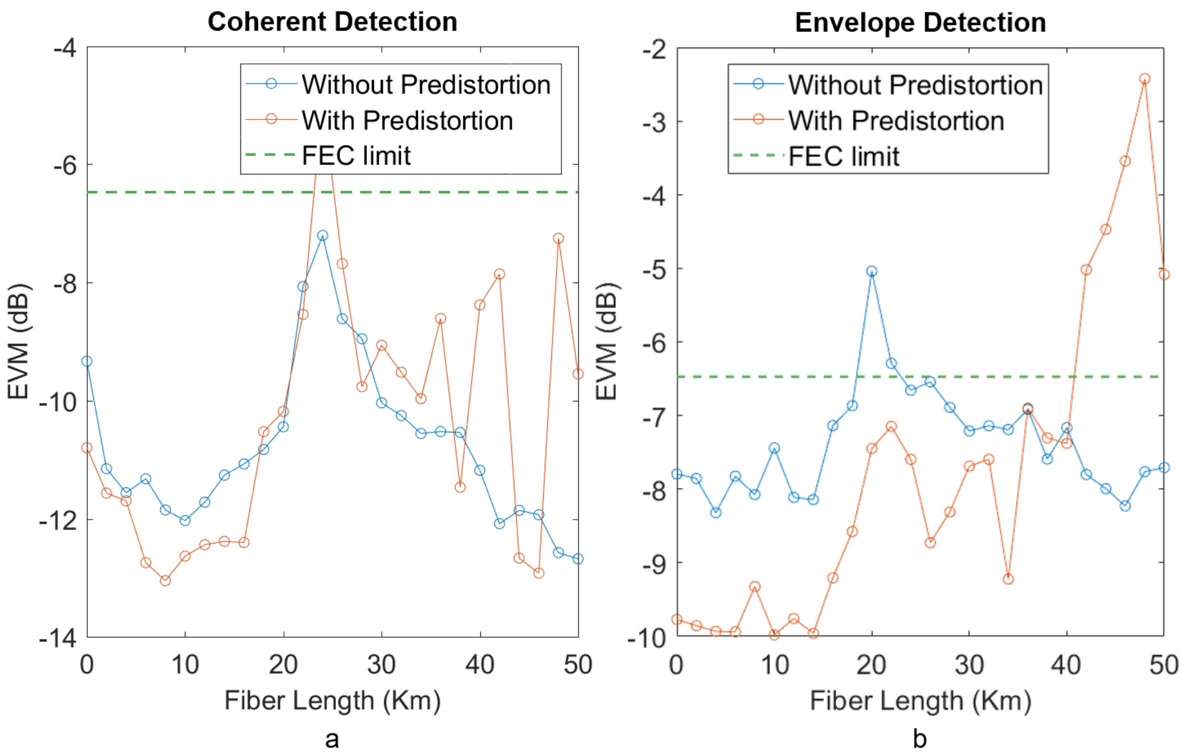


Figure 47 - Impact on EVM of the Fiber's Length. a) With Coherent Detection. b) With Envelope Detection.

5.6. The impact of the Lasers' Linewidth[7]

The linewidth of the laser introduces noise that can decrease the performance of the system. In this section, the impact of the linewidth is analyzed, using 100 different values from $10KHz$ to $10\ MHz$. The amplitude factor was 17.5.

Figure 48 shows the performance of the systems with Coherent Detection for different values of L. From Figure 48-a, we see that without using fiber the performance is barely affected by the laser's linewidth. When using fiber, we see that the EVM increases with the linewidth, which is expected. We can also conclude that for higher values of fiber's length, the sensitivity to the linewidth increases. Table 4shows the electrical spectrum for different values of linewidth; we can see that there is an increase of power in the laterals of the side bands, which explains the decrease of performance. From Figure 48-b, we see that for $L = 0$ the predistorter can increase the performance. For $L = 25\ Km$, it seems the predistorter has little impact; however, for values of linewidth just below $5000\ Hz$, the system without predistortion had EVM above the FEC limit while for the system with predistortion this limit was not exceeded. For $L = 50\ Km$, the predistorter decreases and increases the performance, being highly unstable due to the loss of shape in the AM/AM plot that is shown in Table 4.

In Figure 49, the performance of the systems with Envelope Detection is presented. From Figure 49-a, we see that the system with $L = 50\ Km$ has lower EVM compared to $L = 25Km$. Figure 47 already showed that the growth of EVM is not proportional to L, nonetheless it is interesting to see these results. Figure 49-b shows the system with predistortion. From this figure, we see that the predistorter increased the performance for $L = 0\ \ Km$. The system with $L = 25\ Km$ has now a value of EVM below the FEC limit while the EVM for the system with $L = 50\ Km$ became completely above.

From Figure 48 and Figure 49, we conclude that the system with Coherent Detection is more sensitive to an increase of linewidth. In Chapter 3, it was discussed that the optical carriers were generated with a coherent method, which subtracts the noise introduced by the linewidth; thus, the impact of the increase of linewidth is reduced though not negligible. However, OCS does not have the optical carrier at the optical frequency v_1 present in the spectrum.

The electrical spectrums for ED for some values of linewidth is present in Table 4. We see that ED is more robust to the noise created by the increase of linewidth; in fact, the AM/AM plot for Coherent Detection scatters for higher values of linewidth, as it is also shown in Table 4.

We conclude that the linewidth for all the analyzed values of L can be up to $7.5\ MHz$ with Coherent Detection. For the systems with Envelope Detection, the linewidth can go up to $8\ MHz$.

[7] AM/AM plots and electrical spectrum for different values of linewidth are presented in the appendix

The optimal linewidth is 1Hz, although this value is not achievable in real systems.

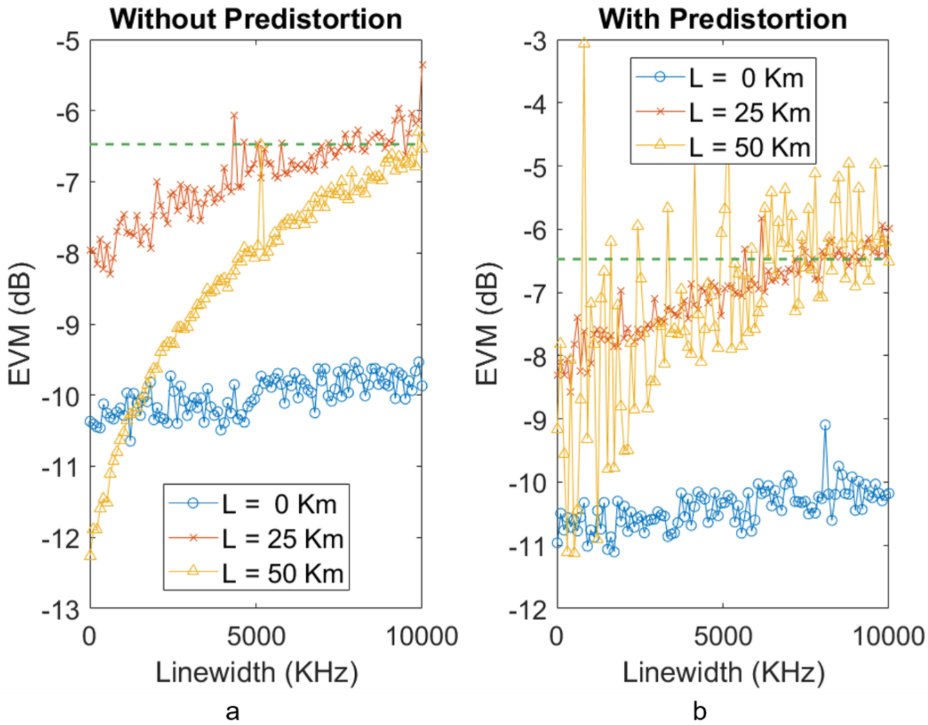


Figure 48 - Impact on EVM of the Linewidth for systems with Coherent Detection. a) Without Predistortion. b) With Predistortion.

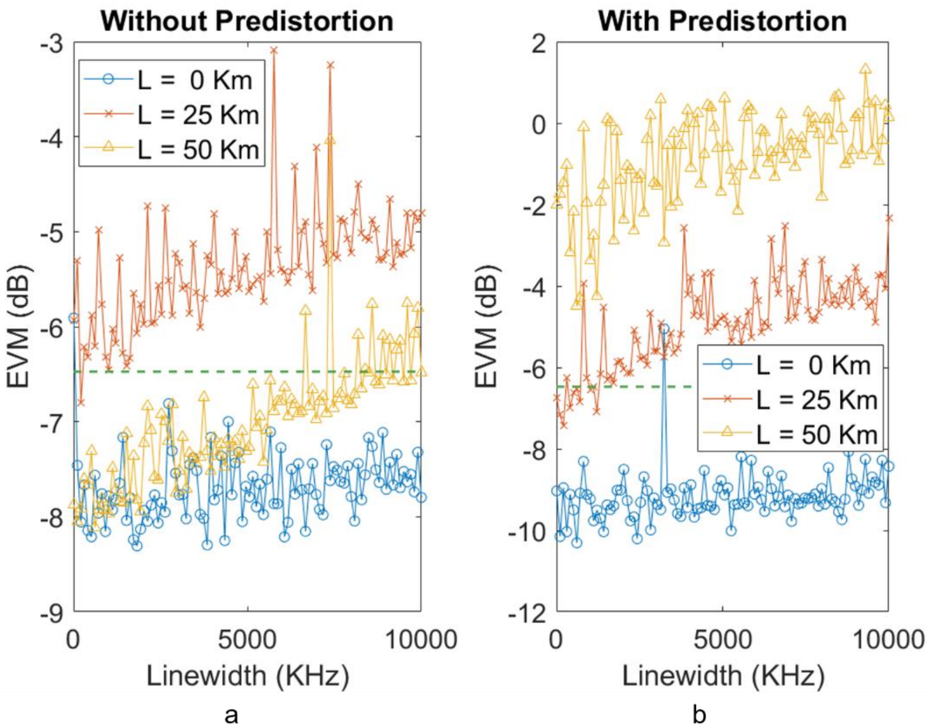


Figure 49 - Impact on EVM of the Linewidth for systems with Envelope Detection. a) Without Predistortion. b) With Predistortion.

Chapter 6 - Conclusions

In this chapter, the conclusions of this work are discussed in the first section. The second section presents suggestions of future work.

6.1. Conclusions

In this work, an Optical OFDM system operating at 60 GHz was implemented and analyzed. A predistorter was developed to compensate for the nonlinearities of such system. Parameters for the modulation and the transmission were changed to study the impact on the performance, measured mainly through EVM. Two detection schemes were compared: Coherent Detection with OCS and Envelope Detection with OSSB.

The Mach-Zehnder modulator is a relevant component of this system. Chapter 2 presents the model and introduces the modulation formats that can result from different electrical signals at the arms of the MZM. The resultant spectrums and performance were presented in Chapter 4; the format that lead to the best performance with Coherent Detection was OCS.

The first parameter of the modulation to be changed was the number of bits per symbol. Once again, the resulting spectrums were presented and the performance analyzed. It was concluded that the optimum number of bits per symbol is 256, since it leads to smaller bandwidth and thus the system is less affected by noise.

The impact of the bit rate was studied; we concluded that the optimum bit rate within the analyzed values would be 5Gbit/s because it leads to smaller bandwidth.

Throughout the aforementioned discussions, the use of the predistorter was analyzed. We conclude that the predistorter increases the performance when the main reason of the system's degradation is intermodulation; however, when the system is dominated by noise, the predistorter cannot compensate. We also concluded that the predistorter can compensate the distortions in AM/AM and AM/PM. The performance was also analyzed for schemes without and with fiber transmission. The predistorter can compensate the distortions for a fiber with 10 Km of length.

The impact of the fiber's length was also discussed. We concluded that the EVM is not linear with the fiber's length and that the predistorter can only compensate the distortion with small values of fiber's length.

Finally, the lasers' linewidth was changed. We concluded that the performance decreases with the linewidth. This is because higher values of linewidth introduce intensity noise which can be observed with the EVM graphics and the AM/AM plots present in the Appendix A. We also concluded that the impact of the linewidth increase is intensified by higher values of L.

6.2. Future Work

In this section, we suggest the following topics for future work:

1. Introduction of a wireless link and analysis of the performance of the system without and with predistortion of such link.

2. Study of the intermediate frequency for the electrical OFDM signal, $f_{IF,OFDM}$, and its impact on the performance of the system.

3. Use of an MB-OFDM system and the impact on the performance of the bands.

4. Implementation of automatic optimization of the values for K and Q in each run.

5. Experimental setup of the described system and performance analysis.

6. Development of a Graphical User's Interface with Matlab that allows the easy definition of the parameters and output results (e.g. Constellation without and with predistortion, Optical and Electrical spectrums, etc.).

Bibliography

[1] “Cisco VNI Global IP Traffic Forecast, 2015-2020.”

[2] P. Monteiro and A. Gameiro, “Convergence of Optical and Wireless Technologies for 5G,” in *Opportunities in 5G Networks*, F. Hu, Ed. CRC Press, 2016, pp. 179–215.

[3] E. Hossain and M. Hasan, “5G Cellular: Key Enabling Technologies and Research Challenges.”

[4] J. C. Z. Jeffrey G. Andrew, Stefano Buzzi, Wan Choi, Stephen V. Hanly, Angel Lozano, Anthony C. K. Soong, “What Will 5G Be?,” *IEEE J. Sel. Areas Commun. Top. Quantum Electron.*, vol. 32, no. 6, pp. 1065–1083, 2014.

[5] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, “A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges,” *Wirel. Networks*, vol. 21, no. 8, pp. 2657–2676, Nov. 2015.

[6] O. Omomukuyo, M. P. Thakur, and J. E. Mitchell, “Simple 60-GHz MB-OFDM Ultrawideband RoF System Based on Remote Heterodyning,” *IEEE Photonics Technol. Lett.*, vol. 25, no. 3, pp. 268–271, Feb. 2013.

[7] M. C. R. Medeiros, M. P. Thakur, S. Mikroulis, and J. E. Mitchell, “Wavelength shift tolerance of a heterodyne detection scheme for cost-efficient DWDM-PON / 60 GHz wireless integration,” in *Proc. SPIE 9387*, 2015, pp. 1–5.

[8] J. J. O’Reilly, P. M. Lane, J. Attard, and R. Griffin, “Broadband wireless systems and networks: an enabling role for radio-over-fibre,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 358, no. 1773, pp. 2297–2308, Aug. 2000.

[9] J. Capmany and D. Novak, “Microwave photonics combines two worlds,” *Nat. Photonics*, vol. 1, no. 6, pp. 319–330, Jun. 2007.

[10] “The 1st C-RAN International Workshop,” *China Mobile Research Institute*. [Online]. Available: http://labs.chinamobile.com/focus/C-RAN. [Accessed: 21-Apr-2010].

[11] B. Dang, R. Prasad, I. Niemegeers, M. Larrode, and A. J. Koonen, “Toward a Seamless Communication Architecture for In-building Networks at the 60 GHz band,” in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, 2006, pp. 300–307.

[12] ITU-R, “ITU-R Recommendations, Attenuation by atmospheric gases,” p. ITU-R P.676-5, 2001.

[13] T. Kurniawan, A. Nirmalathas, C. Lim, D. Novak, and R. Waterhouse, “Performance

analysis of optimized millimeter-wave fiber radio links," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 2, pp. 921–928, Feb. 2006.

[14] A. Nirmalathas, P. A. Gamage, C. Lim, D. Novak, and R. Waterhouse, "Digitized Radio-Over-Fiber Technologies for Converged Optical Wireless Access Network," *J. Light. Technol.*, vol. 28, no. 16, pp. 2366–2375, Aug. 2010.

[15] V. Jungnickel *et al.*, "Backhaul requirements for inter-site cooperation in heterogeneous LTE-Advanced networks," in *2013 IEEE International Conference on Communications Workshops (ICC)*, 2013, pp. 905–910.

[16] J. J. O'Reilly, P. M. Lane, R. Heidemann, and R. Hofstetter, "Optical generation of very narrow linewidth millimetre wave signals," *Electron. Lett.*, vol. 28, no. 25, pp. 2309–2311, Dec. 1992.

[17] M. C. R. Medeiros, P. Almeida, B. M. Oliveira, P. Laurêncio, and P. M. Monteiro, "Fiber Wireless-Optical Transceiver Architectures for 60 GHz Systems," pp. 1–4, 2017.

[18] R. Van Nee and R. Prasad, *OFDM For Wireless Multimedia Communications*. Artech House, Inc., 2000.

[19] C. G. Cho, Yong Soo and Kim, Jaekwon and Yang, Won Young and Kang, "Introduction to OFDM," in *MIMO-OFDM wireless communications with MATLAB*, John Wiley & Sons, 2010, pp. 111–152.

[20] I. Morita, "Optical OFDM for high-speed transmission," *2009 14th Optoelectron. Commun. Conf. OECC 2009*, pp. 8–9, 2009.

[21] Q. Yang, A. Al Amin, and W. Shieh, "Optical OFDM Basics," in *Impact of Nonlinearities on Fiber Optic Communications*, 2011, pp. 43–85.

[22] Y. Gao, J. Yu, J. Xiao, Z. Cao, F. Li, and L. Chen, "Direct-detection optical OFDM transmission system with pre-emphasis technique," *J. Light. Technol.*, vol. 29, no. 14, pp. 2138–2145, 2011.

[23] C.-H. Li, M.-F. Wu, C.-H. Lin, and C.-T. Lin, "W-band OFDM RoF System with Simple Envelope Detector Down-Conversion," in *Optical Fiber Communications Conference and Exhibition (OFC)*, 2015, pp. 6–8.

[24] A. Hekkala *et al.*, "Predistortion of Radio Over Fiber Links : Algorithms , Implementation , and Measurements," vol. 59, no. 3, pp. 664–672, 2012.

[25] A. Hekkala, J. Perttu, M. Lasanen, and L. C. Vieira, "Predistortion study of Radio over Fibre," pp. 1–8, 2010.

[26] A. Hekkala, M. Lasanen, L. C. Vieira, N. J. Gomes, and A. Nkansah, "Architectures for joint compensation of RoF and PA with nonideal feedback," *IEEE Veh. Technol. Conf.*, 2010.

[27] L.Vieira, “Digital Baseband Modelling and Predistortion of Radio over Fibre Links,” 2012.

[28] a. Hekkala and M. Lasanen, “Performance of adaptive algorithms for compensation of radio over fiber links,” *2009 Wirel. Telecommun. Symp.*, pp. 8–12, 2009.

[29] L. C. Vieira, N. J. Gomes, A. Nkansah, and F. Van Dijk, “Behavioral modeling of radio-over-fiber links using memory polynomials,” *Proc. - 2010 IEEE Int. Top. Meet. Microw. Photonics, MWP 2010*, pp. 85–88, 2010.

[30] L. Ding *et al.*, “A Robust Digital Baseband Predistorter Constructed Using Memory Polynomials,” *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, 2004.

[31] L. Aladren and P. Garcia-Ducar, “Digital predistortion optimization using normalization gain adjustment in wideband systems,” *Microw. …*, no. 1, pp. 420–423, 2013.

[32] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, “A generalized memory polynomial model for digital predistortion of RF power amplifiers,” *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, 2006.

