Master Thesis

Mode Division Multiplexing using MIMO Processing

Oriol Porcar Pujol

Supervised by:

Christophe Peucheret, Jing Xu, Joan M. Gené

High-Speed Optical Communications Group

February 2012

A la meva família

Technical University of Denmark
Department of Photonics Engineering - High Speed Optical Communications
Elektrovej building 343, DK-2800 Kgs. Lyngby
Denmark
Tel: (+45) 4525 6352
Fax: (+45) 4593 6581
Email: info@fotonik.dtu.dk
http://www.fotonik.dtu.dk/

ABSTRACT

Mode division multiplexing technologies in multimode optical fibers are being considered as one of the possible solution to the ever growing user capacity demands. Problems derived from this kind of fibers can be overcome using Multiple-Input Multiple-Output (MIMO) techniques. In this thesis, computer simulations for mode division multiplexing in multimode optical fibers using MIMO processing are reported. The implementation of the mathematical model is based on a coherent optical communication system with multimode fibers and MIMO equalization. Digital Signal processing is used to estimate the MIMO channel matrix and invert its effects. The simulation environment is an error counting tool able to quickly setup customized experiments through a user-friendly graphical interface, providing numerous output displays. Theoretical validation simulations and system performance results are obtained to validate the system, check parameter dependencies and evaluate the link performance results. An overall insight of the effectiveness of MIMO technology applied to mode division multiplexed transmission systems is provided.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

In the post-globalized world we live in, real-time communications are a key element in our society. Users have an ever growing bandwidth demand of quality services and multimedia contents (like cloud computing or high quality video streams). In the near future, the current communication network infrastructure will not be able to support these demands. New technological approaches are required to provide this world's capacity demands.

Optical communication systems are one of these approaches. In fact, optical fibers have allowed high-capacity long-haul transmissions, and are huge part of making the internet available around the world. Most of these fibers are single-mode fibers (SMF), and have been fully developed for the past decades. Another kind of fibers, multi-mode fibers (MMF), have also been used for much less demanding services. The reasons of choosing MMF instead of SMF lie within MMF properties. Unlike SMF, MMFs are easier to make, cable, connect and splice, mostly because their large core dimensions (50-100 micrometers). Combined with the lower cost and hardware requirements, it looks clear that MMFs are the most appropriate optical solution for metro-access communications, where connections and optical devices are more concentrated.

However, using MMF has a major downside: due to their larger core-size (much larger than the light's wavelength carried within), it supports more than one propagation mode; hence it is limited by modal dispersion, lowering their capacity throughput. Nevertheless, these propagated modes have the property to be orthogonal eigenmodes of the electromagnetic input fields, thus allowing mode division multiplexing. Unfortunately, the mode coupling between the propagated modes breaks this orthogonality, complicating the exploit of this property.

It has not been until recently that mode coupling is seen instead as an advantage (with the concept of principal modes), turning it into an extra diversity dimension that Multiple-Input Multiple-Output (MIMO) technologies can use to make MMF a viable multi-channel transmission system through mode division multiplexing.

The purpose of this thesis is to evaluate the use of multimode fibers with mode division multiplexing and MIMO processing. This thesis will analyze the viability of this option as a possible solution for the increasing capacity demands, especially on the metro and access networks.

While SMF research seems to be reaching their limits, MMF new technologies offer a new challenging field to investigate and bring the capacity further. Several approaches to mode division multiplexing with MIMO processing have been reported for the past years ([5], [22], [28], [30], [31], [44]). Some of them have already proven high capacity and performance, but the implemented solutions are quite different in every situation. It is yet another proof that there are still many possibilities to explore.

The theoretical background required to understand the elements, parameters, effects and technologies involved in this particular optical communications system is explained in chapter 2. This includes the light source, the medium and the receiver.

The system solution implemented here is coded with Matlab®, and contains all the elements of this optical communication system considering a symmetric MIMO channel: N transmitters, one MMF supporting N modes, and N receivers with the corresponding MIMO equalizer. The full description and implementation of these elements can be found in chapter 3.

Different simulations and system validations are exposed in chapter 4. The simulation procedure starts with the transmission of known pseudo-random sequences through a mode-multiplexed MMF, which are then received and equalized by the MIMO receiver, and then evaluated by counting the errors produced. The implemented solution in this thesis is provided with an easy-to-use interface to allow customized simulations, encouraging the readers to try their own experiments or verify the results.

And last, the conclusions of the results obtained with the implemented model can be found in chapter 5. An analysis of the model structure and behavior is discussed, and further improvements and extensions for this particular solution are exposed.

As a personal comment, this thesis has proven to be very challenging, and has required many trial and errors to understand that there is not only one right answer to this solution. I would also like to thank all my supervisors, Christophe, Jing and Joan M., for their support and time dedication during these last six months. Thanks as well to the whole Fotonik High-Speed Optical Communications group for creating such an encouraging working atmosphere. Special thanks to Francesco Da Ros for his help and support during the realization of this project.

2
THEORETICAL OVERVIEW

In order to fully understand the system implementation and the parameters involved in the current project, it is required to have a good understanding of the basics of optical communication theory and MIMO techniques. In this chapter a general overview of every component involved in the communication system is exposed.

All the theoretic statements in this chapter without reference mention are covered in well-known scientific literature, like [1], [8] for optical communication theory, and [3], [7] for MIMO technology. Many figures used in this chapter have their source from [2] and have their owner's approval to be used in this report.

2.1 The optical fiber

Optical silica-based fibers are the most important transmission medium in nowadays global communications. They allow long-distance and large-capacity communications, mostly because of their very low loss characteristics.

In a few words, optical fibers are waveguides where light (an electromagnetic wave) can propagate through their core, confined by the cladding. The electromagnetic field propagation in the medium will be ruled by Maxwell's equations, while boundary conditions imposed by the continuity of the fields determine the wave solutions (modes) that will travel through it. The physical structure of the fiber can be seen in figure 2.1.