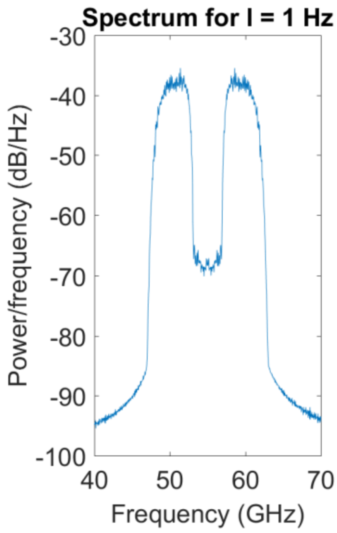
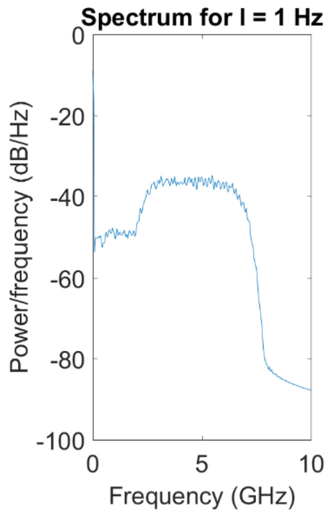
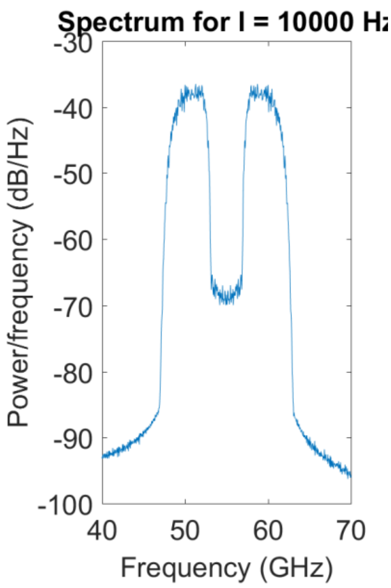
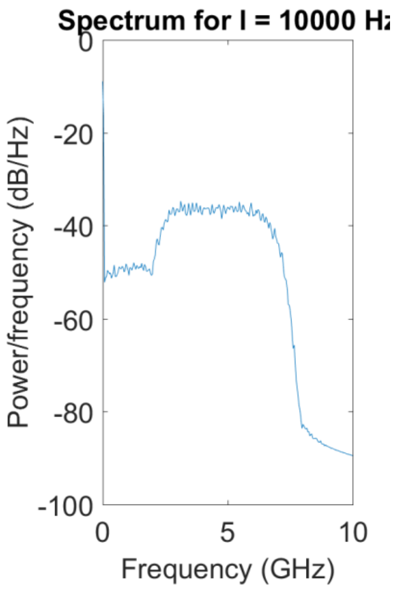
[33] R. S. Costa, “Métodos de transmissão de rádio sobre fibra,” Universidade de Aveiro, 2015.

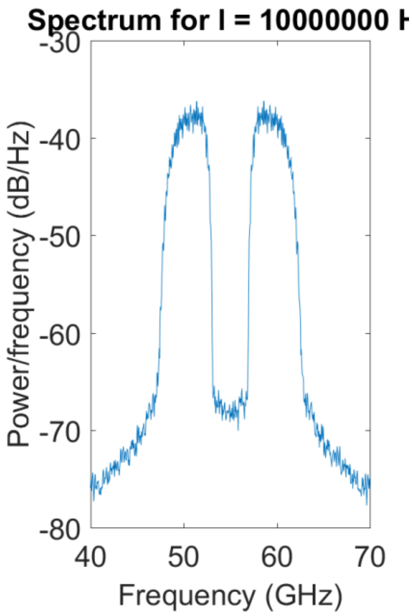
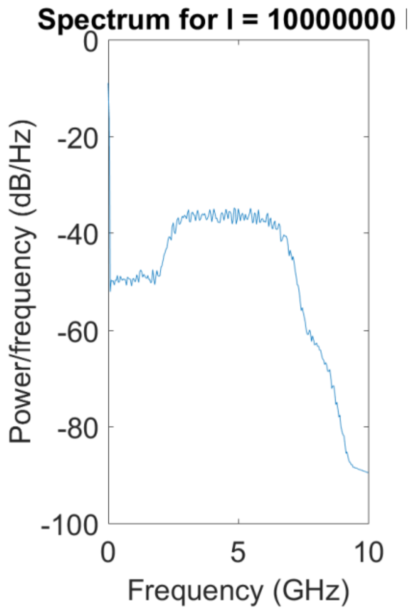
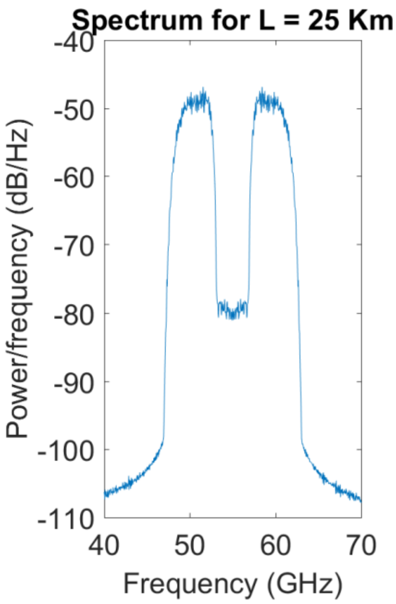
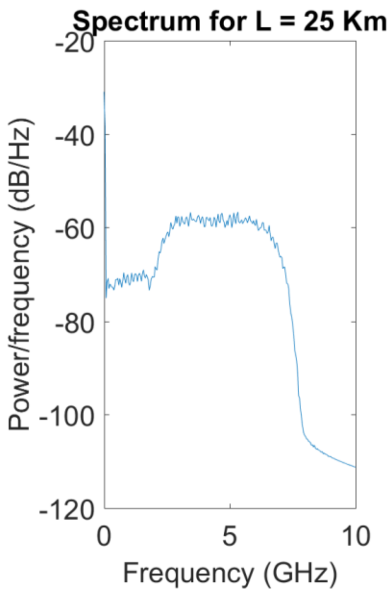
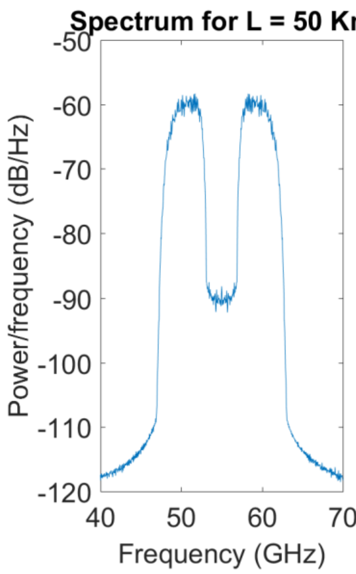
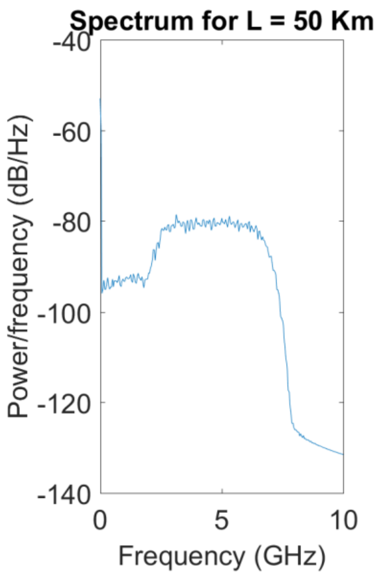
[34] C. R. J. Jr, W. A. Sethares, and A. G. Klein, “Chapter 13. Linear Equalization,” in *Software receiver design: build your own digital communication system in five easy steps*, Cambridge University Press, 2011, pp. 274–278.

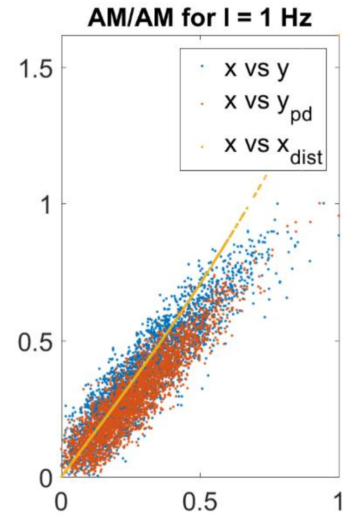
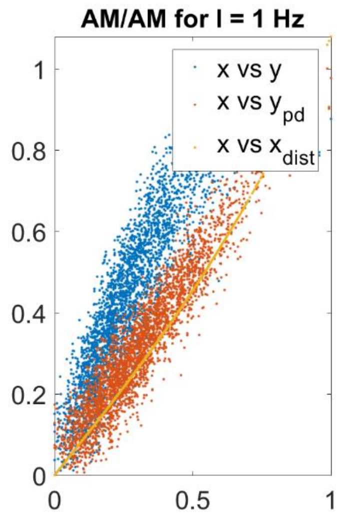
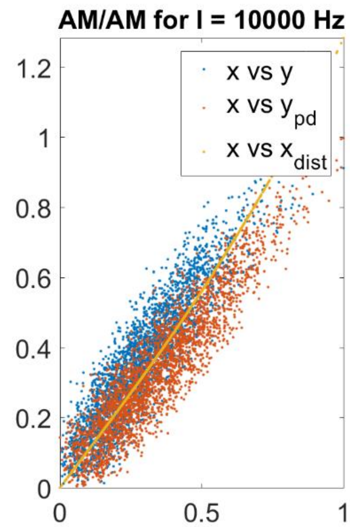
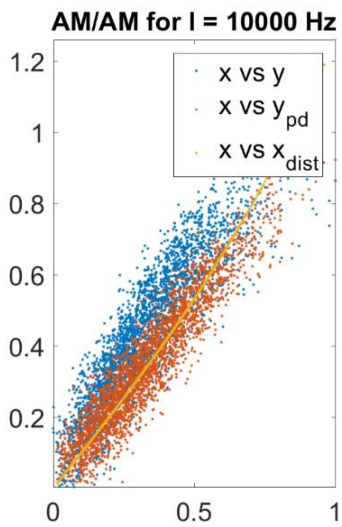
[35] R. A. Shafik, M. S. Rahman, and A. H. M. R. Islam, “On the extended relationships among EVM, BER and SNR as performance metrics,” *Proc. 4th Int. Conf. Electr. Comput. Eng. ICECE 2006*, no. December, pp. 408–411, 2007.

Appendix

Appendix A: Electrical Spectrum and AM/AM plot for different values of fiber's length and laser's linewidth

<table>
<tr><td rowspan="2">Fiber's Length</td><td rowspan="2">Linewidth</td><td colspan="2">Electrical Spectrum</td></tr>
<tr><td>Coherent Detection</td><td>Envelope Detection</td></tr>
<tr><td>0 Km</td><td>1 Hz</td><td>
</td><td>
</td></tr>
<tr><td>0 Km</td><td>10 KHz</td><td>
</td><td>
</td></tr>
</table>

0 Km	10 MHz		
25 Km	1 Hz		
50 Km	1 Hz		

Fiber's Length	Linewidth	AM/AM	
		Coherent Detection	Envelope Detection
0 Km	1 Hz		
0 Km	10 KHz		

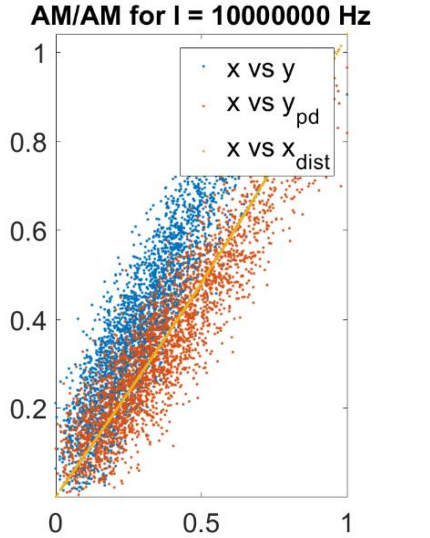
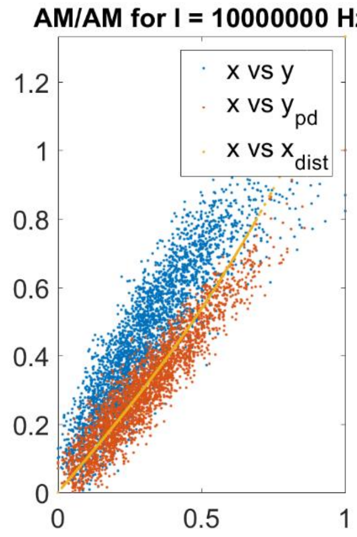
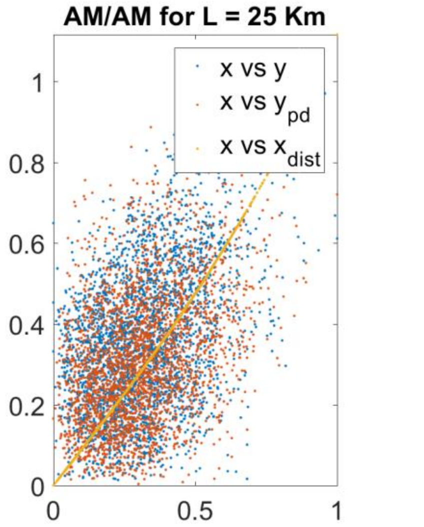
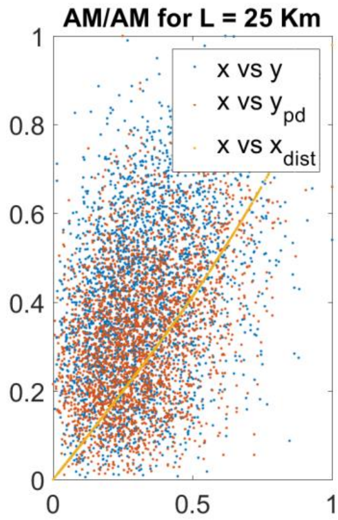
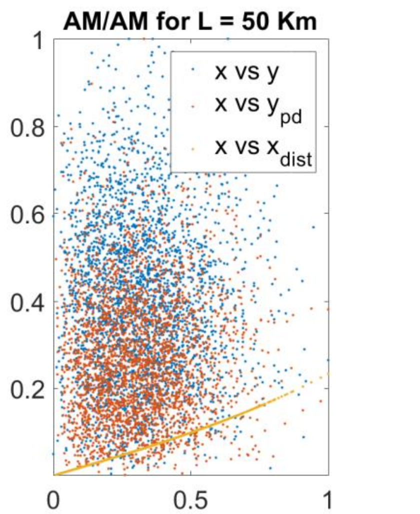
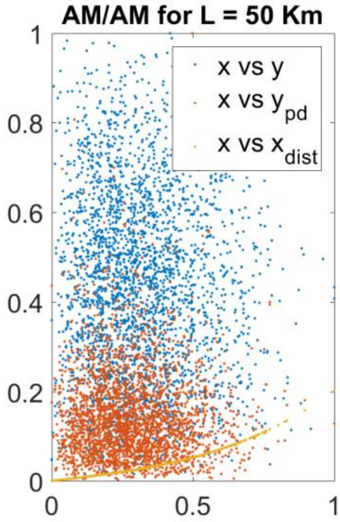
0 Km	10 MHz		
25 Km	1 Hz		
50 Km	1 Hz		

Table 4 – Electrical Spectrum and AM/AM plot for different values of fiber’s length and laser’s linewidth.

Appendix B: Mach-Zehnder

The transfer function of the Mach-Zehnder is given by equation (29), where $\Delta\beta(V_i)$ represents the phase shift dependent on the applied voltage Θ_i, $i = 1{,}2$ represents the upper and lower arm, respectively. L represents the length of interaction between the arms of the modulator and r is the power splitting ratio.

$$E_{out} = E_{in}(r\times \exp(j\Delta\beta(\Theta_1)L) + (1-r)\times\exp(j\Delta\beta(\Theta_2)L)) \tag{29}$$

Using Equations (30), (31) and (32), we can express Equation (29) as a function of the parameter V_π (voltage needed to cause a phase shift of π), resulting in Equation (33). $\Delta\beta_i$ is used as a simplified notation for $\Delta\beta(V_i)$, $i = 1{,}2$.

$$\Delta\beta_i = \frac{2\pi n_i}{\lambda_o} \tag{30}$$

$$n_i = n_o + \eta_i . \Theta_i \tag{31}$$

$$V_\pi = \frac{\lambda_o}{2L\eta} \tag{32}$$

$$E_{out} = E_{in}\left(r\times \exp\left(j\left(\frac{2\pi n_o L}{\lambda_o} + \frac{\pi}{V_\pi}\Theta_1\right)\right) + (1-r)\times \exp\left(j\left(\frac{2\pi n_o L}{\lambda_o} + \frac{\pi}{V_\pi}\Theta_2\right)\right)\right) \tag{33}$$

When the splitting ratio r is 0.5 (power splitting equal in both arms), we can derive Equation (34) using the identity $\cos a = \frac{\exp(ja)+\exp(-ja)}{2}$. This equation displays the phase and amplitude modulations using separate factors, being useful to understand the bias point of the Mach-Zehnder modulator.

$$E_{out} = \frac{E_{in}}{2}\times \exp\left(\frac{j\pi(\Theta_1+\Theta_2)}{2V_\pi}\times 2\times\frac{2\pi n_o L}{\lambda_o}\right)\times \cos\left(\frac{\pi}{2V_\pi}(\Theta_1-\Theta_2)\right) \tag{34}$$

Using different signals for Θ_1 and Θ_2, different modulation formats can be achieved.

Table 1 in Chapter 2 shows the DC bias in each electrical signal. Figure 9 shows the total DC bias for each modulation format. From Equation (34) and Table 1, we can derive the values in Figure 9, which are present in Table 5.

	Θ_1	Θ_2	$\Theta_1 - \Theta_2$	$\frac{\pi}{2V_\pi}(\Theta_1 - \Theta_2)$
OSSB Optical Single Side Band	$x_{RF} + \frac{V_\pi}{4}$	$H\{x_{RF}\} - \frac{V_\pi}{4}$	$x_{RF} + H\{x_{RF}\} + \frac{V_\pi}{2}$	$\frac{\pi}{2V_\pi}(x_{RF} + H\{x_{RF}\}) + \boldsymbol{\frac{\pi}{4}}$
ODSB Optical Dual Side Band	$x_{RF} + \frac{V_\pi}{2}$	$-\frac{V_\pi}{2}$	$x_{RF} + V_\pi$	$\frac{\pi}{2V_\pi}x_{RF} + \boldsymbol{\frac{\pi}{2}}$
OCS Optical Carrier Suppressed	$x_{RF} + \frac{V_\pi}{2}$	$-x_{RF} - \frac{V_\pi}{2}$	$2x_{RF} + V_\pi$	$\frac{\pi}{V_\pi}x_{RF} + \boldsymbol{\frac{\pi}{2}}$

Table 5 - DC bias point of the Mach-Zehnder Modulator

Appendix C: Least Squares Solution

The Least Mean Squares solution is a fast and simple algorithm when compared with RLS or the modified LMS solution presented in [6], [7]. It is a well-studied algorithm; however, it is difficult to find literature on this method since it is always stated and not explained. This section presents the Least Squares Solution.

Figure 50 shows the problem of finding an equalizer: finding a function f that is the inverse of the function of the channel. The source signal is known at the receiver when a training sequence is used.

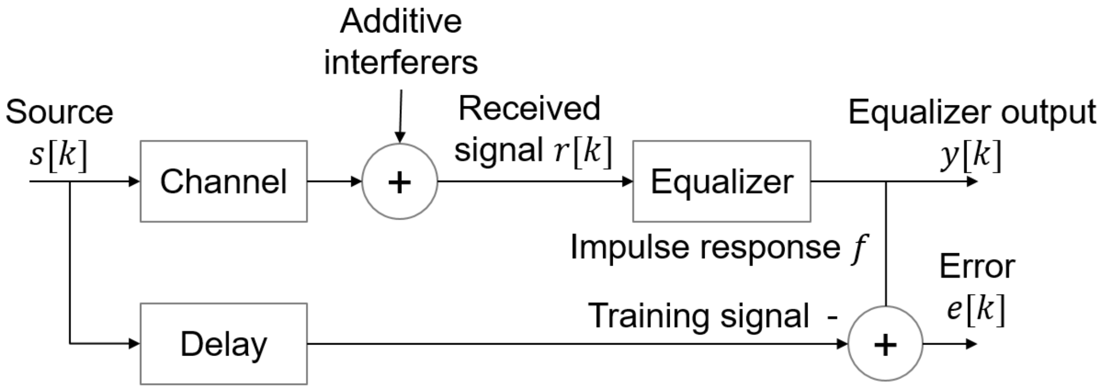


Figure 50 – Block diagram of the equalizer. Source [34]

The goal of an equalizer is to obtain an output $y[k]$ equal to the input $s[k]$ with some delay δ: $y[k] \approx s[k-\delta]$.