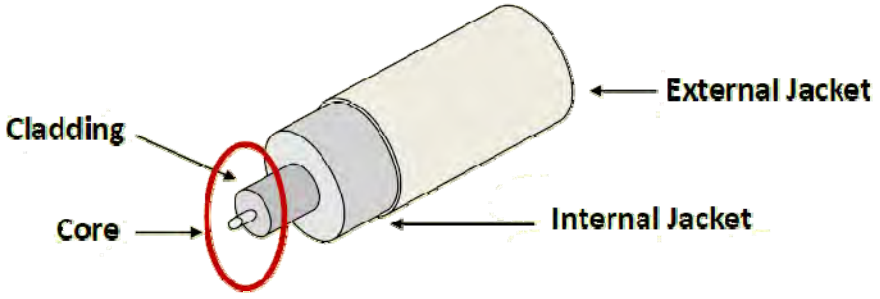

Figure 2.1: Physical structure of an optical fiber. [2]

2.1.1 Propagation

The propagation in optical fibers can be described at different levels: from a geometrical point of view (ray optics), to electromagnetic optics (Maxwell) or quantum optics. For the current project the ray optics approximation will prove sufficient.

In the ray optics approximation, four postulates are described:

1) Light travel in the forms of rays
2) An optical medium is characterized by its refractive index n, defined as the ratio of the speed of light in free space ($c \cong 3\text{x}10^8$ m/s) to that in the medium (v):

$$n \equiv \frac{c}{v} \geq 1 \tag{2.1}$$

3) In an inhomogeneous medium, $n(r)$ is a function of the position $r(x, y, z)$.
4) Light rays travel along the path of least time (Fermat's principle).

These postulates explain the way light will behave inside the optical waveguide. We can see this effect in figure 2.2. In order to achieve low loss propagation, light needs to achieve total internal reflection (TIR) inside the core of the optical fiber. This will happen when the refractive index of the cladding (n_2) is smaller than the one of the core (n_1) and the incident angle of the light ray in the core (Φ_r for Refracted Ray 1) is higher than the critical angle (Φ_c). If these conditions are met, there will be no transmitted beam to the cladding and all the power will be reflected back to the core. The relations between angles and refractive indexes follow Snell's law.

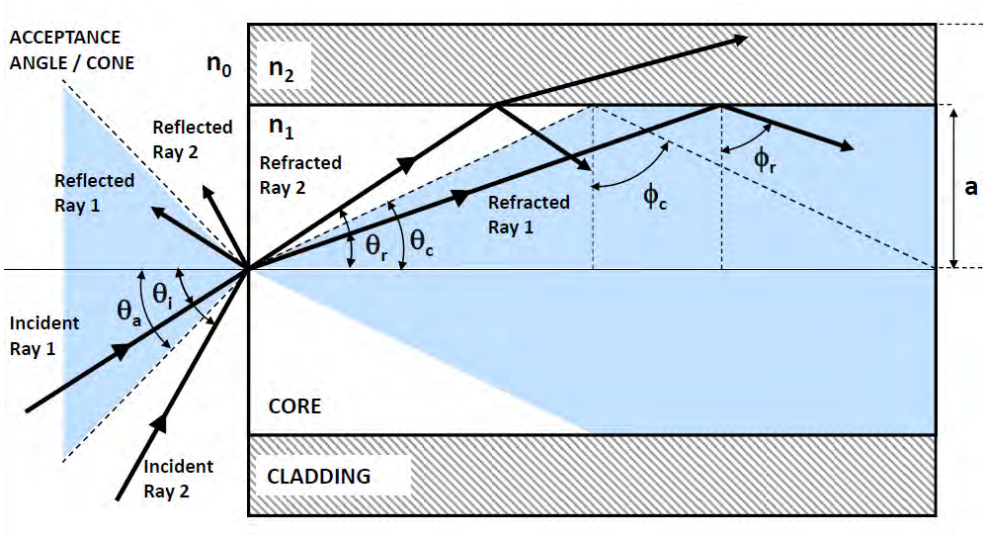

Figure 2.2: Total internal reflection zone inside the core marked with blue area. The radius of the core is defined by a. [2]

The refractive index profile of the core is crucial for the propagation of the electromagnetic field along the fiber. The most common profiles are the step index profile (SI), where the refractive index of the core and the cladding are two discrete values, and the graded index profile (GI), where the index profile is a function of the radial distance.

As seen in figure 2.3, the graded index profiles can causes the different light modes (displayed as different orange light paths) to travel faster as they leave the center (since there the refractive index is lower, and v bigger in equation 2.1). The modes who travel longer distances by going far away from the center of the fiber do so at higher speed than the ones staying closer to the center. This attempts to synchronize the arrival at the output for all the modes.

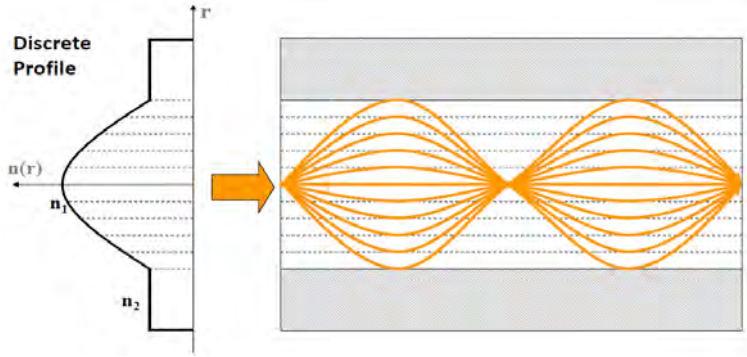

Figure 2.3: Example of a graded index profile fiber. [2]

2.1.2 Modes

An optical guided transversal mode refers to a particular solution to the electromagnetic propagation equations subjected to certain boundary conditions imposed by the waveguide, and its transversal field distribution remains ideally constant with propagation if the structure remains longitudinally invariant. Figure 2.4 shows an example of field distribution that has propagated through the optical fiber.

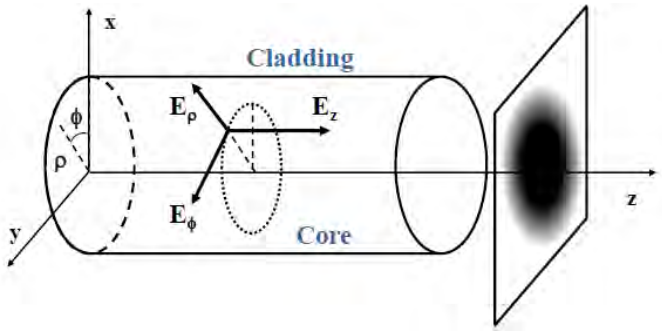

Figure 2.4: Transversal propagation of an electromagnetic wave inside an optical fiber, with its field distribution measured on the plane perpendicular to the propagation axis. [2]

The presence of multiple modes is determined by different factors, being the radius of the core over the wavelength the most obvious one. The number of modes propagated can be obtained calculating the normalized frequency of the light in the fiber. For step index fibers, equations 2.2 can be applied.

$$v = 2\pi \frac{a}{\lambda} NA \quad , \quad NA = \sqrt{{n_1}^2 - {n_2}^2} \quad , \quad b = \frac{n_{eff} - n_1}{n_1 - n_2} \tag{2.2}$$

NA is the Numerical Aperture, λ is the wavelength of the signal, a the radius of the core, and b the normalized propagation constant. n_{eff} refers to the effective refractive index. Once the normalized frequency has been calculated we can see how many modes the current fiber can support in figure 2.5 (for step index profile fibers).