Equation (35) shows the output samples as a function of the input of the equalizer and the equalizer coefficients, $f = [f_0\ f_1 \ldots f_n]^T$. A more compact notation of this equation is present in Equation (36). The source and error signals can also be written using compact notation resulting in matrixes S and E, defined in Equations (37) and (38), respectively.

$$\begin{bmatrix} y[n+1] \\ y[n+2] \\ y[n+3] \\ \vdots \\ y[p] \end{bmatrix} = \begin{bmatrix} r[n+1] & r[n] & \cdots & r[1] \\ r[n+2] & r[n+1] & \cdots & r[2] \\ r[n+3] & r[n+2] & \cdots & r[3] \\ \vdots & \vdots & \ddots & \vdots \\ r[p] & r[p-1] & \cdots & r[p-n] \end{bmatrix} \times \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} \tag{35}$$

$$Y = RF \tag{36}$$

$$S = \begin{bmatrix} s[n+1-\delta] \\ s[n+2-\delta] \\ s[n+3-\delta] \\ \vdots \\ s[p-\delta] \end{bmatrix} \tag{37}$$

$$E = \begin{bmatrix} e[n+1] \\ e[n+2] \\ e[n+3] \\ \vdots \\ e[p] \end{bmatrix} \tag{38}$$

As Figure 50 shows, the recovery error is given by

$$e[k] = s[k-\delta] - y[k] \tag{39}$$

Using Equations (36) and (39), we obtain Equation (40):

$$E = S - Y = S - RF \tag{40}$$

The performance of the equalizer is often a measured by the summed squared error, given by Equation (41). The goal is to obtain the coefficients of f that minimize such error. Using Equations (37), (38) and (40) we can rewrite J_{LS} as Equation (42).

$$J_{LS} = \sum_{i=n+1}^{p} e^2[i] \tag{41}$$

$$J_{LS} = E^T E = S^T S - (RF)^T S - S^T RF + (RF)^T RF \tag{42}$$

$S^T S$ and $(RF)^T RF$ are scalars and because J_{LS} is a scalar, $(RF)^T S$ and $S^T RF$ must be scalars. Thus, $(RF)^T S = S^T RF$ because the transpose of a scalar is equal to itself.

We obtain the performance of the equalizer as the following equation:

$$J_{LS} = S^T S - 2S^T RF + (RF)^T RF \tag{43}$$

Let ψ be defined by Equation (44).

$$\psi = [F - (R^T R)^{-1} R^T S]^T (R^T R)[F - (R^T R)^{-1} R^T S] \tag{44}$$

It is important to note that ψ is simply the summed squared error rewritten. From Equation (40), we can write:

$$\begin{aligned} (R^T R)^{-1} R^T E &= (R^T R)^{-1} R^T S - (R^T R)^{-1} R^T RF = \\ (R^T R)^{-1} R^T E &= (R^T R)^{-1} R^T S - F \end{aligned} \tag{45}$$

Defining $A = F - (R^T R)^{-1} R^T S$, we obtain $\psi = A^T (R^T R) A$ from equation (44) and we have $A = -(R^T R)^{-1} R^T E$. Thus:

$$\begin{aligned} \psi &= [(R^T R)^{-1} R^T E]^T (R^T R)[(R^T R)^{-1} R^T E] = \\ &= [(R^T R)^{-1} R^T E]^T R^T E = \\ &= [(R^T R)^{-1} R^T E R]^T E \end{aligned} \tag{46}$$

Since $(R^T R)^{-1} R^T$ is the pseudo-inverse of R, we have $\psi = E^T E$ when there is no error computing the pseudo-inverse.

J_{LS} can be written in terms of ψ:

$$J_{LS} = \psi + S^T S - S^T R(R^T R)^{-1} R^T S \tag{47}$$

Again, if R is invertible, there is no error and we have:

$$\begin{aligned} &S^T S - S^T R(R^T R)^{-1} R^T S = \\ &= S^T S - S^T S = 0 \end{aligned} \tag{48}$$

Remember the goal is to minimize the summed squared error; i.e., we want the coefficients of F that lead to the minimum value of J_{LS}. From Equation (47) we see that F affects only ψ and the value of F that can accomplish this is in equation (49).

$$F^{\dagger} = (R^T R)^{-1} RS \tag{49}$$

Note that when $F = F^{\dagger}$, $\psi = 0$. When R is not invertible, we achieved the minimum error, given by J_{LS}^{min} in Equation (50).

$$J_{LS}^{min} = S^T S - S^T R(R^T R)^{-1} R^T S \tag{50}$$

Appendix D: Orthogonality in Multi-carrier systems

In multi-carrier systems, the carrier spacing is fundamental. In Orthogonal Frequency Division Multiplexing, the spacing allows for savings in bandwidth because the spectrum of each subcarrier overlaps with its neighbors. Equation (53) shows that with the OFDM spacing of $\Delta f = 1/T_{sym}$, there is no Inter-Channel Interference, where $(\,.\,)^*$ represents the complex conjugate. In OFDM, a symbol is defined by Equation (51). Equation (52) shows that each subcarrier's frequency, f_k, is a multiple of the spacing; f_o is the frequency of the first carrier. In Equation (53), the integral of two carriers k and i is calculated for the duration of an OFDM symbol, T_{sym}. From Equation (53) we conclude that two subcarriers are orthogonal and thus other subcarriers cause no interference in the interval of T_{sym} when demodulating.

$$\phi_k(t) = e^{j2\pi f_k t} \tag{51}$$

$$f_k = f_o + k\Delta f \tag{52}$$

$$\begin{aligned} &\int_0^{T_{sym}} \phi_k(t).\phi_i^*(t)\; dt = \int_0^{T_{sym}} e^{j2\pi f_k t}.\left(e^{j2\pi f_i t}\right)^* dt = \\ &= \int_0^{T_{sym}} e^{j2\pi f_k t} e^{-j2\pi f_i t}\, dt = \int_0^{T_{sym}} e^{j2\pi (k-i)_k t}\, dt = \begin{cases} 0, & k \neq i \\ T_{sym}, & k = i \end{cases} \end{aligned} \tag{53}$$

In the digital domain, the sampling of the signal is equally important. The sampling instant must be $t = lT_{sym} + nT_s$, where T_s is the length of a symbol (QAM or PSK). This way, the modulation and demodulation become the IFFT and FFT, respectively. Chapter 2 focuses on the discrete domain.

Appendix E: BER from EVM

Throughout this work, the performance was analyzed mainly through EVM. EVM measures the distance between the points in the constellation and the ideal points, without giving any direct information about the probability of error. However, the analysis of the probability of error is especially interesting. It is related to the EVM: with higher values of EVM, there is a bigger probability of error.

The Bit Error Rate (BER) represents the probability of an error occurring when deciding what bit was transmitted.

Based on [35], we can derive an equation that relates EVM with BER. From this article, several assumptions are made:

- Communication based on a stream of data
- System sampled at data rate
- Raised cosine pulses
- Gaussian noise
- The number of transmitted symbols, T, is much greater than the number of unique symbols in the constellation, N ($T \gg N$)

From [35], we present Equation (54), where P_b represents the probability of error, EVM_{RMS} represents the EVM of the constellation normalized to the mean square, L is the number of levels in each dimension of the constellation, M is the number of symbols and Q[.] is the Gaussian co-error function.

$$P_b \approx \frac{2\left(1-\frac{1}{L}\right)}{\log_2 L} Q\left[\sqrt{\left(\frac{3\log_2 L}{L^2-1}\right)\frac{2}{EVM_{RMS}^2 \log_2 M}}\right] \tag{54}$$

In the implemented system, all the assumptions are fulfilled. Thus, we must derive the exact equation for this system.

The number of levels, L, for an M-QAM system is given by $\sqrt{M}$. This is the number of amplitudes in the in-phase and quadrature axis in the ideal constellation.

The Gaussian co-error function was also replaced by the complementary error function ($erfc$) using Equation (55).

Equation (56) shows BER as a function of EVM.

$$Q[x] = \frac{1}{2} erfc\left[\frac{1}{\sqrt{2}}x\right] \tag{55}$$

$$\boldsymbol{P_b} \approx \frac{1-\frac{1}{L}}{\log_2 L} erfc\left[\sqrt{\frac{3\log_2 L}{L^2-1}\frac{1}{\boldsymbol{EVM_{RMS}^2}\log_2 M}}\right] \tag{56}$$

In telecommunication systems, the maximum BER recommended is $\sim 10^{-3}$ so that FEC can correct the errors. Thus, it is interesting to express this FEC limit in terms of EVM.

Equation (57) shows the EVM expressed in terms of BER. From this equation, we can derive the FEC limit by setting $P_b = 10^{-3}$. This value is dependent on the number of symbols, which is a parameter being changed throughout the analysis of the performance in Chapter 4.

$$\boldsymbol{EVM_{RMS}^2} = \frac{\left(1-\frac{1}{L}\right)3\log_2 L}{(L^2-1)(\log_2 M)^2 erfc^{-1}[\boldsymbol{P_b}]} \tag{57}$$

Appendix F: Optical System in Matlab

1. Calculate BER from EVM

Function's name: BERfromEVM.m

Computes the value of BER from a given EVM value, in an M-QAM system.

- Inputs:
  - EVM_in: value of EVM in the format EVMformat
  - m: number of bits per symbol
- Parameters
  - EVMformat: string to identify the format for the output value EVM
    - 'Linear': Linear units (ratio from 0 to 1)
    - 'Percentage': Percentage of EVM (from 0 to 100)
    - 'dB': EVM in dB
    - Default: dB

Source: Shafik, Rishad Ahmed, Md Shahriar Rahman, and AHM Razibul slam. "On the extended relationships among EVM, BER and SNR as performance metrics." Electrical and Computer Engineering, 2006. ICECE'06. International Conference on. IEEE, 2006.

```
function BER_out = BERfromEVM (EVM_in,m,EVMformat)
if(~exist('EVMformat','var'))
    EVMformat = 'dB';
end

switch(EVMformat)
    case 'Linear'
        %No change needed
        %This case is defined so it doesn't show warning dialog when linear format is chosen.
    case 'Percentage'
        EVM_in = EVM_in./100;

    case 'dB'
        EVM_in = 10.^(EVM_in./10);
    otherwise
        warning('Invalid string for EVM format')

end


M = 2^m; %number of symbols
%In an M-QAM system, the number of levels in each dimension (L) is given by
%sqrt(M) (number of amplitudes in in-phase or quadrature components)
L = sqrt(M);
```

```
%The input to the erfc function is erfc_factor/EVM^2
erfc_factor = 3*log2(L)/((L^2 -1)*log2(M));

%Pb = Pb_factor*erfc(erfc_factor/EVM^2);
Pb_factor = (1-1/L)/(log2(M));
BER_out = Pb_factor.*erfc(erfc_factor./(EVM_in.^2));
```

2. **Calculate EVM from BER**

Function's name: EVMfromBER.m

Compute maximum value of EVM that FEC can correct in an M-QAM system.

- Input:
  - m: number of bits per symbol
- Parameters
  - EVMformat: string to identify the format for the output value EVM
    - 'Linear': Linear units (ratio from 0 to 1)
    - 'Percentage': Percentage of EVM (from 0 to 100)
    - 'dB': EVM in dB
    - Default: Db
  - FEC_BERlimit: BER value; maximum probability of bit error that FEC can correct. Default value: 10^{-3}

Source: Shafik, Rishad Ahmed, Md Shahriar Rahman, and AHM Razibul slam. "On the extended relationships among EVM, BER and SNR as performance metrics." Electrical and Computer Engineering, 2006. ICECE'06. International Conference on. IEEE, 2006.

```
function EVM_out = EVMfromBER (m,EVMformat,FEC_BERlimit)
if(~exist('EVMformat','var'))
    EVMformat = 'dB';
end
if(~exist('FEC_BERlimit','var'))
    FEC_BERlimit = 1e-3;
end


M = 2.^m; %number of symbols
%In an M-QAM system, the number of levels in each dimension (L) is
given by
%sqrt(M) (number of amplitudes in in-phase or quadrature components)
L = sqrt(M);

%The input to the erfc function is erfc_factor/EVM^2
erfc_factor = 3.*log2(L)./((L.^2 -1).*log2(M));

%Pb = Pb_factor*erfc(erfc_factor/EVM^2);
```

```
Pb_factor = (1-1./L)./(log2(M));

erfc_out = erfcinv(FEC_BERlimit./Pb_factor);

EVM_out = sqrt(erfc_factor./erfc_out); %linear units

switch(EVMformat)
    case 'Linear'
        %No need; this was calculated before because it is also needed
for every other format
        %This case is defined so it doesn't show warning dialog when
linear format is chosen.
    case 'Percentage'
        EVM_out = EVM_out*100;

    case 'dB'
        EVM_out = 10*log10(EVM_out);

    otherwise
        warning('Invalid string for EVM format')

end
```

3. Mach-Zehnder ModulatorModel- OCS

Function's name: OCS_MZM.m

Optical Modulator Mach-Zehnder, OCS (Optical Carrier Supression)

In an ideal MZM, the optical carrier is suppressed by using symmetrical electrical signals at the input of the Mach-Zehnder.

A real MZM has a finite extinction ratio, which leads to different power splitting ratios in the MZM arms. With a finite extinction ratio, the carrier suppression is not obtained.

- Inputs:
  - E_in: Optical signal that will be modulated
  - RF: Data to modulate (electrical domain)
  - Vpi: tension needed to produce a phase shift of π
  - m_I: Insertion loss of the modulator

- Parameters
  - r1: power splitting in the arms of the Mach-Zehnder

- Output:
  - E_out: Modulated Optical signal

```
function E_out = OCS_MZM(E_in, RF,Vpi,m_I,r1)
```

```
j=sqrt(-1);
V1_dc=Vpi/2;
V2_dc=-Vpi/2;
d1=m_I*RF+V1_dc;
d2=-m_I*RF+V2_dc; %Optical Carrier supression is obtained with d2 = -d1.
E_out=E_in.*(r1*exp(j*pi*d1/Vpi)+(1-r1)*exp(j*pi*d2/Vpi));
end
```

4. Mach-Zehnder ModulatorModel- OSSB

Function's name: OSSB_MZM.m

Optical Modulator Mach-Zehnder, OSSB (Optical Single Side Band)

Using the Hilbert transform, one of the bands is suppressed, resulting in Single Side Band. An advantage to this method is the rise in power.

A disadvantage is the phase shift introduced by the Hilbert transform, which needs to be compensated in the receiver.

The optical carrier is not suppressed.

- Inputs:
  - E_in: Optical signal that will be modulated
  - RF: Data to modulate (electrical domain)
  - Vpi: tension needed to produce a phase shift of π
  - m_I: Insertion loss of the modulator

- Parameters
  - r1: power splitting in the arms of the Mach-Zehnder

- Output:
  - E_out: Modulated Optical signal

```
function E_out = OSSB_MZM(E_in, RF,Vpi,m_I,r1)
j=sqrt(-1);
V1_dc=Vpi/4;
V2_dc=-Vpi/4;
d1=m_I*RF+V1_dc;
d2=-m_I*imag(hilbert(RF))+V2_dc; %Hilbert transforms leads to single side band modulating format
E_out=E_in.*(r1*exp(j*pi*d1/Vpi)+(1-r1)*exp(j*pi*d2/Vpi));
end
```

5. Mach-Zehnder ModulatorModel- ODSB

Function's name: Intensity_MZM.m

Optical Modulator Mach-Zehnder, ODSB (Optical Dual Side Band)

The optical carrier is not suppressed.

- Inputs:
  - E_in: Optical signal that will be modulated
  - RF: Data to modulate (electrical domain)
  - Vpi: tension needed to produce a phase shift of π
  - m_I: Insertion loss of the modulator

- Parameters
  - r1: power splitting in the arms of the Mach-Zehnder

- Output:
  - E_out: Modulated Optical signal

```
function E_out = Intensity_MZM(E_in, RF,Vpi,m_I,r)

V1_dc = Vpi/2;
V2_dc = - Vpi/2;
d1 = m_I*RF + V1_dc;
d2 = V2_dc;

E_out = E_in.*( r*exp(1j*pi*d1/Vpi) +  (1-r)*exp(1j*pi*d2/Vpi)  );

end
```

6. Create Electrical Filter

Function's name: CreateElectricalFilter.m

Create a Super Gaussian electrical filter. The filter is symmetrical to f = 0 Hz.

- Inputs:
  - Fc: Central frequency of the filter
  - Bw: Bandwidth of the filter, at half maximum (FWHM)
  - Gain: Gain of the filter
  - FilterOrder: Order of the Super Gaussian
  - f_vector: vector with the frequencies to be computed
  - SbA: Stopband Attenuation in PSD units

- Output:
  - Hf: Filter, frequency domain

```matlab
function [Hf] = CreateElectricalFilter(Fc,Bw,Gain,FilterOrder,f_vector,SbA)
if(~exist('Fc','var'))
    Fc = 57.5e9;
end

if(~exist('Gain','var'))
    Gain = 1;
end
if(~exist('FilterOrder','var'))
    FilterOrder = 5;
end
if(~exist('f_vector','var'))

    load('PSD_Frequencies');
    f = PSD1.Frequencies;
else
    f = f_vector;
end
if(~exist('Bw','var'))
    Bw = 10e9;
end
if(~exist('SbA','var'))
    SbA = 40;
end


%% Super Gaussian Filter

%Parameters
x = f;
muP = Fc;
muN = -Fc;
A = Gain;

%FWHM: Full width at half maximum: bandwith at 3 dB
%The relationship between sigma and bandwith is:
%FWHM = 2*sqrt(2*ln2)*sigma, so sigma is:
sigma = Bw/(2*sqrt(2*log(2)));


% Electrical signal is in baseband; the filter must be symmetrical:
%Positive frequencies
exp_auxP = -0.5*(abs(x-muP)/sigma).^(FilterOrder+1);

%Negative Frequencies
exp_auxN = -0.5*(abs(-x+muN)/sigma).^(FilterOrder+1);

%Frequency response of filter
Hf = A.*(exp((exp_auxP)) + exp((exp_auxN)));

%We do not want ideal filter:
At_Rej=10^(-SbA/20); %linear units. Electrical Domain -> dB = 20log10(linear)
i_f = Hf <= At_Rej; %find amplitudes smaller than at_rej
Hf(i_f)=At_Rej; %replace those amplitudes with at_rej
```

```
Hf = fftshift(Hf);
```

7. Electrical Filter

Function's name: ElectricalFilter.m

Apply Super Gaussian Filter to the input.

- Input
  - Electrical signal, Base Band, time domain
- Parameters:

  Parameters from global variable PARA:
  - PARA.System.SampRate: Sample Frequency in Hz
  - PARA.Electrical.SuperGaussian.Fc: Center frequency of the filter, in Hz
  - PARA.Electrical.SuperGaussian.Gain: Gain of the filter
  - PARA.Electrical.SuperGaussian.FilterOrder: Order of the Super Gaussian
  - PARA.Electrical.SuperGaussian.Bw: Bandwidth of the filter; Full Width at Half Maximum
  - PARA.Electrical.SuperGaussian.SbA: Stopband Attenuation, in Power Spectral Density units
- Output
  - Electrical signal, Base Band, Filtered, time domain

```
function [output_signal] = ElectricalFilter(input_signal)

%% System Parameters
global PARA;

%% Filter Parameters
Npoints = length(input_signal);

%Renaming for clarity
Fs = PARA.System.SampRate;
Fc = PARA.Electrical.SuperGaussian.Fc;
Gain = PARA.Electrical.SuperGaussian.Gain;
FilterOrder =  PARA.Electrical.SuperGaussian.FilterOrder;
Bw = PARA.Electrical.SuperGaussian.Bw;
SbA = PARA.Electrical.SuperGaussian.SbA;
%% Input Spectrum

hPsd=spectrum.welch('Hamming',Npoints);
hopts=psdopts(hPsd);
set(hopts,'SpectrumType','twosided','NFFT',Npoints,'Fs',PARA.System.
SampRate, 'CenterDC',true);
signal_input = input_signal*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
PSD1=psd(hPsd,signal_input,hopts);
data_in=dspdata.psd([PSD1.Data],
PSD1.Frequencies,'Fs',PARA.System.SampRate);
```

```
%% Vector of frequencies to plot
f_vector = PSD1.Frequencies;


%% Computing Filter: Band Pass centered @ Fc and Bandwidth Bw

[Hf] = CreateElectricalFilter(Fc,Bw,1,FilterOrder,f_vector,SbA);


%% Filtering: Frequency Domain

Sf = fft(input_signal);
Hf = (reshape(Hf,size(Sf)));
Sf_filtered = Gain.*Sf.*Hf;
output_signal = ifft(Sf_filtered);
```

8. Create Optical Filer

Function's name: CreateOpticalFilter.m

Create a Super Gaussian optical filter. Frequencies relative to the optical center frequency v_o.

Eliminates the higher order components of Bessel function, introduced by the Mach-Zehnder Modulator.

- Inputs:
  - Fc: Central frequency of the filter
  - Bw: Bandwidth of the filter, at half maximum (FWHM)
  - Gain: Gain of the filter
  - FilterOrder: Order of the Super Gaussian
  - f_vector: vector with the frequencies to be computed
  - SbA: Stopband Attenuation in PSD units

- Output:
  - Hf: Filter, frequency domain

```
function [Hf] = CreateOpticalFilter(Fc,Bw,Gain,FilterOrder,f_vector,SbA)
if(~exist('Fc','var'))
    Fc = 27.5e9;
end

if(~exist('Gain','var'))
    Gain = 1000;
end
if(~exist('FilterOrder','var'))
    FilterOrder = 5;
end
if(~exist('f_vector','var'))
```

```
    load('PSD_Frequencies');
    f = PSD1.Frequencies;
else
    f = f_vector;
end
if(~exist('Bw','var'))
    Bw = 15e9;
end
if(~exist('SbA','var'))
    SbA = 40;
end


%% Super Gaussian Filter


%Parameters
x = f;
mu = Fc;
A = Gain;

%FWHM: Full width at half maximum: bandwith at 3 dB
%The relationship between sigma and bandwith is:
%FWHM = 2*sqrt(2*ln2)*sigma, so sigma is:
sigma = Bw/(2*sqrt(2*log(2)));


exp_aux = -0.5*(abs(x-mu)/sigma).^(FilterOrder+1);
Hf = A.*exp((exp_aux));



%We do not want ideal filter:
At_Rej=10^(-SbA/10);  %linear  units.  Optical  Domain  ->  dB  =
10log10(linear)
i_f = Hf <= At_Rej; %find amplitudes smaller than at_rej
Hf(i_f)=At_Rej; %replace those amplitudes with at_rej


Hf = fftshift(Hf);
```

9. Optical Filter

Function's name: OpticalFilter.m

Apply Super Gaussian Filter to the input

- Input
  - Optical signal, Base Band

- Parameters:

Parameters from global variable PARA:

  - PARA.System.SampRate: Sample Frequency in Hz

- PARA.Optical.SuperGaussian.Fc: Center frequency of the filter, in Hz
- PARA.Optical.SuperGaussian.Gain: Gain of the filter
- PARA.Optical.SuperGaussian.FilterOrder: Order of the Super Gaussian
- PARA.Optical.SuperGaussian.Bw: Bandwidth of the filter; Full Width at Half Maximum
- PARA.Optical.SuperGaussian.SbA: Stopband Attenuation, in Power Spectral Density units

- Output
  - Optical signal, Base Band, Filtered

```
function [output_signal] = OpticalFilter(input_signal)

%% System Parameters
global PARA;

%% Filter Parameters
Npoints = length(input_signal);

%Renaming for clarity
Fs = PARA.System.SampRate;
Fc = PARA.Optical.SuperGaussian.Fc;
Gain = PARA.Optical.SuperGaussian.Gain;
FilterOrder =  PARA.Optical.SuperGaussian.FilterOrder;
Bw = PARA.Optical.SuperGaussian.Bw;
SbA = PARA.Optical.SuperGaussian.SbA;
%% Input Spectrum

hPsd=spectrum.welch('Hamming',Npoints);
hopts=psdopts(hPsd);
set(hopts,'SpectrumType','twosided','NFFT',Npoints,'Fs',PARA.System.
SampRate, 'CenterDC',true);
signal_input = input_signal*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
PSD1=psd(hPsd,signal_input,hopts);
data_in=dspdata.psd([PSD1.Data],
PSD1.Frequencies,'Fs',PARA.System.SampRate);


%% Vector of frequencies to plot
f_vector = PSD1.Frequencies;


%% Computing Filter: Band Pass centered @ Fc and Bandwidth Bw

[Hf] = CreateOpticalFilter(Fc,Bw,1,FilterOrder,f_vector,SbA);


%% Filtering: Frequency Domain
```

```
Sf = fft(input_signal);
Hf = (reshape(Hf,size(Sf)));
Sf_filtered = Gain.*Sf.*Hf;
output_signal = ifft(Sf_filtered);
```

10. Memory Vector

Function's name: MemoryVector.m

Computes a matrix with highest non-linear order K and memory length Q.

If evenk is set to 1, computes only odd values of K

- Inputs
  - x : vector to transform to Matrix with memory

- Parameters
  - K : Order of Non-linearity for Volterra's polynomial vectors; default value is 5
  - Q : Memory length of Memory Polynomial; default value is 3
  - evenk: (0 or 1) if evenk=1 the nonlinearities of all orders are considered

- Outputs
  - X : Memory Matrix with number of rows equal to the length of input x

```
function X = MemoryVector(x,K,Q,evenk)

%% Checking Variables
N=length(x);
evenk=fix(evenk); % makes sure is an integer

% Checking if we have all the variables; if not, use values from Hekkala's paper
if(~exist('K','var'))
    K = 2;
    warning(['Order of Non-linearity not defined; using default value K = ' num2str(K)]);
end

if(~exist('Q','var'))
    Q = 1;
    warning(['Memory length not defined; using default value Q = ' num2str(Q)]);
end


%% Computing X:

idx=1;
```

```
  for k = 0:Q-1
      % Delay signal with k samples
      x_kq = [zeros(k,1); x(1:end-k)];
      for n = 1:1+evenk:K% another option is to not use the 2 order
nonlinearity n = 1:2:K;
          X(:,idx) = x_kq.*abs(x_kq).^(n-1);
          idx = idx+1;
      end
  end
  end
```

11. Plot Multiple Spectrums

Function's name: plot_multipleSpectrums.m

Function of Matlab for objects of type dspdata.psd that was modified to be able to plot more than one spectrum in one figure, with different colours

```
  function varargout = plot_multipleSpectrums(this,colorofgraph)
  %PLOT   Plot the response.

  %   Copyright 1988-2014 The MathWorks, Inc.

  if length(this) > 1
      
error(message('signal:dspdata:abstractfreqresp:plot:InvalidInputs'));
  end

  normfreq = get(this, 'NormalizedFrequency');

  % Determine the frequency range to plot.
  freqrange = 'whole';
  if ishalfnyqinterval(this)
      freqrange = 'half';
  end
  centerdc = getcenterdc(this);

  % Create a new plot or reuse an available one.
  hax = newplot;hold on;

  % Get the data from this object.
  [H,                               W]                               =
getdata(this,isdensity(this),plotindb(this),normfreq,freqrange,centerd
c);

  % Set up the xlabel.
  if normfreq
      W    = W/pi;
      xlbl = getfreqlbl('rad/sample');
  else
      [W, ~, xunits] = engunits(W);
      xlbl = getfreqlbl([xunits 'Hz']);
  end
```

```
  % Plot the data.

  h = line(W, H, 'Parent', hax,'Color', colorofgraph);

  if((strcmp(this.Name, 'Power Spectral Density') || strcmp(this.Name,
'Mean-Square Spectrum')) && ~isempty(this.ConfInterval))
      CI = this.ConfInterval;
      CL = this.ConfLevel;
      Hc = db(CI,'power');

      % Plot the Confidence Intervals.
      for i=1:size(H,2)
          baseColor = get(h(i,1),'Color');
          h(i,2) = line(W, Hc(:,2*i-1),'Color',baseColor,'LineStyle','-
.','Parent',hax);
          h(i,3)  =  line(W,  Hc(:,2*i),'Color',baseColor,'LineStyle','-
.','Parent', hax);
      end

      % convert to row vector for backwards compatibility
      h = h(:)';

      % re-order the children so first two legend entries are 'correct'.
      hc = get(hax,'Children');
      if numel(hc)==numel(h)

set(hax,'Children',reshape(reshape(hc,numel(hc)/3,3)',1,numel(hc)));
      end

      % re-save as rows for backwards compatibility.
      h = h';

      if strcmp(this.Name, 'Power Spectral Density')
        Estimate                                                          =
getString(message('signal:dspdata:abstractfreqresp:plot:PowerSpectralD
ensity'));
      else
        Estimate                                                          =
getString(message('signal:dspdata:abstractfreqresp:plot:PowerSpectrum'
));
      end
      Interval                                                            =
getString(message('signal:dspdata:abstractfreqresp:plot:ConfidenceInte
rvalPct',num2str(CL*100)));
      legend(Estimate,Interval,'Location','best');
  end

  xlabel(hax, xlbl);

  % Set up the ylabel
  ylabel(hax, getylabel(this));

  title(hax,
getTranslatedString('signal:dspdata:dspdata',gettitle(this)));

  set(hax, 'Box', 'On', ...
```

```
    'XGrid', 'On', ...
    'YGrid', 'On', ...
    'XLim', [min(W) max(W)]);

% Ensure axes limits are properly cached for zoom/unzoom
resetplotview(hax,'SaveCurrentView');

if nargout
    varargout = {h};
end

% [EOF]
```

12. Create Figure

Function's name: CreateFigure.m

Create figure from specified parameters or add plots to given figure.

- Inputs:
  - GraphType: Type of Graphic to be plotted

    Accepted Values: 'AM-AM', 'AM-PM', 'Error', 'EVM', 'Const', 'Spectrum2', 'Spectrum', 'Const2'

    This way, each type of graph always has same marker and formats
  - x: data to be plotted along x-axis (or, for 'Spectrum2' and 'Const2', the first plot) (for 'Spectrum', x can be the dspdata.psd structure to plot)
  - y: data to be plotted along y-axis (or, for 'Spectrum2' and 'Const2', the second plot) (for 'Spectrum', y can be the dspdata.psd structure to plot)
  - NumberOfImagesPerPageWidth: Number of images per page width

    The size of the graphs will be according to how many we want to be side-by-side in one page width
  - PageWidth [cm]: Width of the page in which the graphics will be side-by-side (does not include page margins)
  - GraphHeight [cm]: Height of the graph
  - h: hande to previously created figure

    If we want to add plots to one figure which was already created. Be sure to set hold on before calling this function
  - GraphFontSize [pt]: Font Size of every text in the current figure
  - L1, L2: Legend; L1 corresponds to plot of x and L2 to the plot of y (Only for 'Const2' graph type)

```
function [h] = CreateFigure(GraphType,x,y,NumberOfImagesPerPageWidth,PageWidth,GraphHeight,h,GraphFontSize,L1,L2)
%% Parameters of colors: Default colors
% These are the default colors of the plots
c1 = [0     4.470000000000000e-01     7.410000000000000e-01];
```

```
  c2   =   [8.500000000000000e-01             3.250000000000000e-01
9.800000000000000e-02];


  %if there is no handle to figure, create one
  if(~exist('h','var') || h == -1 || isempty(h))
      if(exist('PageWidth','var')                                    &&
exist('NumberOfImagesPerPageWidth','var')                            &&
exist('GraphHeight','var'))
          GraphWidth = PageWidth/(NumberOfImagesPerPageWidth+0.01);
          h = figure('units','centimeters','position',[10 3 GraphWidth
GraphHeight]);
      else
          error('Missing data to create figure. Please insert Page
Width, Number of Images per Page Width and Graph Height');
      end
  end


  if(exist('x','var') && exist('y','var'))

      figure(h);

      switch(GraphType)

          case 'EVM'

              plot(x,y,'o-');
              if(exist('GraphFontSize','var'))
                  set(findall(h,'-
property','FontSize'),'FontSize',GraphFontSize);
              end

          case {'AM-AM','AM-PM'}
              plot(x,y,'.');
              if(exist('GraphFontSize','var'))
                  set(findall(h,'-
property','FontSize'),'FontSize',GraphFontSize);
              end

          case 'Error'
              plot(x,y,'x-');
              if(exist('GraphFontSize','var'))
                  set(findall(h,'-
property','FontSize'),'FontSize',GraphFontSize);
              end

          case 'Const' %Constellation
              plot(x,y,'.','color',[0.5 0.5 0.5])
              axis([-1.5 1.5 -1.5 1.5]);
              if(exist('GraphFontSize','var'))
                  set(findall(h,'-
property','FontSize'),'FontSize',GraphFontSize);
              end

          case 'Const2' %Two constellations in one plot
              h1 = plot(real(x(1)),imag(x(1)),'.','color',c1);
```

```
                hold on;
                plot(real(x(2:end)),imag(x(2:end)),'.','color',c1);
                h2 = plot(real(y(1)),imag(y(1)),'.','color',c2);
                plot(real(y(2:end)),imag(y(2:end)),'.','color',c2)
                axis([-1.5 1.5 -1.5 1.5]);
                if( exist('L1','var') && exist('L2','var'))
                    legend([h1 h2],{L1,L2});
                end

                if(exist('GraphFontSize','var'))
                    set(findall(h,'-
property','FontSize'),'FontSize',GraphFontSize);
                end

            case 'Spectrum2' %Two Spectrums in one plot

                plot_multipleSpectrums(x,c1); hold on;
                plot_multipleSpectrums(y,c2);

            case 'Spectrum' %Spectrum
                if(x == -1)

                    plot_multipleSpectrums(y,c1);
                else
                    if(y == -1)
                        plot_multipleSpectrums(x,c1);

                    else
                        error('Neither x or y was defined to plot one
Spoectrum');
                    end
                end

            otherwise
                error(['Unkown graphic type ' GraphType '. Use one of the
following:                ''EVM'',''AM-AM'',''AM-PM'',''Error'',''Const'',
''Spectrum'' ']);
        end

    else
        warning('Data to plot not inserted. Figure was created but it is
empty');
    end
```

13. Fiber's Model

Function's name: smf.m

SMF - Linear transfer function model for purely dispersive SMF.

- Inputs:
    - t - time vector[s] (starting at zero with constant step)
    - Ein - input field phasor vector[V/m]

- lambda - central wavelength [nm]
- L - distance [km]
- alpha - attenuation coefficient[dB/km] (default 0.22 dB/km)
- D - dispersion coeffiencient[ps/km/nm] (default 16 ps/km/nm)
- debugPlot - do debug plot {1,0}

- Outputs:
  - Eout - output field phasor
  - f - output frequency vector
  - Sout - output field spectrum

(c) Paulo Ferreirinha

Routine developed for: Radio-Over-Fiber Toolbox

Last Revision: 14-10-2011

Modified by Beatriz Oliveira, 2017

```
function [Eout,f,Sout,phaseFiber] =
smf(t,Ein,lambda,L,alpha,D,debugPlot)

global PARA;

if nargin < 7
    debugPlot = 0;
    if nargin < 6
        D = 16; %[ps/km/nm] typical
        if nargin < 5
            alpha = 0.22;%[dB/km] typical
        end
    end
end


%speed of light
c = 2.99792458e8;%[m/s]

lambda=lambda*10^-9;%[m]
%L=L*1e3; % [m]

%create frequency vector
fs = 1/(t(2)-t(1));
N = length(t);
f = ((0:N-1)*fs/N)-fs/2;%[Hz]     %baseband representation requires
fftshift

%frequency reference
f_ref=0;

D = (1e-6)*D; %[s/m^2]
% beta1 = 1/(c/1.44);          %beta1 (s/m)
beta2 = -D*lambda.^2/(2*pi*c);


%attenuattion coefficient
G = 10.^(-alpha*L/20);
```

```
%Phase shift
phaseFiber=pi*D*(lambda*(f-f_ref)).^2*(L*1e3)/c;


%Fiber Transfer function
Hfiber = G* exp(1i*phaseFiber);

Sin = fftshift(fft(Ein));          %baseband representation requires
fftshift


Hfiber = reshape(Hfiber, size(Sin));
Sout = Sin.*Hfiber;              %
Eout = ifft(fftshift(Sout));     %



return
```

14. Laser's Model

Function's name: lasercw.m

LASERCW - Continuous Wave Laser

The linewidth of the generated CW signal is modeled using a Gaussian white noise source with variance of 2pi*linewidth corresponding to the laser FWHM. The states of polarization are not considered in this model.

- Inputs:
  - t - Time vector (starting at zero with constant step)[s]
  - powerdBm - Average output power [dBm]
  - initPhase - Initial phase [rad]
  - linewidth - FWHM Linewidth (Hz)
  - varargin:
    - deltaFrequency - Frequency [Hz]
    - noiseResetGen - Flag to reset the random number generator{0,1}
    - noiseState - State of random number generator[]

- Outputs:
  - Eout - Output field

- Call-examples:
  - Eout=lasercw(t,powerdbm,initPhase)
  - Eout=lasercw(t,powerdbm,initPhase,linewidth,resetNoiseGen,noiseState)

- References:

[1]Eric Alpman, Florent Munier, Thomas Eriksson, Arne Sevensson, and Herbert Airath. Estimation of phase noise for QPSK modulation over AWGN channels. http://www.ep.liu.se/ecp/008/posters/004/ecp00804p.pdf.

[2]Xiaopei Chen "Ultra-Narrow Laser Linewidth Measurement",PhD Thesis,Virginia Polytechnic Institute and State University,july 2006.