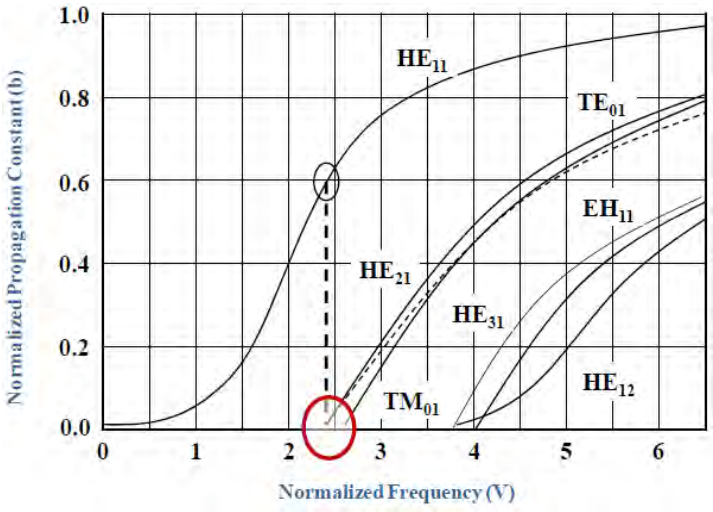

Figure 2.5: Relation between propagation constant and the normalized frequency for the first modes. [2]

Single mode fibers (SMF) only allow the fundamental mode to be propagated along the fiber, and are the best solution for long-haul transmission with high bit rates. These fibers have normalized frequency $v < 2{,}405 \equiv v_c$, known as the single mode condition (marked with a red circle in figure 2.5). Multi Mode fibers (MMF) allow more than the fundamental mode ($v > 2{,}405$), and have more limitations for fast and long transmissions, mainly caused by higher dispersion effects (modal dispersion).

The different modes that propagate through the fiber can be characterized as TE, TM, TEM or Hybrid modes. TE modes (transversal electrical modes) do not have electric field in the propagation direction ($E_z = 0$), TM (transversal magnetic modes) do not have magnetic field in the propagation direction ($H_z = 0$), TEM (transversal electromagnetic modes) have neither, and Hybrid modes have both. Some examples are shown in figure 2.6.

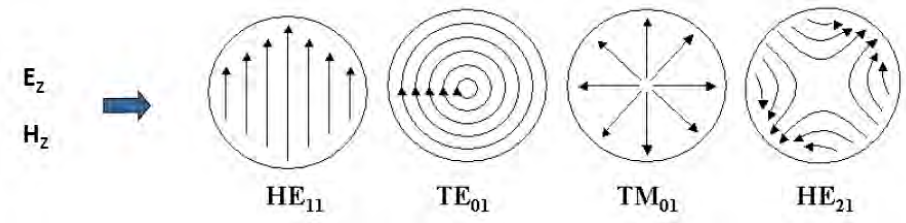

Figure 2.6: Example of four different mode field distributions. [2]

These traditional designations for the conventional modes translate into linearly polarized modes (LP_{ij}) when applying the weakly guiding approximation (meaning that the difference between the refractive indexes of the core and the cladding is very small). In figure 2.7 we can see an example.

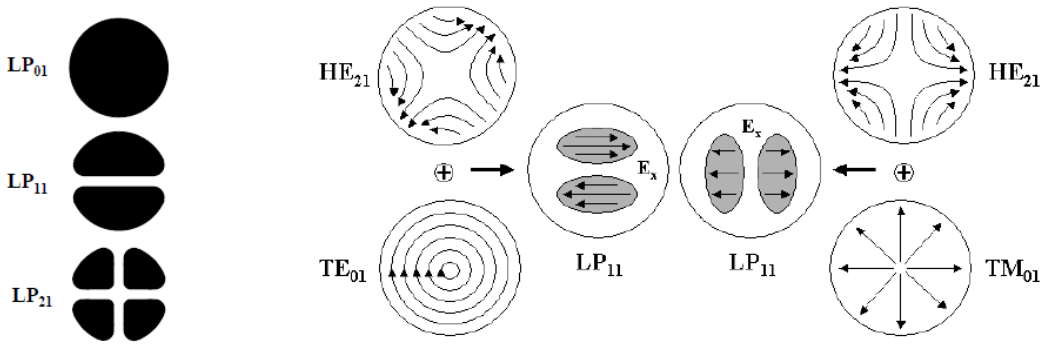

Figure 2.7: Examples of linearly polarized modes and their equivalent traditional designations. [2]

2.1.3 Attenuation

When propagating through the fiber, light experiences exponential decay of its optical power over the distance as a consequence of absorption and scattering effects.

All materials absorb energy at their electronic and vibration resonance wavelengths, and also from other effects such as scattering losses. This absorption can be due to the basic material (intrinsic), like SiO_2 (most common material in optical fibers), or due to impurities in the material (extrinsic) like metals, OH^+ water radicals, dopants in the fiber, etcetera.

The scattering losses come mostly from the Rayleigh scattering effect, which happens due to microscopic density fluctuations of the material, inducing small variations on their refractive index, making some portion of the light to change direction. In an optical fiber, these changes of direction will result in the loss of some optical power that falls outside the total internal reflection area.

Other sources of scattering are the Mie scattering, caused by waveguide structure imperfections (comparable to the size of the wavelength), and bending losses, when part of the evanescent field is forced to be radiated outside the propagation mode.

Figure 2.8 shows the optical fiber attenuation curve for different wavelengths, which include the first three windows at 850nm, 1300nm and 1550nm.