(c) Paulo Ferreirinha

Routine developed for: Radio-Over-Fiber Toolbox

Last Revision: 10-10-2012

```
global PARA

% Sample rate
Ts=t(2)-t(1);
fs=1/Ts;

if nargin <4
    error('The input arguments are not enough')
elseif nargin>7
    error('The input arguments are too many')
end
if nargin>6
    noiseState = varargin{3};
end
if nargin>5
    noiseResetGen = varargin{2};
end
if nargin>4
  deltaFrequency = varargin{1};
end

if nargin <=5
    noiseResetGen=0;
    noiseState=[];
end
if nargin <=4
    noiseResetGen=0;
    noiseState=[];
    deltaFrequency=0;
end



%% add linewidth noise
if linewidth>0

    if noiseResetGen==1
```

```
        rng('shuffle'); % Seed based on the current time. randn returns different values each time you do this.
    end

    % white noise variance = 2*pi*LW

    %% 1st Method

    noiseVariance = 2*pi*linewidth/(fs/2);

    if ~isempty(noiseState)
        rng(noiseState)
    end

    noise = randn(size(t))*sqrt(noiseVariance);

    phaseNoise=cumtrapz(noise);
    phaseNoise(end)=phaseNoise(1);

    %% 2nd  Method
    %     phaseNoise(1)=0;
    %     for i=2:length(t)
    %                                   phaseNoise(i)=phaseNoise(i-1)+randn(1,1)*sqrt(2*pi*linewidth*Ts);
    %     end
    %     phaseNoise(end)=0;
    %===

    %% debug
    %     figure
    %     plot(t/1e-9,phaseNoise/pi)
    %     xlabel('Time[ns]'),ylabel('Normalized phase by \pi')

    FM = exp(1i*phaseNoise);

elseif linewidth==0
    FM = 1;
end

%% create signal
powermW = 10^(powerdBm/10);

if deltaFrequency == 0
    Eout = sqrt(powermW*1e-3).*FM*exp(1i*initPhase);%[V]
else

    Eout                          =                          sqrt(powermW*1e-3).*FM.*exp(1i*2*pi*deltaFrequency*t).*exp(1i*initPhase);%[V]
end


end
```

15. OFDM coder

Function's name: ofdm_coder.m

Generate an OFDM signal in time (series), with the bits present in input prbs

- Inputs:
  - prbs: bits to generate OFDM signal
  - TimeWindow: Time vector, with the instants to be simulated
  - BitRate: Bit Rate, in bit/s
  - BitsPerSymbol: Bits per symbol
  - NumberOfCarriers: Number of OFDM carriers to be used
  - CyclicPrefix: fraction of time that is Cyclic Prefix in each symbol
  - RollofFactor: Roll off factor for the Raised Cosine Filter
  - SampRate: Sample Frequency in Hz
  - ModulationFormat: Format of Modulation for each symbol. Accepts only mQAM or mPSK, M is the number of Symbols = 2^BitsPerSymbol

- Outputs:
  - Tx_bb_samples: Series OFDM signal in time domain
  - Symbol_OFDM_Freq: Parallel OFDM signal

```
  function                                              [Tx_bb_samples,
Symbol_OFDM_Freq]=ofdm_coder(prbs,TimeWindow,BitRate,BitsPerSymbol,Num
berOfCarriers,CyclicPrefix,RolloffFactor,SampRate, ModulationFormat)
  %This program needs more work on: Pilot insertion and removing apply
zero padding
  global PARA
  format long e
  Total_Number_Of_Samples=length(PARA.System.t);
  Nsym = 30;
  N_up=PARA.System.SampRate/(PARA.System.BitRate/PARA.OFDM.BitsPerSymb
ol);
  N_CP=ceil(PARA.OFDM.NumberOfCarriers*PARA.OFDM.CyclicPrefix);
  M=2^PARA.OFDM.BitsPerSymbol;%Number of symbols
  NumberOfOFDMSymbols=floor(length(prbs)/(BitsPerSymbol*ceil(NumberOfC
arriers*(1+CyclicPrefix)))); %Total number of OFDM symbols in the
simulation
  NSamplesFrame=NumberOfCarriers+N_CP;
  NumberSymbolsPerFrame=NumberOfCarriers;
  NumberBitsPerFrame=NumberSymbolsPerFrame*BitsPerSymbol;
  NumberOfFrames=floor(TimeWindow*BitRate/NumberBitsPerFrame);%if
there aren't enough bits to fill the last frame they will be lost
  switch ModulationFormat
          case 'mQAM'
              h   =   modem.qammod('M',   M,   'SymbolOrder',   'Gray',
'InputType', 'Bit');
           case 'mPSK'
              h = modem.pskmod('M', M,'PhaseOffset',pi/2^BitsPerSymbol,
'SymbolOrder', 'Gray', 'InputType', 'Bit');
          otherwise
              disp('Unknown modulation format')
```

```
  end

  % Defining pilot subcarriers
  FFT_Window = NumberOfCarriers;
  Zero_Padding = FFT_Window-NumberOfCarriers;
  % Defining subcarriers with zero padding, adds zeros at the begining
and end
  %Change index
  Data_Subcarriers=[Zero_Padding/2+1:FFT_Window-Zero_Padding/2];
  PilotWord=10+1i*10;
  Pilot_Subcarriers  =  [];%For  example  if  subcarrier  7  is  a  pilot
(Pilot_Subcarriers = [7])
  %Remove data from Pilot Subcarriers

  for i = 1:length(Pilot_Subcarriers)
      i_f = Data_Subcarriers==Pilot_Subcarriers(i);
      Data_Subcarriers(i_f)=[];
  end
  NumberBitsPerFrame=length(Data_Subcarriers)*BitsPerSymbol;
  NumberOfFrames   =   floor(TimeWindow*BitRate/NumberBitsPerFrame);%if
there aren't enough bits to fill the last frame they will be lost
  Symbol_OFDM_Freq = zeros(FFT_Window,NumberOfFrames);
  for i = 1:NumberOfFrames
      Symbol_OFDM_Freq(Data_Subcarriers,i)  =  modulate(h,  prbs((1+(i-
1)*NumberBitsPerFrame):(NumberBitsPerFrame*i))');
      Symbol_OFDM_Freq(Pilot_Subcarriers ,i) = PilotWord;
      OFDM_Symbol_Time(:,i)                                            =
ifft((Symbol_OFDM_Freq(:,i)),FFT_Window);%Converting  Frequency  domain
to time domain
  %Adding the Cyclic Prefix
      OFDM_Symbol_Time_CP(:,i)=[OFDM_Symbol_Time(:,i)'
OFDM_Symbol_Time(1:N_CP,i)']';
  end
  OFDM_Time=reshape(OFDM_Symbol_Time_CP,prod(size(OFDM_Symbol_Time_CP)
),1);
  Tx_bb_samples=OFDM_Time;
  end
```

16. OFDM decoder

Function's name: ofdm_decoder.m

Downsamples and removes cyclic prefix of the received signals. Normalizes to power

- Inputs:
  - bb_received_Id: In-phase photodetected component, downsampled
  - bb_received_Qd: Quadrature photodetected component, downsampled
- Output:
  - OFDM_Rx_Freq: Received OFDM signal, in parallel form

```
function [OFDM_Rx_Freq]=ofdm_decoder(bb_received_Id, bb_received_Qd)
global PARA
Total_Number_Of_Samples=length(PARA.System.t);
NumberOfCarriers=PARA.OFDM.NumberOfCarriers;
N_CP=ceil(NumberOfCarriers*PARA.OFDM.CyclicPrefix);
N_up=PARA.System.N_up;
M=2^PARA.OFDM.BitsPerSymbol;%Number of symbols
NSamplesFrame=NumberOfCarriers+N_CP;
y_ra=bb_received_Id+1i* bb_received_Qd;
y_r_down=downsample(y_ra,N_up);
NFramesRx=floor(length(y_r_down)/NSamplesFrame);
y_r=y_r_down(1:NSamplesFrame*NFramesRx);

y_Rx_paralell=reshape(y_r,NSamplesFrame,NFramesRx);
P_Rx=sum(abs(y_r).^2)/length(y_r);
y_r=y_r/sqrt(P_Rx);


Symbol_OFDM_Freq=zeros(NumberOfCarriers,NFramesRx);
OFDM_Rx_Time=zeros(NumberOfCarriers,NFramesRx);
for i=1:NFramesRx
    OFDM_Rx_Time(:,i)=y_Rx_paralell(1:NumberOfCarriers,i);%Remove
cyclic prefix
    OFDM_Rx_Freq(:,i)=(fft((OFDM_Rx_Time(:,i))));
end

%Normalize the constellation
Pt_Rx=sum(sum(abs(OFDM_Rx_Freq).^2))/(prod(size(OFDM_Rx_Freq)));
OFDM_Rx_Freq=OFDM_Rx_Freq/sqrt(Pt_Rx);
end
```

17. Apply Predistortion

Function's name: apply_predistortion.m

This function applies predistortion and calculates the OFDM IF signal that will modulate the Mach-Zehnder

[xOFDM_bb, y_IF] = apply_predistortion(OFDM_Time) filters OFDM_Time with the Transmitter Raised Cosine filter, upsamples and converts to RF, with frequency IF_OFDM, in global variable PARA.

- Input:
  - OFDM_Time: baseband signal in vector form computed with function ofdm_coder (or predistorted signal computed from this signal)

- Outputs:
  - xOFDM_bb: baseband signal that was only filtered (and upsampled)
  - y_IF: RF signal, with frequency IF_OFDM

```
function [xOFDM_bb, y_IF]=apply_predistortion(OFDM_Time)

global PARA
format long e
Total_Number_Of_Samples=length(PARA.System.t);
Rolloff=PARA.OFDM.RollOff;
Nsym = 30;
N_up=PARA.System.SampRate/(PARA.System.BitRate/PARA.OFDM.BitsPerSymb
ol);
if length(OFDM_Time) <Total_Number_Of_Samples

OFDM_Time(end+1:Total_Number_Of_Samples)=0*[1:Total_Number_Of_Samples-
length(OFDM_Time)];
end

TxFilt  =  comm.RaisedCosineTransmitFilter('OutputSamplesPerSymbol',
N_up,'FilterSpanInSymbols', Nsym, 'RolloffFactor',Rolloff);

y_t=TxFilt(OFDM_Time);

IF_OFDM=PARA.OFDM.IF_OFDM;
t=PARA.System.t';
% fltDelay = Nsym/2*N_up;
% Correct for propagation delay by removing filter transients
% y_t(1: Nsym/2*N_up)=[];
y_t(length(t)+1:end)=[];%Remove the extra time due to the addition of
cyclic prefix
xOFDM_bb=y_t;
RFIcarrierOFDM = cos(2*pi*t*IF_OFDM);%[V]
RFQcarrierOFDM = sin(2*pi*t*IF_OFDM);%[V]
y_IF=real(y_t).*RFIcarrierOFDM+imag(y_t).*RFQcarrierOFDM;

end
```

18. Optical System

Function's name: optical_system_MB.m

Simulates the optical multiband system, where an optical signal is generated in a continuous wave laser (function lasercw). The output of the laser is then modulated to produce two bands: right and left bands.

The right band is modulated with OFDM data (function Intensity_MZM) and the left band is kept pure.

This system has noise with Power Spectral Density given by the global variable PARA.Optical.Noise

- Input:
    - xOFDM_IF: RF electrical signal in vector form computed with function Apply_Predistortion

- Outputs:
    - E_right_mod: Optical signal; output of laser modulated to be the right band of the optical signal; this signal was modulated with OFDM data, using a Mach-Zehnder intensity modulator simulated in functions Intensity_MZM, OCS_MZM or OSSB_MZM; after modulation, this signal is filtered with Super Gaussian Filter to remove Bessel products (function OpticalFilter)
    - Ein_right: Optical signal; output of laser modulated to be the right band of the optical signal; un-modulated; signal used as input of Intensity_MZM, OCS_MZM or OSSB_MZM
    - E_transmitted: Optical signal after coupling E_right_mod with left band of the signal generated with the laser (includes noise)
    - PowerNoise: Power, in Watt, of the noise signal in E_transmitted
    - PowerData: Power, in Watt, of the E_right_mod signal

```
  function                  [E_right_mod,                  Ein_right,
E_transmitted,PowerNoise,PowerData]=optical_system_MB(xOFDM_IF)
  global PARA
  Ein=lasercw(PARA.System.t,10,0,PARA.Optical.LaserLinewidth,1);


  Ein_left=(Ein.*cos(2*pi*PARA.Optical.LO*PARA.System.t)+1i*Ein.*sin(2
*pi*PARA.Optical.LO*PARA.System.t))';
  switch(PARA.Optical.Polarization)
      case 'OSSB' %Optical Single Side Band
          Ein_right  =  (Ein.*cos(2*pi*PARA.Optical.LO*PARA.System.t)-
1i*Ein.*sin(2*pi*PARA.Optical.LO*PARA.System.t))';
          %Ein_right_65=(Ein.*cos(2*pi*(PARA.Optical.LO-
20e9)*PARA.System.t)-1i*Ein.*sin(2*pi*(PARA.Optical.LO-
20e9)*PARA.System.t))';
          %Ein_right=(Ein_right_65);
          % E_right_mod = Intensity_MZM(Ein_right, xOFDM_IF,1,1);
          E_right_mod               =               OSSB_MZM(Ein_right,
xOFDM_IF,PARA.Optical.Vpi,1,PARA.Optical.ExtintionRatio);
          % E_right_mod = OCS_MZM(Ein_right, xOFDM_IF,1,1);

      case 'ODSB' %Optical Dual Side Band (with optical carrier)
          Ein_right  =  (Ein.*cos(2*pi*PARA.Optical.LO*PARA.System.t)-
1i*Ein.*sin(2*pi*PARA.Optical.LO*PARA.System.t))';
          E_right_mod            =            Intensity_MZM(Ein_right,
xOFDM_IF,PARA.Optical.Vpi,1,PARA.Optical.ExtintionRatio);

      case 'OCS' %Optical Supressed Carrier (also Dual Side Band)
          Ein_right  =  (Ein.*cos(2*pi*PARA.Optical.LO*PARA.System.t)-
1i*Ein.*sin(2*pi*PARA.Optical.LO*PARA.System.t))';
          E_right_mod               =                OCS_MZM(Ein_right,
xOFDM_IF,PARA.Optical.Vpi,1,PARA.Optical.ExtintionRatio);


  end
  % %% Filtering
  if(PARA.Optical.SuperGaussian.boolFilter)
      [E_right_mod,timeDiff] = OpticalFilter(E_right_mod);
```

```
else
    timeDiff = 0;
end
PARA.System.t = PARA.System.t(1:(end-abs(timeDiff)));
PARA.Optical.timeDiff = timeDiff;

%% "Coupler" and computing power
Gain = 10^((PARA.Optical.Gain+10)/10);
Pnoise                                                                 =
10*log10((10^((PARA.Optical.Noise)/10)*PARA.System.SampRate));
E_noise=(wgn(length(Ein_left),1,Pnoise))/(sqrt(PARA.System.SampRate)
*2*sqrt(1e-3));
PowerNoise = var(E_noise);
PowerData = var(E_right_mod);


E_transmitted   =   (E_right_mod+Ein_left(1:(end-abs(timeDiff)))   +
E_noise(1:(end-abs(timeDiff))));


end
```

19. Dispersion compensation

Function's name: dispersion_compensation.m

Compensate the distortions introduced by the fiber.

- Inputs:
  - X_Tx: Constellation of the transmitted symbols (ideal points of the Constellation)
  - X_Rx: Constellation of the received symbols

- Outputs:
  - Disp_Comp: Compensation factor for each frequency
  - X_Tx_n: Normalized constellation of the transmitted symbols
  - X_Rx_n: Normalized constellation of the received symbols

```
function
[Disp_Comp,X_Tx_n,X_Rx_n]=dispersion_compensation(X_Tx,X_Rx)
global PARA
NumberOfCarriers=PARA.OFDM.NumberOfCarriers;
%Make sure the constellations are normalized and the number of points
of the Tx and received is the same
aa=size(X_Rx);

N_Frames_Rx=aa(1,2);
 X_Tx(:,N_Frames_Rx+1:end)=[];
 Pt_Tx=sum(sum(abs(X_Tx).^2))/(prod(size(X_Tx)));
X_Tx=X_Tx/sqrt(Pt_Tx);
 Pt_Rx=sum(sum(abs(X_Rx).^2))/(prod(size(X_Rx)));
X_Rx=X_Rx/sqrt(Pt_Rx);
for i=1:NumberOfCarriers
    Phase_diff(i,:)=(X_Tx(i,:)./X_Rx(i,:));
```

```
end;
Disp_Comp=mean(Phase_diff');
X_Tx_n=X_Tx;
X_Rx_n=X_Rx;
```

20. EVM calculation

Function's name: evm_calculation.m

Calculate the EVM, in percentage.

- Inputs:
  - X_Tx: Constellation of the transmitted symbols (ideal points of the Constellation)
  - X_Rx: Constellation of the received symbols

- Outputs:
  - EVM: Error Vector Magnitude, in percentage.
  - X_Tx_n: Normalized constellation of the transmitted symbols
  - X_Rx_n: Normalized constellation of the received symbols

```
function [EVM,X_Tx_n,X_Rx_n]=evm_calculation(X_Tx,X_Rx)
global PARA
NumberOfCarriers=PARA.OFDM.NumberOfCarriers;
%Make sure the constellations are normalized and the number of points
of the Tx and Rx is the same
aa=size(X_Rx);

N_Frames_Rx=aa(1,2);
 X_Tx(:,N_Frames_Rx+1:end)=[];
 Pt_Tx=sum(sum(abs(X_Tx).^2))/(prod(size(X_Tx)));
X_Tx=X_Tx/sqrt(Pt_Tx);
 Pt_Rx=sum(sum(abs(X_Rx).^2))/(prod(size(X_Rx)));
X_Rx=X_Rx/sqrt(Pt_Rx);
for i=1:NumberOfCarriers
    e(i,:)=(real(X_Rx(i,:))-real(X_Tx(i,:))).^2+(imag(X_Rx(i,:))-
imag(X_Tx(i,:))).^2;
    e_mean(i)=(sum(e(i,:)));
end;
X_Tx_n=X_Tx;
X_Rx_n=X_Rx;
EVM=sqrt(sum(e_mean)./(prod(size(X_Rx))))*100;
```

21. Optical Detection – Coherent Detection

Function's name: OpticalDD_Electrical_coherent.m

Simulates Coherent Detection of the optical signal, converts to baseband, applies Receiver Raised Cosine filter and downsamples. By converting to baseband, separates signal in in-phase and quadrature components

- Inputs:
  - Ein: Optical signal, generated in function OPTICAL_SYSTEM_MB

- Outputs:
  - bb_received_I: In-phase photodetected component
  - bb_received_Q: Quadrature photodetected component
  - bb_received_Id: In-phase photodetected component, downsampled
  - bb_received_Qd: Quadrature photodetected component, downsampled

```
function          [bb_received_I,          bb_received_Q,bb_received_Id,
bb_received_Qd,                                                       I_PD
,Rx_bbI,Rx_bbQ]=OpticalDD_Electrical_coherent(E_in)
global PARA
N_up=PARA.System.N_up;
Nsym=PARA.OFDM.FilterSpanInSymbols;
RollOff=PARA.OFDM.RollOff;
%F_Local=PARA.OFDM.IF_OFDM+2*PARA.Optical.LO;
F_Local=PARA.OFDM.IF_Rx;

t=PARA.System.t';

%RFLocalCarrierI=cos(2*pi*t*F_Local-pi);%with 60 GHz
%RFLocalCarrierQ=sin(2*pi*t*F_Local);
RFLocalCarrierI=+cos(2*pi*t*F_Local-pi);%with  70  GHz  or  using  the
upper
% sideband
RFLocalCarrierQ=-sin(2*pi*t*F_Local);


%Photodetection
I_PD=E_in.*conj(E_in);
I_PD=I_PD-mean(I_PD); %Remove DC;

 % Filter I_PD for the components we need
if(PARA.Electrical.SuperGaussian.boolFilter)
    [I_PD,timeDiff] = ElectricalFilter(I_PD);
else
    timeDiff = 0;
end


%Baseband Conversion
Rx_bbI=I_PD.*RFLocalCarrierI;
Rx_bbQ=I_PD.*RFLocalCarrierQ;

%Raised cosine filtering
```

```
  RxFilt    =    comm.RaisedCosineReceiveFilter('InputSamplesPerSymbol',
N_up,'FilterSpanInSymbols',                                         Nsym,
'RolloffFactor',RollOff,'DecimationFactor',1);
  bb_received_I=RxFilt(Rx_bbI);
  bb_received_I  =  bb_received_I(Nsym*N_up+1:end);%Remove  the  extra
samples introduced by the transmitter filter and receiver filter
  bb_received_Q = RxFilt(Rx_bbQ);
  bb_received_Q = bb_received_Q(Nsym*N_up+1:end);
  bb_received_Id=downsample(bb_received_I,N_up);
  bb_received_Qd=downsample(bb_received_Q,N_up);
  end
```