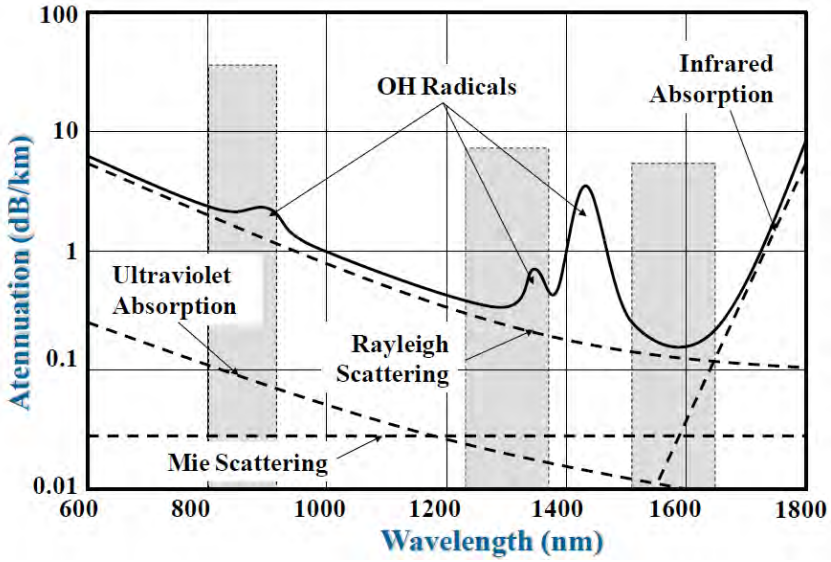

Figure 2.8: Optical fiber attenuation curve. [2]

2.1.4 Dispersion

Optical pulses that propagate through an optical fiber suffer from an energy spreading in time, producing an increase of the pulse width. This pulse spreading is critical and limits the maximum length of the communication link. The effect can be seen in figure 2.9.

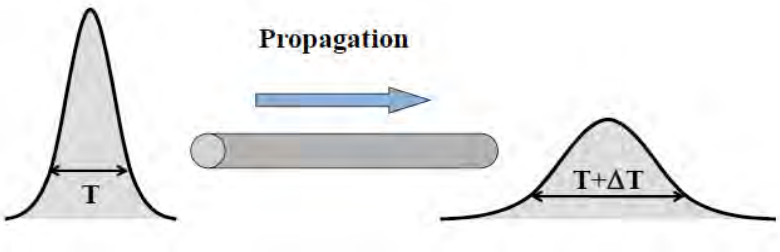

Figure 2.9: Dispersion effect in propagated optical pulses. [2]

The intermodal (or modal) dispersion is induced by the differential delay caused by the different propagation speeds of every mode. The result at the receiver is the broadening of pulses, which are the addition of the received pulses, delayed each one of

them depending on their modal speed. This kind of dispersion, which is dominant compared to other kind of dispersions, only happens in multi-mode fibers, and its effect is shown in figure 2.10.

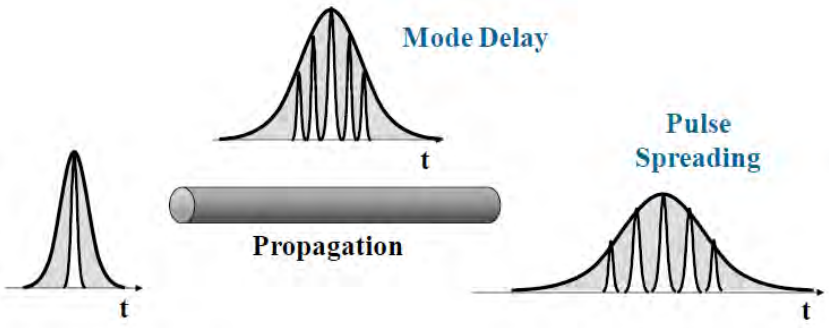

Figure 2.10: Pulse spreading caused by modal dispersion. [2]

The intramodal dispersion is the combined contribution of different dispersion effects: the group velocity dispersion (including material and waveguide dispersion) and the polarization mode dispersion.

The group velocity dispersion (GVD), also known as Chromatic Dispersion (CD), has its origin in the frequency dependence of the refractive index. Since the group velocity is tied with the refractive index, and because the signal spectrum has some bandwidth (‘color’, hence the name chromatic), the different wavelength components will travel at different speeds, broadening the input pulses, as shown in figure 2.11.

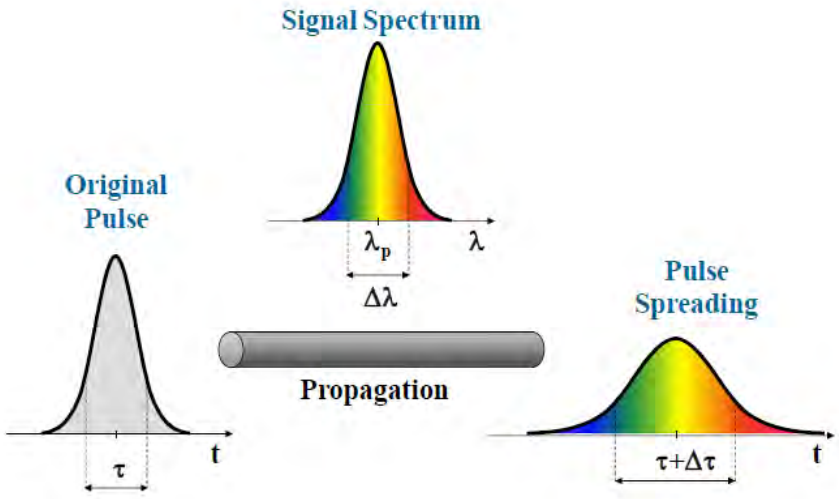

Figure 2.11: Pulse spreading caused by chromatic dispersion. [2]

The propagation constant of every spectral component $\beta(\omega)$ can be expanded as a Taylor series around ω_0 as shown in equation 2.3:

$$\beta(\omega) = \beta_0 + \beta_1(\omega - \omega_0) + \frac{1}{2}\beta_2(\omega - \omega_0)^2 + \frac{1}{6}\beta_3(\omega - \omega_0)^3 + \cdots \qquad (2.3)$$

Since the group velocity is defined as:

$$vg = \left(\frac{d\beta}{d\omega}\right)^{-1} \qquad (2.4)$$

We can see that for $\beta_2 \neq 0$, and higher $\beta_n = \frac{d^n\beta(\omega)}{d\omega^n}\Big|_{\omega=\omega_0} \neq 0$, the group velocity will depend on ω, which leads to the pulse broadening.