22. Optical Detection – Envelope Detection

Function's name: OpticalDD_Electrical_ED.m

Simulates Envelope Detection of the optical signal, converts to baseband, applies Receiver Raised Cosine filter and downsamples. By converting to baseband, separates signal in in-phase and quadrature components

- Inputs:
  - Ein: Optical signal, generated in function OPTICAL_SYSTEM_MB

- Outputs:
  - bb_received_I: In-phase photodetected component
  - bb_received_Q: Quadrature photodetected component
  - bb_received_Id: In-phase photodetected component, downsampled
  - bb_received_Qd: Quadrature photodetected component, downsampled

```
  function        [bb_received_I,        bb_received_Q,bb_received_Id,
bb_received_Qd, I_PD_ED ,Rx_bbI,Rx_bbQ]=OpticalDD_Electrical_ED(E_in)
  global PARA
  N_up = PARA.System.N_up;
  Nsym = PARA.OFDM.FilterSpanInSymbols;
  RollOff = PARA.OFDM.RollOff;
  %F_Local=PARA.OFDM.IF_OFDM+2*PARA.Optical.LO;
  F_Local = PARA.OFDM.IF_OFDM; %fIF for Envelope Detection

  t=PARA.System.t';


  %RFLocalCarrierI=cos(2*pi*t*F_Local-pi);%with 60 GHz
  %RFLocalCarrierQ=sin(2*pi*t*F_Local);
  RFLocalCarrierI=+cos(2*pi*t*F_Local-pi);%with  70  GHz  or  using  the
upper
  % sideband
  RFLocalCarrierQ=-sin(2*pi*t*F_Local);


  %Photodetection
  I_PD=E_in.*conj(E_in);
```

```
I_PD=I_PD-mean(I_PD); %Remove DC;

% Filter I_PD for the components we need
if(PARA.Electrical.SuperGaussian.boolFilter)
    [I_PD,timeDiff] = ElectricalFilter(I_PD);
else
    timeDiff = 0;
end

%% Envelope detection

I_PD_ED = abs(I_PD).^2;

%% Coherent detection
%Baseband Conversion
Rx_bbI=I_PD_ED.*RFLocalCarrierI;
Rx_bbQ=I_PD_ED.*RFLocalCarrierQ;

%Raised cosine filtering
RxFilt = comm.RaisedCosineReceiveFilter('InputSamplesPerSymbol', N_up,'FilterSpanInSymbols', Nsym, 'RolloffFactor',RollOff,'DecimationFactor',1);
bb_received_I=RxFilt(Rx_bbI);
bb_received_I = bb_received_I(Nsym*N_up+1:end);%Remove the extra samples introduced by the transmitter filter and receiver filter
bb_received_Q = RxFilt(Rx_bbQ);
bb_received_Q = bb_received_Q(Nsym*N_up+1:end);
bb_received_Id=downsample(bb_received_I,N_up);
bb_received_Qd=downsample(bb_received_Q,N_up);
```

23. System Parameters

Script's name: SystemParameters.m

Script that loads the system's parameters. These can be changed after loading.

According to the modulation formats being used (OCS, OSSB, ODSB) the filter's parameters must be adjusted.

```
%% System Parameters
% Physics Constants, Modulation parameters, debugging variables, Simulation
% parameters (time, frequency), etc
global PARA;
PARA.System.debug_bool = debug_bool;
PARA.Constants.c0          = 2.99792458e8;     %Light speed in vacuum [m/s]
PARA.System.BitRate         = 5e9;               %Bit rate[bit/s]
PARA.OFDM.FilterSpanInSymbols=30; %
PARA.OFDM.BitsPerSymbol = 4;        %NBits per symbol
PARA.OFDM.NumberOfCarriers=32;     %Number of OFDM Carriers
PARA.OFDM.CyclicPrefix=0.125;      %OFDM cyclic prefix
```

```
  PARA.OFDM.RollOff=0.25  ;          %Raised cosine roll off factor
  PARA.OFDM.ModulationFormat='mQAM';
  PARA.System.SampRate                  =  64*PARA.System.BitRate;%Sampling
frequency [Hz]

  PARA.Optical.LO=27.5e9;
  PARA.Optical.LaserLinewidth=0.00e3;
  PARA.OFDM.NumberOfFrames=80;
  PARA.OFDM.IF_OFDM=5e9;
  PARA.OFDM.IF_Rx=60e9;%Receiver local oscillator
  PARA.System.TimeWindow=PARA.OFDM.NumberOfFrames*(PARA.OFDM.NumberOfC
arriers+ceil(PARA.OFDM.NumberOfCarriers*PARA.OFDM.CyclicPrefix))/(PARA
.System.BitRate/ PARA.OFDM.BitsPerSymbol);
  PARA.System.N_up=
(PARA.System.SampRate/PARA.System.BitRate)*PARA.OFDM.BitsPerSymbol;
  TimeWindow=PARA.System.TimeWindow;
  SampRate=PARA.System.SampRate;
  BitRate=PARA.System.BitRate;
  BitsPerSymbol=PARA.OFDM.BitsPerSymbol;
  NumberOfCarriers=PARA.OFDM.NumberOfCarriers;
  CyclicPrefix=PARA.OFDM.CyclicPrefix;
  ModulationFormat=PARA.OFDM.ModulationFormat;
  RolloffFactor=PARA.OFDM.RollOff;
  PARA.System.t = 0:(1/SampRate):TimeWindow; %time vector [s]
  PARA.Optical.centerFreq = 193.1e12; %[Hz]Reference optical frequency/
Central optical frequency (Frequency from ITU grid)
  PARA.Opical.centerLamb  = PARA.Constants.c0 /PARA.Optical.centerFreq;
%[m] % Reference optical wavelength/ Optical wavelength
  PARA.Optical.f   =  PARA.Optical.centerFreq+(0:length(PARA.System.t)-
1)*PARA.System.SampRate/length(PARA.System.t)-PARA.System.SampRate/2;%
Create optical frequency vector %[Hz]
  PARA.Optical.Gain=20;%Optical Amplifier Gain
  PARA.Optical.Noise=-90;% Power spectral density of the added optical
noise
  PARA.Optical.Vpi  = 3.5;%Voltage needed to cause a phaseshift of pi
between the two arms of the MZM
  PARA.Optical.ExtintionRatio = 0.5;%Splitting of the input field. If
this value is 0.5, then we have an ideal Mach-Zehnder
  % PARA.Optical.ExtintionRatio = 0.47;%Splitting of the input field.
If this value is 0.5, then we have an ideal Mach-Zehnder
  %% Setup spectrum analysis
  % Parameters needed to plot Spectrum of non-deterministic signals
  Npoints=32*1024;
  hPsd=spectrum.welch('Hamming',Npoints);
  hopts=psdopts(hPsd);
  set(hopts,'SpectrumType','twosided','NFFT',Npoints,'Fs',PARA.System.
SampRate, 'CenterDC',true);

  %% Optical Filter Parameters
  % Parameters for the Optical Super Gaussian filter, including Boolean
  % variable
  %Defined in baseband equivalent, which means that we are seeing only
the
  %positive frequencies

  PARA.Optical.SuperGaussian.Fc = 30e9; %Hz
  PARA.Optical.SuperGaussian.Gain = 5;
```

```
  PARA.Optical.SuperGaussian.FilterOrder = 8;
  PARA.Optical.SuperGaussian.Bw = 15e9; %Hz
  PARA.Optical.SuperGaussian.SbA = 50; %dB
  PARA.Optical.SuperGaussian.boolFilter = 0; %if we want to apply the
filter

  %% Electrical Filter Parameters
  % Parameters for the Electrical Super Gaussian filter, including
boolean
  % variable
  % Defined in actual baseband, which means that we are seeing positive
and
  % negative frequencies; The filter is symmetrical to the x-axis
  PARA.Electrical.SuperGaussian.Fc = 57.5e9; %Hz
  PARA.Electrical.SuperGaussian.Gain = 5;
  PARA.Electrical.SuperGaussian.FilterOrder = 8;
  PARA.Electrical.SuperGaussian.Bw = 10e9; %Hz
  PARA.Electrical.SuperGaussian.SbA = 50; %dB
  PARA.Electrical.SuperGaussian.boolFilter = 1; %if we want to apply the
filter


  %% Fiber Parameters
  %Parameters of the Fiber model

  PARA.Fiber.FiberLength = 10; % [km]
  PARA.Fiber.FiberLambda   =   PARA.Opical.centerLamb*1e9;   %   central
wavelength [nm]
  %Dispertion and attenuation according to Wavelength
  PARA.Fiber.FiberDispersion = 16; % dispersion coeffiencient [ps/km/nm]
  PARA.Fiber.FiberAlpha = 0.22; %0.22 alpha - attenuation coefficient
[dB/km]

  %% Graph Parameters
  % Parameters that help to keep the graphics with the same dimensions
  PageWidth     = 20; % [cm]
  GraphHeight   = 15; % [cm]
  GraphFontSize = 16; % pts

  PARA.Graph.PageWidth = PageWidth;
  PARA.Graph.GraphHeight = GraphHeight;
  PARA.Graph.GraphFontSize = GraphFontSize;
```