The chromatic dispersion is the combination of material dispersion and waveguide dispersion. Both are a cause of the different group velocities that affect the different spectral components of the signal. The material dispersion depends on the refractive-index properties of the materials in the fiber, and the waveguide dispersion depends on the waveguide refraction index profile distribution in the core and the cladding.

In figure 2.12 the combined material and waveguide dispersions are represented, and shows that this combination might result in a zero chromatic dispersion at a certain wavelength.

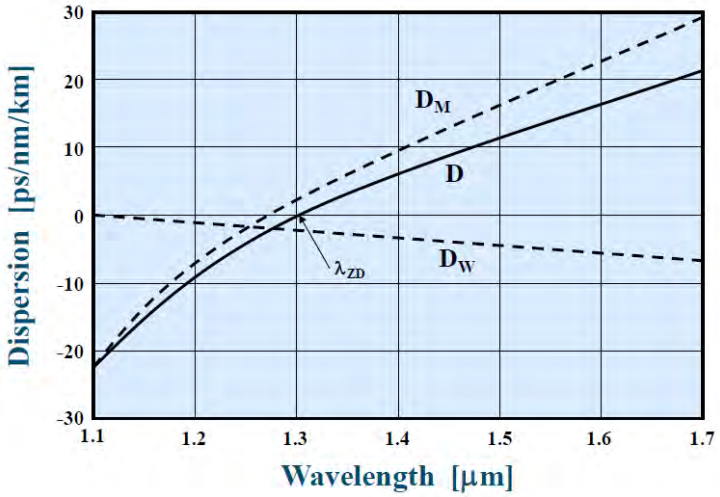

Figure 2.12: Combined material & waveguide dispersion. [2]

The polarization mode dispersion (PMD) comes from the polarization dependence of the refractive index. As a consequence, the light experiences different group delays depending on its polarization. In most situations the effect of PMD can be compared to the chromatic dispersion, in part because the PMD-parameter is proportional to the square root of the link length for long fibers (due to coupling between polarizations), which results in very low dispersions even for symbol rates of 50 Gbaud.

2.1.5 Other effects

There are other effects in advanced optical fiber communications, such as differential group delay decorrelation or non-linear effects on optical fibers (Kerr nonlinearity). These effects will not be considered in this project.

2.2 The optical transmitter

Optical transmitters are the transducer required to convert electrical signals to optical ones that will travel along the optical fibers. To achieve this function, an optical transmitter will require an optical source, a modulator, and multiplexing devices. If the channel is going to be shared with other signals, some kind of diversity is required to recover them. In the particular case of this project, the diversity from the fiber modes will be used, and mode division multiplexing techniques will be required (see 2.4.2).

2.2.1 Optical Source

There are two well-known optical sources: LED and LASER. LEDs (light emitting diode) are diodes (PN junction) directly polarized which emit light by spontaneous emission thanks to an electron-hole recombination process. They produce incoherent light, with random frequency, phase and direction, resulting in a very high bandwidth and low directional power. However, they are very cheap and simple to implement.

LASER (Light Amplification by Stimulated Emission of Radiation) on the other hand consists of an optical resonant cavity (figure 2.13) based on the stimulated emission process, and can be modeled as an amplification system with feedback. It produces coherent light, with very stable frequency, phase and direction of the light. This results in a very narrow spectrum and good directivity.

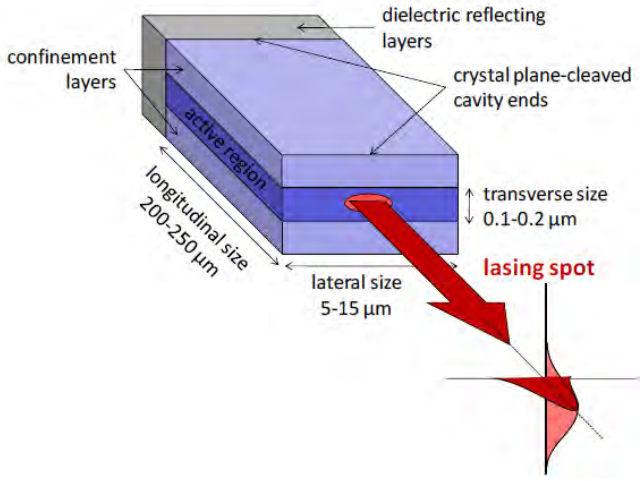

Figure 2.13: Fabry-Perot resonant cavity used as source for coherent light. [2]

The process of coherent light generation depends on the magnitude of the input current. This divides the working regimes of the laser in four regions, where only in the third one we achieve the laser effect, as seen in figure 2.14. In this region, the relation between the input current and the output optical power is considered lineal.

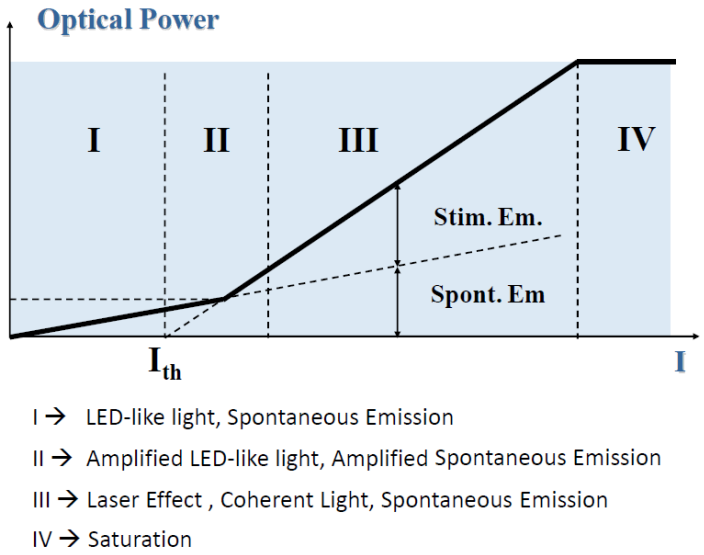

Figure 2.14: Light-current characteristic of a laser, showing its different working regions. [2]

In nowadays communications lasers are used most of the times, while LEDs are used for some less demanding applications, like LAN or free space communication. Many laser systems work with a continuous wave (CW) laser to maintain constant the amplitude and frequency during communication (phase noise is always present, and is responsible for the finite linewidth). This CW signal will be used as the carrier of information once the data is incorporated to the signal in the modulation process.