24. System Total – Coherent Detection

Function's name: SystemTotal.m

Simulation of the total system, without and with predistortion, using Coherent Detector.

```
  function [OSNR_noPD,OSNR_PD,RF_power,...

evm_nodispersioncomp,evm_dispersioncomp_pd,evm_dispersioncomp_pd_2]...
      = SystemTotal(Polarization,Amplitude_Factor,FiberTransmission)
  format longe
```

```
%% Parameters
Kinv = 4;
Qinv = 1;
odd_order = 1;
debug_bool = 1; %if 0, doesn't plot constelations, etc
fiber_akq_bool = 1; %if 0, akq are estimated without the fiber
transmission
if(exist('FiberTransmission','var'))
    %if 0, fiber is never used (system where RH is directly connected
to PU)
    fiber_bool = FiberTransmission;
else
    fiber_bool = 1;
end
%Load system parameters
SystemParameters

%If necessary modify the loaded parameters here:


switch(Polarization)
    case 'OSSB'
        PARA.Optical.SuperGaussian.boolFilter = 1;   %if we want to
apply the filter
        PARA.Optical.SuperGaussian.Fc = 30e9;        %[Hz]
        PARA.Optical.SuperGaussian.Bw = 10e9;        %[Hz]

        PARA.Electrical.SuperGaussian.boolFilter = 1;    %if we want
to apply the filter
        PARA.Electrical.SuperGaussian.Fc = 59e9;        %[Hz]
        PARA.Electrical.SuperGaussian.Bw = 10e9;        %[Hz]

    case 'ODSB'
        PARA.Optical.SuperGaussian.boolFilter = 1;   %if we want to
apply the filter
        PARA.Optical.SuperGaussian.Fc = 27.5e9;      %[Hz]
        PARA.Optical.SuperGaussian.Bw = 15e9;        %[Hz]

        PARA.Electrical.SuperGaussian.boolFilter = 0;


    case 'OCS'
        PARA.Optical.SuperGaussian.boolFilter = 1;
        PARA.Optical.SuperGaussian.Fc = 27.5e9;      %[Hz]
        PARA.Optical.SuperGaussian.Bw = 15e9;        %[Hz]

        %Coherent Detection won't need Electrical Filtering
        PARA.Electrical.SuperGaussian.boolFilter = 0;

    otherwise
        error('Invalid Polarization. Using OSSB');
        Polarization = 'OSSB';
end
PARA.Optical.Polarization = Polarization;


if(~fiber_bool && fiber_akq_bool)
```

```
  warning('Parameters do not match: no fiber transmission but variable
set to use fiber transmission when estimating predistorter
coefficients');
  warning('Changing parameters: no fiber transmission to estimate the
coefficients');
  fiber_akq_bool = 0;
  end

  %% Compute Predistorter
  % Run the nonlinear system as is and compute the predistorter
coefficients
  % - Predistorter not apllied

  if(debug_bool)
     disp('Starting nonlinear system ... ');
  end

  % TRANSMITTER
  if(debug_bool)
     disp('   > Transmitter: ');
  end
  rng('shuffle');
  prbs=randi([0 1],1,ceil(TimeWindow*BitRate)); %generate bits
  [Tx_bb_samples,
X_Tx]=ofdm_coder(prbs,TimeWindow,BitRate,BitsPerSymbol,NumberOfCarrier
s,CyclicPrefix,RolloffFactor,SampRate,ModulationFormat);
  %If we wanted to apply predistortion, it would be here. We are not
  %applying, so the input of APPLY_PREDISTORTION is the output of
OFDM_CODER
  [xOFDM_bb, xOFDM_IF]=apply_predistortion(Tx_bb_samples); %The signal
is not predistorted here
  [Ein_right_mod, Ein_right, E_transmitted,PowerNoise,PowerData] =
optical_system_MB(Amplitude_Factor*xOFDM_IF);

  %Get OSNR: Optical Signal-to-Noise Ratio (without predistortion)
  OSNR_noPD = PowerData/PowerNoise;

  %Get RF Power: Power of the OFDM signal after RF conversion (it will
be the
  % x-axis in some graphics)
  RF_power = var(Amplitude_Factor*xOFDM_IF);

  if(debug_bool)
     disp(['       OSNR: ' num2str(10*log10(OSNR_noPD)) ' dB']);
     disp(['          RF Power: ' num2str(10*log10(RF_power/1e-3)) '
dBm']);
  end

  % FIBER LINK
  % Only transmit through fiber when fiber_bool is set to 1

  %if we are in the system without fiber, then fiber_akq_bool is sure
to be
  %zero (previously verified)
  if(fiber_akq_bool)
     %fiber model: output signal will have distortion
     if(debug_bool)
```

```
        disp(['      > Transmitting through fiber of length ' num2str(PARA.Fiber.FiberLength) '...']);
      end
      [E_transmitted_fiber,f,Sout,phaseFiber_debug] = smf(PARA.System.t,E_transmitted,PARA.Fiber.FiberLambda,PARA.Fiber.FiberLength,PARA.Fiber.FiberAlpha,PARA.Fiber.FiberDispersion,1);
  else
      %no fiber applied: signals don't have distortion
      if(debug_bool)
          disp('   > This system has no fiber transmission.')
      end
      E_transmitted_fiber  = E_transmitted;
  end


  %Plot Fiber ouput only if debug_bool is set to 1
  if(debug_bool)
      %Plot Optical Spectrum
      signal=E_transmitted*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
      PSD1=psd(hPsd,signal,hopts);
      data_noPD=dspdata.psd([PSD1.Data], PSD1.Frequencies,'Fs',PARA.System.SampRate);
      hOptSpect = CreateFigure('Spectrum',data_noPD,-1,2,PageWidth,GraphHeight,-1,GraphFontSize);
  %     title('Spectrum without equalization, Optical Domain')
  %   set(findall(hOptSpect,'-property','FontSize'),'FontSize',GraphFontSize);
      title('        ')
      set(findall(hOptSpect,'-property','FontSize'),'FontSize',GraphFontSize);
      if(PARA.System.BitRate == 40e9)
          xlim([-0.032 0.037]);
      else
          xlim([-32 37]);
      end

      grid off;
  end

  %RECEIVER
  if(debug_bool)
      disp('   > Receiver:')
  end
  [bb_received_I, bb_received_Q, bb_received_Id, bb_received_Qd, I_PD_noPD,Rx_bbI,Rx_bbQ] = OpticalDD_Electrical_coherent(E_transmitted_fiber);
  [X_Rx_noPD]=ofdm_decoder(bb_received_I, bb_received_Q);

  [Disp_Comp_nopd,X_Tx_n,X_Rx_n] = dispersion_compensation(X_Tx,X_Rx_noPD);
  X_Rx_noPD_comp=X_Rx_n.*Disp_Comp_nopd';   %Apply dispersion compensation

  [evm_nodispersioncomp,X_Tx_n,X_Rx_n] = evm_calculation(X_Tx,X_Rx_noPD_comp);
```

```
  %Plot these graphics/Display text  only if debug_bool is set to 1
  if(debug_bool)
      %Plot Electric Spectrum
      signal=I_PD_noPD*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
      PSD1=psd(hPsd,signal,hopts);
      data_noPD_el=dspdata.psd([PSD1.Data],
PSD1.Frequencies,'Fs',PARA.System.SampRate);
      hElecSpect        =        CreateFigure('Spectrum',data_noPD_el,-
1,2,PageWidth,GraphHeight,-1,GraphFontSize);
      title('Spectrum without equalization, Electrical Domain')
      set(findall(hElecSpect,'-
property','FontSize'),'FontSize',GraphFontSize);


      %Plot Constellation
      hConst_noPD                                                    =
CreateFigure('Const',real(X_Rx_noPD),imag(X_Rx_noPD),1,PageWidth,Graph
Height);
      title('Constellation of non-linear system')
      xlabel('Real');ylabel('Imaginary');
      set(findall(hConst_noPD,'-
property','FontSize'),'FontSize',GraphFontSize);


      % Display EVM
      disp(['                      EVM   of   non-linear   system:   '
num2str(evm_nodispersioncomp) ' %']);
  end


  %Prepare to compute coefficients:
  Rx_bb_samples = bb_received_Id+1i*bb_received_Qd;  %Received Samples;
  Tx_bb_samples(length(Rx_bb_samples)+1:end)=[];           %Transmitted
Samples;

  %Renaming for clarity
  xit = Tx_bb_samples;
  yit = Rx_bb_samples;

  %"Saving" variables (we need un-predistorted signals for AM-AM plot)
  xit_noPD = xit;
  yit_noPD = yit;

  yit_noPD = yit_noPD/max(abs(yit_noPD));
  xit_noPD = xit_noPD/max(abs(xit_noPD));




  xit(1)=[];
  yit(1)=[];

  N = length(xit);

  %Making sure we have column-vectors
  xit = reshape(xit,N,1);
  yit = reshape(yit,N,1);
```

```
%Normalizing
Gx = max(abs(xit));
xit = xit./Gx;
yit = yit./max(abs(yit));



%Computing Memory vectors:
% Xinv is needed to compute predistortion signal
% Yinv is needed to compute coefficients for predistorter
Xinv = MemoryVector(xit,Kinv,Qinv,odd_order);
Yinv = MemoryVector(yit,Kinv,Qinv,odd_order);

%Compute Coefficients: Minimum Squares Method
a_kq=pinv(Yinv)*xit;%Compute the coeficients of the predistorter




%% System with distortion compensation and pre-distorter
% Run the non-linear system with predistortion, compute dispersion
% compensation factor and apply it
% - Predistortion applied
% - dispersion compensation factor computed
% - dispersion compensation applied


if(debug_bool)
    disp('Starting system with predistortion... ');
end

% TRANSMITTER
if(debug_bool)
    disp('    > Transmitter: ');
end
rng('shuffle');
prbs=randi([0 1],1,ceil(TimeWindow*BitRate)); %Generating bits
[Tx_bb_samples,
X_Tx]=ofdm_coder(prbs,TimeWindow,BitRate,BitsPerSymbol,NumberOfCarrier
s,CyclicPrefix,RolloffFactor,SampRate,ModulationFormat);

if(debug_bool)
    disp('    Applying predistortion... ');
end
%Transmitter - prepare to apply predistortion
xit = Tx_bb_samples;
xit = reshape(xit,length(xit),1);%making sure its a column-vector

%Normalizing
Gx=max(abs(xit));
xit=xit./Gx;

X = MemoryVector(xit,Kinv,Qinv,odd_order);
xdist_it=Gx*X*a_kq; %This iteration's predistortion signal (a_kq was
previously computed)
```

```
    %Applying predistortion
    [xOFDM_bb, xOFDM_IF] = apply_predistortion(xdist_it); %The signal is predistorted here

    %Transmitter - continued
    [Ein_right_mod,                                    Ein_right, E_transmitted,PowerNoise,PowerData]=optical_system_MB(Amplitude_Factor*xOFDM_IF);


    %Get OSNR: Optical Signal-to-Noise Ratio (with predistortion)
    OSNR_PD = PowerData/PowerNoise;
    if(debug_bool)
        disp(['        OSNR: ' num2str(10*log10(OSNR_PD)) 'dB']);
    end

    % FIBER LINK
    if(fiber_bool)
        % Here, we always want to transmit through fiber (in the system with fiber) because we will compute
        % the dispersion compensation factor
        if(debug_bool)
            disp(['       >  Transmitting  through  fiber  of  length  ' num2str(PARA.Fiber.FiberLength) '...']);
        end

        [E_transmitted_fiber,f,Sout]                               = smf(PARA.System.t,E_transmitted,PARA.Fiber.FiberLambda,PARA.Fiber.FiberLength,PARA.Fiber.FiberAlpha,PARA.Fiber.FiberDispersion,1);
    else
        %In the system without fiber (RH conected to PU):
        E_transmitted_fiber = E_transmitted;
    end

    % RECEIVER
    if(debug_bool)
        disp('    > Receiver: ');
    end
    [bb_received_I,   bb_received_Q,   bb_received_Id,   bb_received_Qd, I_PD_PD ] = OpticalDD_Electrical_coherent(E_transmitted_fiber);
    [X_Rx_PD]=ofdm_decoder(bb_received_I, bb_received_Q);


    %Prepare to plot AM-AM
    Rx_bb_samples=bb_received_Id+1i*bb_received_Qd;%Received Samples;
    Tx_bb_samples(length(Rx_bb_samples)+1:end) = [];%Transmitted Samples;
    xdist_it(length(Rx_bb_samples)+1:end) = [];

    %"Saving" variables (we also need predistorted signals for AM-AM plot)
    yit_PD = Rx_bb_samples/max(abs(Rx_bb_samples));
    xit_PD = Tx_bb_samples/max(abs(Tx_bb_samples));
    xdist_it = xdist_it/max(abs(xdist_it));


    %Plot only if debug_bool is set to 1
    if(debug_bool)
```

```
        %Amplitude: AM-AM plot
        hAMAM                      =                    CreateFigure('AM-
AM',abs(xit_noPD),abs(yit_noPD),1,PageWidth,GraphHeight);
        figure(hAMAM); hold on;
        title('AM/AM of Predistorter');
        CreateFigure('AM-
AM',abs(xit_PD),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM);
        CreateFigure('AM-
AM',abs(xit_PD),abs(xdist_it),1,PageWidth,GraphHeight,hAMAM);
    %      plot(abs(xit),abs(xit),'k--');
        legend('Original   input   vs.   Original   Output','Input   vs.
Predistorted Output','Input vs. Predistortion signal ','y=x');
        grid minor;
        set(findall(hAMAM,'-
property','FontSize'),'FontSize',GraphFontSize);


         %Phase: AM-PM plot
        thetax_PD = angle(xit_PD);
        idx = thetax_PD < 0;
        thetax_PD(idx) = thetax_PD(idx)+2*pi;

        thetax_noPD = angle(xit_noPD);
        idx = thetax_noPD < 0;
        thetax_noPD(idx) = thetax_noPD(idx)+2*pi;

        theta = angle(yit_noPD);
        theta = theta - thetax_noPD;
        theta = rem(theta,pi);



        thetaap = angle(yit_PD);
        idx = thetaap < 0;
        thetaap(idx) = thetaap(idx)+2*pi;
        thetaap = thetaap - thetax_PD;
        thetaap = rem(thetaap,pi);

        thetaz = angle(xdist_it);
        idx = thetaz < 0;
        thetaz(idx) = thetaz(idx)+2*pi;
        thetaz = thetaz - thetax_PD;
        thetaz = rem(thetaz,pi);

        hAMPM                      =                    CreateFigure('AM-
PM',abs(xit_noPD),theta/pi,1,PageWidth,GraphHeight);
        figure(hAMPM)
        title('AM/PM of Predistorter');
        hold on;
        CreateFigure('AM-
PM',abs(xit_PD),thetaap/pi,1,PageWidth,GraphHeight,hAMPM);
        CreateFigure('AM-
PM',abs(xit_PD),thetaz/pi,1,PageWidth,GraphHeight,hAMPM);
        xlabel('Absolute Value In'); ylabel('Phase (\pi radians)');
```

```
        legend('Original   input   vs.   Original   Output','Input   vs.
Predistorted Output','Input vs Predistortion signal');
        set(findall(hAMPM,'-
property','FontSize'),'FontSize',GraphFontSize);
        grid minor;



    end


    %Receiver - Distortion compensation
    % Even if the fiber is not used, dispersion compensation should be
used because OSSB introduces a phase shift sue to the use of the Hilbert
Transform when modulating the Mach-Zehnder
    % If we are not using OSSB, then there is no dispersion nor phase
shift being introduced and this step does no harm
    [Disp_Comp_pd,X_Tx_n,X_Rx_n] = dispersion_compensation(X_Tx,X_Rx_PD);
    X_Rx_dist_comp=X_Rx_n.*Disp_Comp_pd'; %Apply dispersion compensation

    [evm_dispersioncomp_pd,X_Tx_n,X_Rx_n]                                =
evm_calculation(X_Tx,X_Rx_dist_comp);

    if(debug_bool)

        %Plot Optical Spectrum
        signal=E_transmitted*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
        PSD1=psd(hPsd,signal,hopts);
        data_PD=dspdata.psd([PSD1.Data],
PSD1.Frequencies,'Fs',PARA.System.SampRate);
        hOptSpect          =          CreateFigure('Spectrum',data_PD,-
1,2,PageWidth,GraphHeight,-1,GraphFontSize);
         title('Spectrum with equalization, Optical Domain')
        set(findall(hOptSpect,'-
property','FontSize'),'FontSize',GraphFontSize);

        %Plot Electric Spectrum
        signal=I_PD_PD*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
        PSD1=psd(hPsd,signal,hopts);
        data_PD_el=dspdata.psd([PSD1.Data],
PSD1.Frequencies,'Fs',PARA.System.SampRate);
        hElecSpect         =        CreateFigure('Spectrum',data_PD_el,-
1,2,PageWidth,GraphHeight,-1,GraphFontSize);
        title('Spectrum with equalization, Electrical Domain')
        set(findall(hElecSpect,'-
property','FontSize'),'FontSize',GraphFontSize);


        % Plotting both spectrums
        hSpec                                                            =
CreateFigure('Spectrum2',data_noPD,data_PD,1,PARA.Graph.PageWidth,PARA
.Graph.GraphHeight);
        legend('Without Predistortion', 'With Predistortion');
        title('Optical Spectrums');
        xlim([20 35]);
        grid off
```

```
      set(findall(hSpec,'-
property','FontSize'),'FontSize',PARA.Graph.GraphFontSize);

       % Plotting both spectrums
      hSpec                                                            =
CreateFigure('Spectrum2',data_noPD_el,data_PD_el,1,PARA.Graph.PageWidt
h,PARA.Graph.GraphHeight);
      legend('Without Predistortion', 'With Predistortion');
      if(FiberTransmission)
      title(['Electrical     Spectrums,      L_f_i_b_e_r      =      '
num2str(PARA.Fiber.FiberLength) ' Km']);
      else
          title('Electrical Spectrums, Without Fiber Transmission');
      end
        set(findall(hSpec,'-
property','FontSize'),'FontSize',PARA.Graph.GraphFontSize);


      %Plot Constellation
      hConst_PD                                                        =
CreateFigure('Const',real(X_Rx_dist_comp),imag(X_Rx_dist_comp),1,PageW
idth,GraphHeight);
      title('Constellation    with    dispersion    compensation    and
predistorter')
      set(findall(hConst_PD,'-
property','FontSize'),'FontSize',GraphFontSize);



      % Display EVM
      disp(['         EVM 1st  run:  '  num2str(evm_dispersioncomp_pd)  '
%']);


      %Plot BOTH Constellation
      L1 = 'Without PD'; L2 = 'With PD';
      hConst2                                                          =
CreateFigure('Const2',X_Rx_noPD,X_Rx_dist_comp,1,PageWidth,GraphHeight
,-1,GraphFontSize,L1,L2);
      title('Received Constellations')
      set(findall(hConst2,'-
property','FontSize'),'FontSize',GraphFontSize);

  end

  %% 2nd run: System with distortion compensation and pre-distorter
  % Run system with predistortion and with dispersion compensation
  % - Predistortion applied
  % - dispersion compensation factor not computed
  % - dispersion compensation applied

  if(debug_bool)
      disp(['Starting 2nd run: system with predistortion and dispersion
'...
      'compensation  with  the  previously  computed  coefficients  and
dispersion'...
      ' compensation factor... ']);
```

```
  end

  % TRANSMITTER
  if(debug_bool)
      disp('    > Transmitter: ');
  end
  rng('shuffle');
  prbs=randi([0 1],1,ceil(TimeWindow*BitRate)); %Generating bits
  [Tx_bb_samples,
X_Tx]=ofdm_coder(prbs,TimeWindow,BitRate,BitsPerSymbol,NumberOfCarrier
s,CyclicPrefix,RolloffFactor,SampRate,ModulationFormat);

  %Transmitter - prepare to apply predistortion
  if(debug_bool)
      disp('    Applying predistortion...');
  end
  xit=Tx_bb_samples;
  xit=reshape(xit,length(xit),1);

  %Normalizing
  Gx=max(abs(xit));
  xit=xit./Gx;
  X = MemoryVector(xit,Kinv,Qinv,odd_order);
  xdist_it=Gx*X*a_kq; %This iteration's predistortion signal (a_kq was
previously computed)


  %Applying predistortion
  [xOFDM_bb, xOFDM_IF] = apply_predistortion(xdist_it); %The signal is
predistorted here
  [Ein_right_mod,                                            Ein_right,
E_transmitted,PowerNoise,PowerData]=optical_system_MB(Amplitude_Factor
*xOFDM_IF);



  % FIBER LINK
  if(fiber_bool)
      %System with Fiber (RH not directly conected to PU)
      % We always want to transmit through fiber here because we will
apply
      % distortion compensation
      if(debug_bool)
          disp(['       >  Transmitting  through  fiber  of  length  '
num2str(PARA.Fiber.FiberLength) '...']);
      end

      [E_transmitted_fiber,f,Sout]                                    =
smf(PARA.System.t,E_transmitted,PARA.Fiber.FiberLambda,PARA.Fiber.Fibe
rLength,PARA.Fiber.FiberAlpha,PARA.Fiber.FiberDispersion,1);
  else
      %In the system without fiber (RH conected to PU):
      E_transmitted_fiber = E_transmitted;
  end


  %RECEIVER
```

```
  if(debug_bool)
      disp('    > Receiver: ');
  end
  [bb_received_I,    bb_received_Q,    bb_received_Id,    bb_received_Qd,
I_PD_PD ] = OpticalDD_Electrical_coherent(E_transmitted_fiber);
  [X_Rx_PD2]=ofdm_decoder(bb_received_I, bb_received_Q);

  Rx_bb_samples=bb_received_Id+1i*bb_received_Qd;%Received Samples;
  Tx_bb_samples(length(Rx_bb_samples)+1:end) = [];%Transmitted Samples;
  xdist_it(length(Rx_bb_samples)+1:end) = [];

  %"Saving" variables (for an updated AM-AM plot)
  yit_PD = Rx_bb_samples/max(abs(Rx_bb_samples));
  xit_PD = Tx_bb_samples/max(abs(Tx_bb_samples));
  xdist_it = xdist_it/max(abs(xdist_it));

  %Plot only if debug_bool is set to 1
  if(debug_bool)
      %Amplitude: AM-AM plot
      hAMAM_2                  =                  CreateFigure('AM-
AM',abs(xit_noPD),abs(yit_noPD),1,PageWidth,GraphHeight);
      figure(hAMAM_2); hold on;
      title('AM/AM of Predistorter');
      CreateFigure('AM-
AM',abs(xit_PD),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM_2);
      CreateFigure('AM-
AM',abs(xit_PD),abs(xdist_it),1,PageWidth,GraphHeight,hAMAM_2);
      plot(abs(xit),abs(xit),'k--');
      legend('x vs y','x vs y_P_D','x vs x_d_i_s_t');
      grid minor;
      set(findall(hAMAM_2,'-
property','FontSize'),'FontSize',GraphFontSize);


       %Phase: AM-PM plot
      thetax_PD = angle(xit_PD);
      idx = thetax_PD < 0;
      thetax_PD(idx) = thetax_PD(idx)+2*pi;

      thetax_noPD = angle(xit_noPD);
      idx = thetax_noPD < 0;
      thetax_noPD(idx) = thetax_noPD(idx)+2*pi;

      theta = angle(yit_noPD);
      theta = theta - thetax_noPD;
      theta = rem(theta,pi);



      thetaap = angle(yit_PD);
      idx = thetaap < 0;
      thetaap(idx) = thetaap(idx)+2*pi;
      thetaap = thetaap - thetax_PD;
      thetaap = rem(thetaap,pi);

      thetaz = angle(xdist_it);
      idx = thetaz < 0;
```

```
        thetaz(idx) = thetaz(idx)+2*pi;
        thetaz = thetaz - thetax_PD;
        thetaz = rem(thetaz,pi);

        hAMPM_2                    =                    CreateFigure('AM-
PM',abs(xit_noPD),theta/pi,1,PageWidth,GraphHeight);
        figure(hAMPM_2)
        title('AM/PM of Predistorter');
        hold on;
        CreateFigure('AM-
PM',abs(xit_PD),thetaap/pi,1,PageWidth,GraphHeight,hAMPM_2);
        CreateFigure('AM-
PM',abs(xit_PD),thetaz/pi,1,PageWidth,GraphHeight,hAMPM_2);
        xlabel('Absolute Value'); ylabel('Phase (\pi radians)');
        legend('x vs y','x vs y_P_D','x vs x_d_i_s_t');
        set(findall(hAMPM_2,'-
property','FontSize'),'FontSize',GraphFontSize);
        grid minor;


        hAMAM_3                    =                    CreateFigure('AM-
AM',abs(xit_PD),abs(xdist_it),1,PageWidth,GraphHeight);
        figure(hAMAM_3)
        title(['Total   AM/AM   of   the   system   for   RF   power   =   '
num2str(10*log10(RF_power/1e-3)) 'dBm']);
        hold on;
        CreateFigure('AM-
AM',abs(xdist_it),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM_3);
        CreateFigure('AM-
AM',abs(xit_PD),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM_3);
        xlabel('Absolute Value'); ylabel('Phase (\pi radians)');
        legend('x vs x_d_i_s_t','x_d_i_s_t vs y_P_D','x vs y_P_D');
        set(findall(hAMAM_3,'-
property','FontSize'),'FontSize',GraphFontSize);
        grid minor;


    end


    %Distortion compensation
    %Apply dispersion compensation with previously computed dispersion
compensation factor
    X_Rx_dist_comp=X_Rx_PD2.*Disp_Comp_pd';

    %EVM
    [evm_dispersioncomp_pd_2,X_Tx_n,X_Rx_n]=evm_calculation(X_Tx,X_Rx_di
st_comp);

    if(debug_bool)

        %Plot Constellation
        hConst_PD_Comp                                                   =
CreateFigure('Const',real(X_Rx_dist_comp),imag(X_Rx_dist_comp),1,PageW
idth,GraphHeight);
        title('Constellation    with    dispersion    compensation    and
predistorter')
```

```
      set(findall(hConst_PD_Comp,'-
property','FontSize'),'FontSize',GraphFontSize);

      %Display EVM
      disp(['        EVM 2nd run: ' num2str(evm_dispersioncomp_pd_2) '
%']);

  end
```