2.2.2 Modulation

Although modulating the optical sources can be done by changing the electrical input of the transducer (Direct Modulation), it is much better to modulate the constant amplitude lightwave of the CW laser signal by an external modulator, especially at high bit rates and high order modulation formats where direct modulation response times might be too slow compared to external modulators.

The basic component of an external modulator is the phase modulator. This device allows phase modulation of the propagating light through the voltage-induced refractive index of the crystal it contains. This shift is linearly proportional to the input voltage that changes the refractive index of the crystal. This shifting process can be considered instant (negligible compared to the modulation speeds).

With phase modulators we can build more complex modulators, like Mach-Zehnder Modulators, which are based on the Mach-Zehnder interferometer (MZI) (figure 2.15).

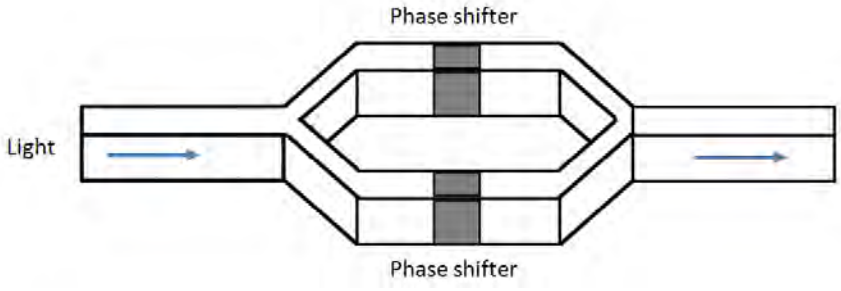

Figure 2.15: Mach-Zehnder Interferometer.

These modulators exploit the phase modulation effects in order to modulate the output signal. If the phase shifts introduced by the phase modulators ($\emptyset_1, \emptyset_2$) are chosen zero ($\emptyset_2 = \emptyset_1 = 0$), then the output field equals to the input field since no shifting has happened. However, if the difference between them is π ($\emptyset_2 - \emptyset_1 = \pi$), they both cancel and the output field is nil. This can be seen in the next equation:

$$E_{out} = \frac{1}{2} E_{in}\left[e^{-j\emptyset_1} + e^{-j\emptyset_2}\right] = E_{in} cos\left[\frac{\emptyset_2 - \emptyset_1}{2}\right] e^{-j\left(\frac{\emptyset_1 + \emptyset_2}{2}\right)} \qquad (2.5)$$

We can define V_π as the input voltage needed to obtain a π shift. If we express equation 2.5 as function of V_π and the differential voltage input $\Delta V = V_2 - V_1$, then the field and the power output are:

$$E_{out} = E_{in} cos\left[\frac{\pi}{2V_\pi}\Delta V\right] e^{-j\frac{\pi}{2V_\pi}(V_1 + V_2)} \qquad P_{out} = P_{in} cos^2\left[\frac{\pi}{2V_\pi}\Delta V\right] \qquad (2.6)$$

If we set $V_2 = -V_1$ then $\Delta V = 2V_2$, and we maintain $\emptyset_2 - \emptyset_1 = \pi$. In this situation the MZI operates in push-pull mode with no phase term in equation 2.6.

In a similar way, an IQ Mach-Zehnder modulator structure uses the previous effect in order to exploit the in-phase (I) and quadrature (Q) of the signal. This requires two arms with a MZ modulator in each, and an additional phase modulator on the Q arm that provides a $\frac{\pi}{2}$ phase shift, as seen in figure 2.16.

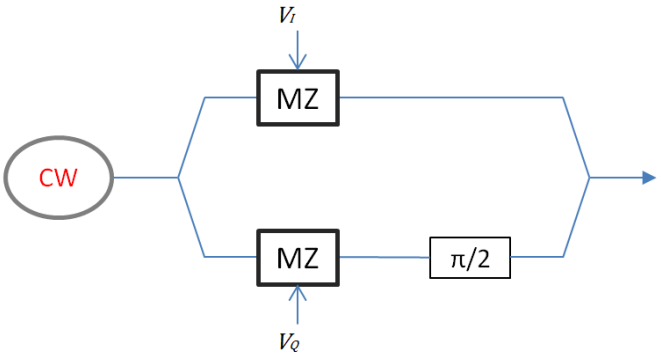

Figure 2.16: IQ Mach-Zehnder modulator.

The expression of the output field with the previous considerations ($V_2 = -V_1$ for each pair of phase shifts), where V_I and V_Q are now the differential voltages for each Mach-Zehnder, is shown in next equation:

$$E_{out} = \frac{1}{2}E_{in}cos\left[\frac{\pi}{2V_\pi}V_I\right] + j\frac{1}{2}E_{in}cos\left[\frac{\pi}{2V_\pi}V_Q\right] \quad (2.7)$$

For the current project only two modulation formats have been included: On Off Keying (OOK) and Differential Quadrature Phase-Shift Keying (DQPSK) [6], [7], [19], with no return to zero (NRZ). Return to zero (RZ) modulation has not been considered.

2.2.2.1 OOK modulation

The On Off Keying (OOK) modulation is the simplest modulation, and basically works as a digital gate: the output signal is the input signal for input bit 1, and returns null for input bit 0. To achieve this, one Mach-Zehnder modulator and a CW Laser are enough. The required differential voltages of the electric signal for the Mach-Zehnder interferometer depend on the modulator structure. For a common modulator, these values are 0 and V_π for bits 1 and 0 respectively in equation 2.6.

Figure 2.17 shows the block diagram of an OOK transmitter, while figure 2.18 represents the modulation constellation normalized to the transmitter power. Only bit 1 transmits energy (at full power).

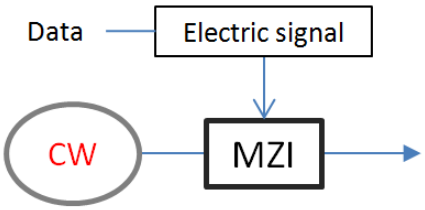

Figure 2.17: Block diagram of an OOK transmitter.

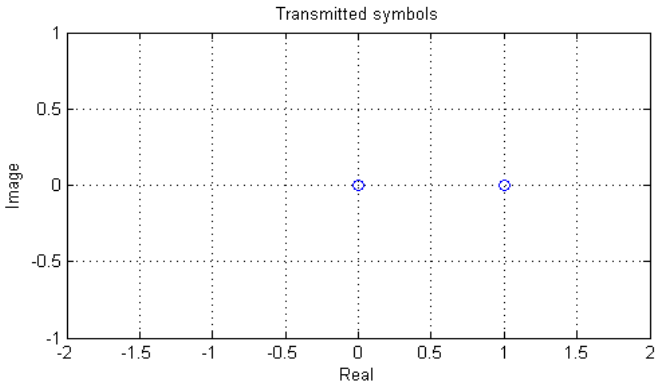

Figure 2.18: OOK constellation.

2.2.2.2 DQPSK modulation

The Differential Quadrature Phase-Shift Keying (DQPSK) modulation generates a multilevel signal of 2 bits per symbol. The modulation uses four points in the constellation diagram equispaced around a circle. The coding of these symbols follows gray coding, where two neighbor symbols only differ by 1 bit, in order to reduce the bit error rate.

One way to modulate a DQPSK signal requires an IQ Mach-Zehnder modulator, and requires some demultiplexing of the input data in order to feed each of the arms. This demultiplexing process has to be known by the receiver to multiplex the incoming bit signals in order to restore the whole original data sequence, and can be as simple as an even-odd splitting process. The block structure in figure 2.19 represents this modulator diagram. Figure 2.20 represents the modulation constellation normalized to the transmitter power, and we can see all symbols transmit at maximum power (are located in the radius one circumference) but have different phase.

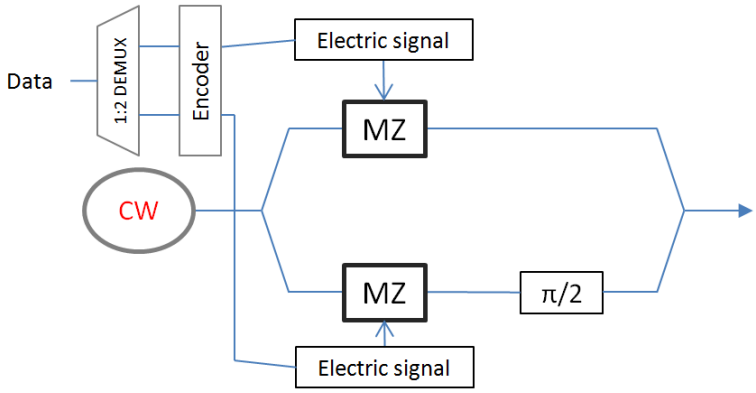

Figure 2.19: Block diagram of a DQPSK transmitter.

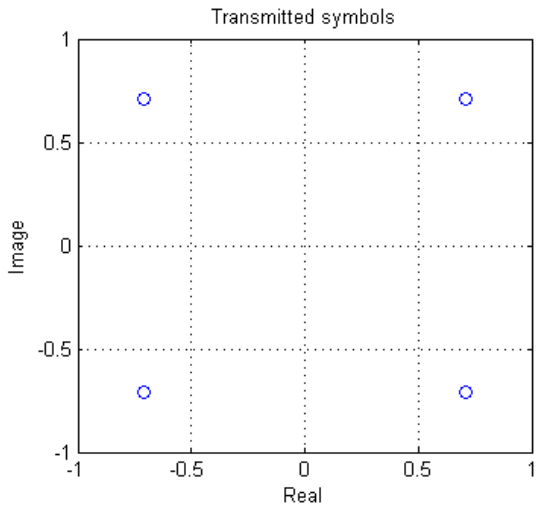

Figure 2.20: DQPSK constellation.

2.3 The optical receiver

The optical receiver is a transducer that converts optical power to electrical power, bringing the original signal back from the optical domain to the electric domain. It also needs to treat the input signal and process it to be able to recover as much as possible the original signal, to finally demodulate the data it contains.

The signal detection can be done in two ways: direct detection or coherent detection.

2.3.1 Direct detection

The typical structure of a direct detection receiver can be divided in five stages, and I shown in figure 2.21. First, a band-pass optical filter is applied to reduce the noise and eliminate any signal outside the working bandwidth. This signal has to be amplified by an optical amplifier to achieve the desired optical power. The amplification can be done before reception, in-line, or right after the transmitter. The photodetection process (using photodiodes) converts the optical signal to electronic signal. This electronic signal might be then low-pass filtered electronically if any sampling is involved (i.e., analog to digital converter) to remove signal aliasing. Finally, the electronic signal enters the decision circuit, which recovers the original data contained in the signal. Clock recovery circuits are needed in order to sample the received signal at the optimum instant.

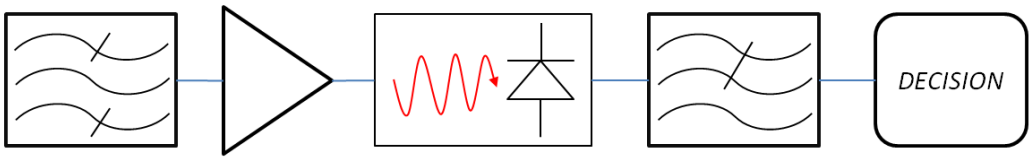

Figure 2.21: Block diagram of a generic direct detection optical receiver with optical amplification.

The photocurrent delivered by the photodiode is linearly proportional to the incident power of the optical signal (proportional by a factor called responsivity). There are different kind of photodiodes, like PN, PIN and APD, but only the last two are useful for optical communications. Thanks to the stimulated absorption principle (where electron-hole pairs are created and can absorb incident photons to increase the energy level of electrons) an increase in the photocurrent is produced. The stimulated absorption process is shown in figure 2.22. The energy from the incoming photon is absorbed by an electron in the ground state, which increases its energy to reach the excited state. E_{gap} is the energy difference between both states.

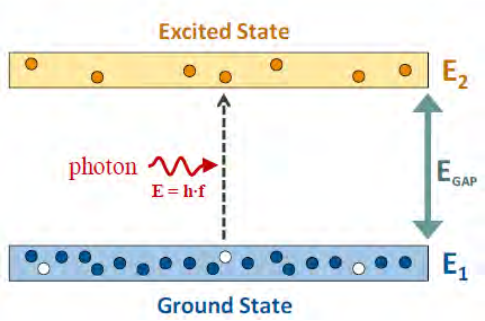

Figure 2.22: Stimulated absorption principle. [2]

After the optical signal has been detected by the photodiode, the decision circuit decides whether the received photocurrent corresponds to logical bit 1 or 0. The decision instants and the photocurrent thresholds differ a lot for different system setups, and some kind of optimization process is needed to obtain the best values.