25. System Total – Envelope Detection

Function's name: SystemTotal_ED.m

Simulation of the total system, without and with predistortion, using Envelope Detector.

```
function [OSNR_noPD,OSNR_PD,RF_power,...
evm_nodispersioncomp,evm_dispersioncomp_pd,evm_dispersioncomp_pd_2]...
    = SystemTotal_ED(Amplitude_Factor,FiberTransmission)

format longe
%% Parameters
Kinv = 4;
Qinv = 1;
odd_order = 1;
debug_bool = 1; %if 0, doesn't plot constelations, etc
fiber_akq_bool = 1; %if 0, akq are estimated without the fiber
transmission
if(exist('FiberTransmission','var'))
    %if 0, fiber is never used (system where RH is directly connected
to PU)
    fiber_bool = FiberTransmission;
else
    fiber_bool = 0;
end
%Load system parameters
SystemParameters

%If necessary modify the loaded parameters here

% Envelope Detection needs the carrier
Polarization = 'OSSB';
PARA.Optical.SuperGaussian.boolFilter = 1;  %if we want to apply the
filter
PARA.Optical.SuperGaussian.Fc = 30e9;       %[Hz]
PARA.Optical.SuperGaussian.Bw = 10e9;       %[Hz]

PARA.Electrical.SuperGaussian.boolFilter = 1;   %if we want to apply
the filter
PARA.Electrical.SuperGaussian.Fc = 59e9;        %[Hz]
PARA.Electrical.SuperGaussian.Bw = 10e9;        %[Hz]

PARA.Optical.Polarization = Polarization;


if(~fiber_bool && fiber_akq_bool)
warning('Parameters do not match: no fiber transmission but variable
set to use fiber transmission when estimating predistorter
coefficients');
warning('Changing parameters: no fiber transmission to estimate the
coefficients');
fiber_akq_bool = 0;
end

%% Compute Predistorter
```

```
  % Run the nonlinear system as is and compute the predistorter
coefficients
  % - Predistorter not apllied

  if(debug_bool)
      disp('Starting nonlinear system ... ');
  end

  % TRANSMITTER
  if(debug_bool)
      disp('    > Transmitter: ');
  end
  rng('shuffle');
  prbs=randi([0 1],1,ceil(TimeWindow*BitRate)); %generate bits
  [Tx_bb_samples,
X_Tx]=ofdm_coder(prbs,TimeWindow,BitRate,BitsPerSymbol,NumberOfCarrier
s,CyclicPrefix,RolloffFactor,SampRate,ModulationFormat);
  %If we wanted to apply predistortion, it would be here. We are not
  %applying, so the input of APPLY_PREDISTORTION is the output of
OFDM_CODER
  [xOFDM_bb, xOFDM_IF]=apply_predistortion(Tx_bb_samples); %The signal
is not predistorted here
  [Ein_right_mod, Ein_right, E_transmitted,PowerNoise,PowerData] =
optical_system_MB(Amplitude_Factor*xOFDM_IF);

  %Get OSNR: Optical Signal-to-Noise Ratio (without predistortion)
  OSNR_noPD = PowerData/PowerNoise;

  %Get RF Power: Power of the OFDM signal after RF conversion (it will
be the
  % x-axis in some graphics)
  RF_power = var(Amplitude_Factor*xOFDM_IF);

  if(debug_bool)
      disp(['       OSNR: ' num2str(10*log10(OSNR_noPD)) ' dB']);
      disp(['         RF Power: ' num2str(10*log10(RF_power/1e-3)) '
dBm']);
  end

  % FIBER LINK
  % Only transmit through fiber when fiber_bool is set to 1

  %if we are in the system without fiber, then fiber_akq_bool is sure
to be
  %zero (previously verified)
  if(fiber_akq_bool)
      %fiber model: output signal will have distortion
      if(debug_bool)
          disp(['      > Transmitting through fiber of length '
num2str(PARA.Fiber.FiberLength) '...']);
      end
      [E_transmitted_fiber,f,Sout,phaseFiber_debug] =
smf1(PARA.System.t,E_transmitted,PARA.Fiber.FiberLambda,PARA.Fiber.Fib
erLength,PARA.Fiber.FiberAlpha,PARA.Fiber.FiberDispersion,1);
  else
      %no fiber applied: signals don't have distortion
      if(debug_bool)
```

```
            disp('    > This system has no fiber transmission.')
        end
        E_transmitted_fiber  = E_transmitted;
    end


    %Plot Fiber ouput only if debug_bool is set to 1
    if(debug_bool)
        %Plot Optical Spectrum
        signal=E_transmitted*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
        PSD1=psd(hPsd,signal,hopts);
        data=dspdata.psd([PSD1.Data],
PSD1.Frequencies,'Fs',PARA.System.SampRate);
        hOptSpect             =             CreateFigure('Spectrum',data,-
1,1,PageWidth,GraphHeight,-1,GraphFontSize);
        title('Spectrum without equalization, Optical Domain')
        set(findall(hOptSpect,'-property','FontSize'),'FontSize',20);
        if(PARA.System.BitRate == 40e9)
            xlim([-0.032 0.037]);
        else
            xlim([-32 37]);
        end

        grid off;
    end

    %RECEIVER
    if(debug_bool)
        disp('    > Receiver:')
    end
    [bb_received_I,    bb_received_Q,    bb_received_Id,    bb_received_Qd,
I_PD,Rx_bbI,Rx_bbQ] = OpticalDD_Electrical_ED(E_transmitted_fiber);
    [X_Rx_noPD] = ofdm_decoder(bb_received_I, bb_received_Q);


    [Disp_Comp_nopd,X_Tx_n,X_Rx_n]                                          =
dispersion_compensation(X_Tx,X_Rx_noPD);
    X_Rx_noPD_comp=X_Rx_n.*Disp_Comp_nopd';        %Apply       dispersion
compensation

    [evm_nodispersioncomp,X_Tx_n,X_Rx_n]                                    =
evm_calculation(X_Tx,X_Rx_noPD_comp);


    %Plot these graphics/Display text  only if debug_bool is set to 1
    if(debug_bool)
        %Plot Electric Spectrum
        signal=I_PD*sqrt(PARA.System.SampRate)*2*sqrt(1e-3);
        PSD1=psd(hPsd,signal,hopts);
        data=dspdata.psd([PSD1.Data],
PSD1.Frequencies,'Fs',PARA.System.SampRate);
        hElecSpect            =             CreateFigure('Spectrum',data,-
1,1,PageWidth,GraphHeight,-1,GraphFontSize);
        title('Spectrum without equalization, Electrical Domain')
        set(findall(hElecSpect,'-
property','FontSize'),'FontSize',GraphFontSize);
```

```
    %Plot Constellation
    hConst_noPD                                                  =
CreateFigure('Const',real(X_Rx_noPD),imag(X_Rx_noPD),1,PageWidth,Graph
Height);
    title('Constellation of non-linear system')
    xlabel('Real');ylabel('Imaginary');
    set(findall(hConst_noPD,'-
property','FontSize'),'FontSize',GraphFontSize);

    % Display EVM
    disp(['                      EVM    of    non-linear    system:    '
num2str(evm_nodispersioncomp) ' %']);
  end


  %Prepare to compute coefficients:
  Rx_bb_samples = bb_received_Id+1i*bb_received_Qd;   %Received Samples;
  Tx_bb_samples(length(Rx_bb_samples)+1:end)=[];           %Transmitted
Samples;

  %Renaming for clarity
  xit = Tx_bb_samples;
  yit = Rx_bb_samples;

  %"Saving" variables (we need un-predistorted signals for AM-AM plot)
  xit_noPD = xit;
  yit_noPD = yit;

  yit_noPD = yit_noPD/max(abs(yit_noPD));
  xit_noPD = xit_noPD/max(abs(xit_noPD));




  xit(1)=[];
  yit(1)=[];

  N = length(xit);

  %Making sure we have column-vectors
  xit = reshape(xit,N,1);
  yit = reshape(yit,N,1);

  %Normalizing
  Gx = max(abs(xit));
  xit = xit./Gx;
  yit = yit./max(abs(yit));



  %Computing Memory vectors:
  % Xinv is needed to compute predistortion signal
  % Yinv is needed to compute coefficients for predistorter
  Xinv = MemoryVector(xit,Kinv,Qinv,odd_order);
  Yinv = MemoryVector(yit,Kinv,Qinv,odd_order);

  %Compute Coefficients: Minimum Squares Method
```

```
a_kq=pinv(Yinv)*xit;%Compute the coeficients of the predistorter




%% System with distortion compensation and pre-distorter
% Run the non-linear system with predistortion, compute dispersion
% compensation factor and apply it
% - Predistortion applied
% - dispersion compensation factor computed
% - dispersion compensation applied


if(debug_bool)
    disp('Starting system with predistortion... ');
end

% TRANSMITTER
if(debug_bool)
    disp('    > Transmitter: ');
end
rng('shuffle');
prbs=randi([0 1],1,ceil(TimeWindow*BitRate)); %Generating bits
[Tx_bb_samples,
X_Tx]=ofdm_coder(prbs,TimeWindow,BitRate,BitsPerSymbol,NumberOfCarrier
s,CyclicPrefix,RolloffFactor,SampRate,ModulationFormat);

if(debug_bool)
    disp('    Applying predistortion... ');
end
%Transmitter - prepare to apply predistortion
xit = Tx_bb_samples;
xit = reshape(xit,length(xit),1);%making sure its a column-vector

%Normalizing
Gx=max(abs(xit));
xit=xit./Gx;

X = MemoryVector(xit,Kinv,Qinv,odd_order);
xdist_it=Gx*X*a_kq; %This iteration's predistortion signal (a_kq was
previously computed)


%Applying predistortion
[xOFDM_bb, xOFDM_IF] = apply_predistortion(xdist_it); %The signal is
predistorted here

%Transmitter - continued
[Ein_right_mod,                                            Ein_right,
E_transmitted,PowerNoise,PowerData]=optical_system_MB(Amplitude_Factor
*xOFDM_IF);


%Get OSNR: Optical Signal-to-Noise Ratio (with predistortion)
OSNR_PD = PowerData/PowerNoise;
if(debug_bool)
```

```
      disp(['       OSNR: ' num2str(10*log10(OSNR_PD)) 'dB']);
  end

  % FIBER LINK
  if(fiber_bool)
      % Here, we always want to transmit through fiber (in the system
with fiber) because we will compute
      % the dispersion compensation factor
      if(debug_bool)
          disp(['       >  Transmitting  through  fiber  of  length  '
num2str(PARA.Fiber.FiberLength) '...']);
      end

      [E_transmitted_fiber,f,Sout]                                    =
smf1(PARA.System.t,E_transmitted,PARA.Fiber.FiberLambda,PARA.Fiber.Fib
erLength,PARA.Fiber.FiberAlpha,PARA.Fiber.FiberDispersion,1);
  else
      %In the system without fiber (RH conected to PU):
      E_transmitted_fiber = E_transmitted;
  end

  % RECEIVER
  if(debug_bool)
      disp('    > Receiver: ');
  end
  [bb_received_I, bb_received_Q, bb_received_Id, bb_received_Qd, I_PD ]
= OpticalDD_Electrical_ED(E_transmitted_fiber);
  [X_Rx_PD]=ofdm_decoder(bb_received_I, bb_received_Q);


  %Prepare to plot AM-AM
  Rx_bb_samples=bb_received_Id+1i*bb_received_Qd;%Received Samples;
  Tx_bb_samples(length(Rx_bb_samples)+1:end) = [];%Transmitted Samples;
  xdist_it(length(Rx_bb_samples)+1:end) = [];

  %"Saving" variables (we also need predistorted signals for AM-AM plot)
  yit_PD = Rx_bb_samples/max(abs(Rx_bb_samples));
  xit_PD = Tx_bb_samples/max(abs(Tx_bb_samples));
  xdist_it = xdist_it/max(abs(xdist_it));


  %Plot only if debug_bool is set to 1
  if(debug_bool)
      %Amplitude: AM-AM plot
      hAMAM                         =                    CreateFigure('AM-
AM',abs(xit_noPD),abs(yit_noPD),1,PageWidth,GraphHeight);
      figure(hAMAM); hold on;
      title('AM/AM of Predistorter');
      CreateFigure('AM-
AM',abs(xit_PD),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM);
      CreateFigure('AM-
AM',abs(xit_PD),abs(xdist_it),1,PageWidth,GraphHeight,hAMAM);
  %     plot(abs(xit),abs(xit),'k--');
      legend('Original   input   vs.   Original   Output','Input   vs.
Predistorted Output','Input vs. Predistortion signal ','y=x');
      grid minor;
```

```
        set(findall(hAMAM,'-property','FontSize'),'FontSize',GraphFontSize);

         %Phase: AM-PM plot
        thetax_PD = angle(xit_PD);
        idx = thetax_PD < 0;
        thetax_PD(idx) = thetax_PD(idx)+2*pi;

        thetax_noPD = angle(xit_noPD);
        idx = thetax_noPD < 0;
        thetax_noPD(idx) = thetax_noPD(idx)+2*pi;

        theta = angle(yit_noPD);
        theta = theta - thetax_noPD;
        theta = rem(theta,pi);



        thetaap = angle(yit_PD);
        idx = thetaap < 0;
        thetaap(idx) = thetaap(idx)+2*pi;
        thetaap = thetaap - thetax_PD;
        thetaap = rem(thetaap,pi);

        thetaz = angle(xdist_it);
        idx = thetaz < 0;
        thetaz(idx) = thetaz(idx)+2*pi;
        thetaz = thetaz - thetax_PD;
        thetaz = rem(thetaz,pi);

        hAMPM = CreateFigure('AM-PM',abs(xit_noPD),theta/pi,1,PageWidth,GraphHeight);
        figure(hAMPM)
        title('AM/PM of Predistorter');
        hold on;
        CreateFigure('AM-PM',abs(xit_PD),thetaap/pi,1,PageWidth,GraphHeight,hAMPM);
        CreateFigure('AM-PM',abs(xit_PD),thetaz/pi,1,PageWidth,GraphHeight,hAMPM);
        xlabel('Absolute Value In'); ylabel('Phase (\pi radians)');
        legend('Original input vs. Original Output','Input vs. Predistorted Output','Input vs Predistortion signal');
        set(findall(hAMPM,'-property','FontSize'),'FontSize',GraphFontSize);
        grid minor;
    end


    %Receiver - Distortion compensation
    % Even if the fiber is not used, dispersion compensation should be used
    % because OSSB introduces a phase shift because of the use of the Hilbert
    % Transform when modulating the Mach-Zehnder
    % If we are not using OSSB, then there is no dispersion nor phase shift
    % being introduced and this step does no harm
```

```
  [Disp_Comp_pd,X_Tx_n,X_Rx_n] = dispersion_compensation(X_Tx,X_Rx_PD);
  X_Rx_dist_comp=X_Rx_n.*Disp_Comp_pd'; %Apply dispersion compensation

  [evm_dispersioncomp_pd,X_Tx_n,X_Rx_n]                          =
evm_calculation(X_Tx,X_Rx_dist_comp);

  if(debug_bool)

      %Plot Constellation
      hConst_PD                                                   =
CreateFigure('Const',real(X_Rx_dist_comp),imag(X_Rx_dist_comp),1,PageW
idth,GraphHeight);
      title('Constellation    with    dispersion    compensation    and
predistorter')
      set(findall(hConst_PD,'-
property','FontSize'),'FontSize',GraphFontSize);

      % Display EVM
      disp(['         EVM 1st run: ' num2str(evm_dispersioncomp_pd) '
%']);


      %Plot BOTH Constellation
      L1 = 'Without PD'; L2 = 'With PD';
      hConst2                                                     =
CreateFigure('Const2',X_Rx_noPD,X_Rx_dist_comp,1,PageWidth,GraphHeight
,-1,GraphFontSize,L1,L2);
      title('Received Constellations')
      set(findall(hConst2,'-
property','FontSize'),'FontSize',GraphFontSize);

  end

  %% 2nd run: System with distortion compensation and pre-distorter
  % Run system with predistortion and with dispersion compensation
  % - Predistortion applied
  % - dispersion compensation factor not computed
  % - dispersion compensation applied

  if(debug_bool)
      disp(['Starting 2nd run: system with predistortion and dispersion
'...
      'compensation  with  the  previously  computed  coefficients  and
dispersion'...
      ' compensation factor... ']);
  end

  % TRANSMITTER
  if(debug_bool)
      disp('   > Transmitter: ');
  end
  rng('shuffle');
  prbs=randi([0 1],1,ceil(TimeWindow*BitRate)); %Generating bits
  [Tx_bb_samples,
X_Tx]=ofdm_coder(prbs,TimeWindow,BitRate,BitsPerSymbol,NumberOfCarrier
s,CyclicPrefix,RolloffFactor,SampRate,ModulationFormat);
```

```
    %Transmitter - prepare to apply predistortion
    if(debug_bool)
        disp('    Applying predistortion...');
    end
    xit=Tx_bb_samples;
    xit=reshape(xit,length(xit),1);

    %Normalizing
    Gx=max(abs(xit));
    xit=xit./Gx;
    X = MemoryVector(xit,Kinv,Qinv,odd_order);
    xdist_it=Gx*X*a_kq; %This iteration's predistortion signal (a_kq was
previously computed)


    %Applying predistortion
    [xOFDM_bb, xOFDM_IF] = apply_predistortion(xdist_it); %The signal is
predistorted here
    [Ein_right_mod,                                        Ein_right,
E_transmitted,PowerNoise,PowerData]=optical_system_MB(Amplitude_Factor
*xOFDM_IF);



    % FIBER LINK
    if(fiber_bool)
        %System with Fiber (RH not directly conected to PU)
        % We always want to transmit through fiber here because we will
apply
        % distortion compensation
        if(debug_bool)
            disp(['       > Transmitting through fiber of length '
num2str(PARA.Fiber.FiberLength) '...']);
        end

        [E_transmitted_fiber,f,Sout]                                    =
smf1(PARA.System.t,E_transmitted,PARA.Fiber.FiberLambda,PARA.Fiber.Fib
erLength,PARA.Fiber.FiberAlpha,PARA.Fiber.FiberDispersion,1);
    else
        %In the system without fiber (RH conected to PU):
        E_transmitted_fiber = E_transmitted;
    end


    %RECEIVER
    if(debug_bool)
        disp('    > Receiver: ');
    end
    [bb_received_I, bb_received_Q, bb_received_Id, bb_received_Qd, I_PD ]
= OpticalDD_Electrical_ED(E_transmitted_fiber);
    [X_Rx_PD2]=ofdm_decoder(bb_received_I, bb_received_Q);

    Rx_bb_samples=bb_received_Id+1i*bb_received_Qd;%Received Samples;
    Tx_bb_samples(length(Rx_bb_samples)+1:end) = [];%Transmitted Samples;
    xdist_it(length(Rx_bb_samples)+1:end) = [];

    %"Saving" variables (for an updated AM-AM plot)
```

```
  yit_PD = Rx_bb_samples/max(abs(Rx_bb_samples));
  xit_PD = Tx_bb_samples/max(abs(Tx_bb_samples));
  xdist_it = xdist_it/max(abs(xdist_it));

  %Plot only if debug_bool is set to 1
  if(debug_bool)
      %Amplitude: AM-AM plot
      hAMAM_2                    =                    CreateFigure('AM-
AM',abs(xit_noPD),abs(yit_noPD),1,PageWidth,GraphHeight);
      figure(hAMAM_2); hold on;
      title('AM/AM of Predistorter');
      CreateFigure('AM-
AM',abs(xit_PD),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM_2);
      CreateFigure('AM-
AM',abs(xit_PD),abs(xdist_it),1,PageWidth,GraphHeight,hAMAM_2);
  %     plot(abs(xit),abs(xit),'k--');
  %         legend('Original  input  vs.  Original  Output','Input  vs.
Predistorted Output','Input vs. Predistortion signal ','y=x');
      legend('x vs y','x vs y_P_D','x vs x_d_i_s_t');
      grid minor;
      set(findall(hAMAM_2,'-
property','FontSize'),'FontSize',GraphFontSize);

       %Phase: AM-PM plot
      thetax_PD = angle(xit_PD);
      idx = thetax_PD < 0;
      thetax_PD(idx) = thetax_PD(idx)+2*pi;

      thetax_noPD = angle(xit_noPD);
      idx = thetax_noPD < 0;
      thetax_noPD(idx) = thetax_noPD(idx)+2*pi;

      theta = angle(yit_noPD);
      theta = theta - thetax_noPD;
      theta = rem(theta,pi);


      thetaap = angle(yit_PD);
      idx = thetaap < 0;
      thetaap(idx) = thetaap(idx)+2*pi;
      thetaap = thetaap - thetax_PD;
      thetaap = rem(thetaap,pi);

      thetaz = angle(xdist_it);
      idx = thetaz < 0;
      thetaz(idx) = thetaz(idx)+2*pi;
      thetaz = thetaz - thetax_PD;
      thetaz = rem(thetaz,pi);

      hAMPM_2                    =                    CreateFigure('AM-
PM',abs(xit_noPD),theta/pi,1,PageWidth,GraphHeight);
      figure(hAMPM_2)
      title('AM/PM of Predistorter');
      hold on;
      CreateFigure('AM-
PM',abs(xit_PD),thetaap/pi,1,PageWidth,GraphHeight,hAMPM_2);
```

```
        CreateFigure('AM-
PM',abs(xit_PD),thetaz/pi,1,PageWidth,GraphHeight,hAMPM_2);
        xlabel('Absolute Value'); ylabel('Phase (\pi radians)');
        legend('x vs y','x vs y_P_D','x vs x_d_i_s_t');
        set(findall(hAMPM_2,'-
property','FontSize'),'FontSize',GraphFontSize);
        grid minor;

        hAMAM_3                          =                    CreateFigure('AM-
AM',abs(xit_PD),abs(xdist_it),1,PageWidth,GraphHeight);
        figure(hAMAM_3)
        title(['Total   AM/AM   of   the   system   for   RF   power   =   '
num2str(10*log10(RF_power/1e-3)) 'dBm']);
        hold on;
        CreateFigure('AM-
AM',abs(xdist_it),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM_3);
        CreateFigure('AM-
AM',abs(xit_PD),abs(yit_PD),1,PageWidth,GraphHeight,hAMAM_3);
        xlabel('Absolute Value'); ylabel('Phase (\pi radians)');
        legend('x vs x_d_i_s_t','x_d_i_s_t vs y_P_D','x vs y_P_D');
        set(findall(hAMAM_3,'-
property','FontSize'),'FontSize',GraphFontSize);
        grid minor;

    end


    %Distortion compensation
    %Apply dispersion compensation with previously computed dispersion
compensation factor
    X_Rx_dist_comp=X_Rx_PD2.*Disp_Comp_pd';

    %EVM
    [evm_dispersioncomp_pd_2,X_Tx_n,X_Rx_n]=evm_calculation(X_Tx,X_Rx_di
st_comp);

    if(debug_bool)

        %Plot Constellation
        hConst_PD_Comp                                                     =
CreateFigure('Const',real(X_Rx_dist_comp),imag(X_Rx_dist_comp),1,PageW
idth,GraphHeight);
        title('Constellation    with    dispersion    compensation    and
predistorter')
        set(findall(hConst_PD_Comp,'-
property','FontSize'),'FontSize',GraphFontSize);

        %Display EVM
        disp(['        EVM 2nd run: ' num2str(evm_dispersioncomp_pd_2) '
%']);

    end
```