2.3.2 Coherent receiver

Coherent receivers have been recently investigated for optical communication systems [4], [5]. These receivers are very valuable because they use all the information of the optical signal wave: amplitude, frequency, phase and polarization. This has allowed higher complexity modulation formats for higher spectral efficiency in all kind of communication systems.

The advantages of using coherent detection versus a direct detection one include the ability to detect frequency and phase information, improving receiver sensitivity, and making demultiplexing filters more selective. However, coherent receivers are much more complex, less stable (because of temperature), are limited by the local oscillator phase noise (require phase tracking), and require polarization tracking.

The basic structure of the coherent receiver is shown in figure 2.23.

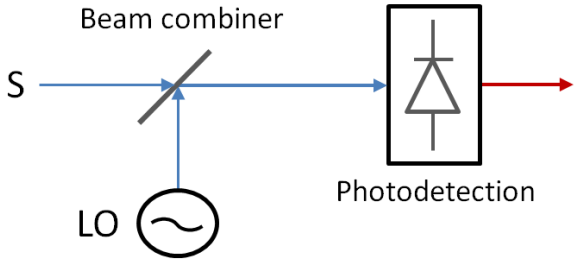

Fig 2.23: Basic structure of coherent detection.

The main idea behind a coherent receiver is the combination of the input field of the signal with a local signal named local oscillator (LO). The combination of the fields of these two signals is then detected by a photodiode. The resultant photocurrent generated

is proportional to their square root product of their power (ignoring the DC component), as we can see in equation 2.8:

$$P_{out} = P_S + P_{LO} + 2\sqrt{P_S P_{LO}} \cdot cos[\omega_{FI} t + \theta(t)] \tag{2.8}$$

If the signal-carrier frequency and the LO frequency are the same, we have homodyne detection ($\omega_{FI} = 0$ in equation 2.8), and the optical spectrum is directly converted to base band. If the frequencies do not match, then we have heterodyne detection ($\omega_{FI} > 0$), and the spectrum is moved to the intermediate frequency, higher than baseband. Heterodyne detection is usually easier to demodulate and synchronize than homodyne.

To avoid constellation rotations in the detector scheme, it is important to compensate the phase shifting effects caused by variations on the signal-carrier or the local oscillator. One way to deal with this problem is with digital signal processing (DSP) circuits that estimate the current carrier frequency and phase differences.

The physical implementation of the optical coherent receiver can be achieved with a Quadrature Optical Hybrid and balanced receivers in order to obtain the in-phase (I) and quadrature (Q) components of the signal. Figure 2.24 represents this implementation.

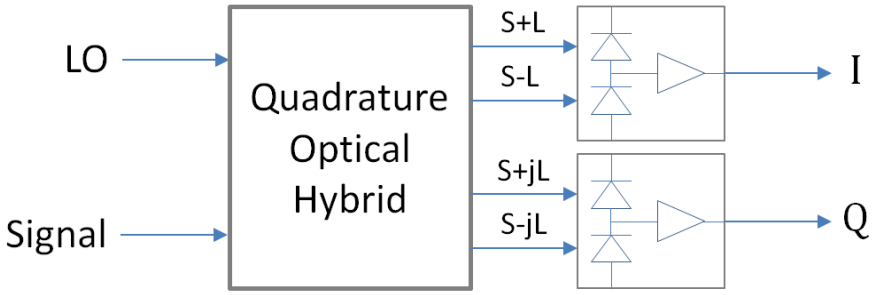

Figure 2.24: Optical coherent receiver (L and LO is the local oscillator and S the optical signal).

The quadrature optical hybrid creates four combinations with the signal and the local oscillator fields, which can be photodetected in the balanced receiver and then arranged in a way to obtain the electrical I and Q components of the original signal.

2.3.3 Optical amplifier and ASE noise

The optical amplifier is usually located inside the receiver. Like any other amplifier, its function is to increase the level of the received signal power. There are different types of optical amplifiers: semiconductor optical amplifiers (SOA), doped-fiber amplifiers (EDFA) and non-linear amplifiers (like Raman and Parametric). As opposite to electric amplifiers, optical amplifiers work with photons rather than with electrons (much heavier). The generic optical amplifier working principle is the same used by

lasers, but without feedback in the active region. The amplifier is pumped (pump signal) to achieve population inversion (more electrons on the excited state than in the ground state), which will produce amplified stimulated emission.

Optical amplification is required to compensate for fiber loss in long-haul systems. The light amplification process is accomplished through stimulated emission, and since most of the laser sources also generate some spontaneous emission due to spontaneous recombination of electron-hole pairs in the active region, this effect generates amplified spontaneous emission (ASE) noise.

In our project, ASE noise will be considered the most important source of noise, and it will be considered with Gaussian statistics. Other noise sources in the receiver (thermal noise, shot noise) will be considered negligible. Furthermore, the ASE spectrum will be considered flat (same value for all wavelengths). In figure 2.25 an example of an optically amplified signal spectrum in the third window (1550 nm) is shown (B_0 represents the signal bandwidth).

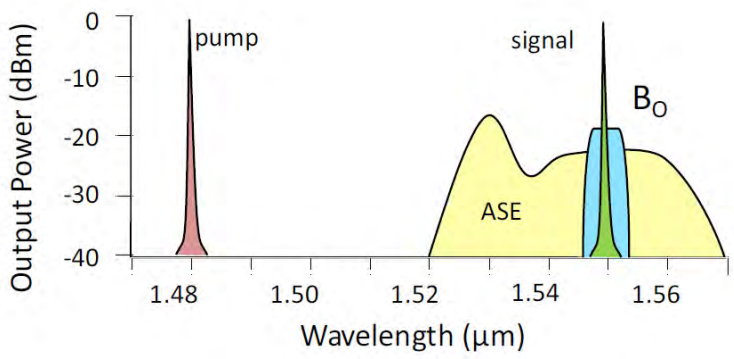

Figure 2.25: Signal spectrum example after optical amplification. [2]

Once we have noise present in the system, we can define an Optical to Signal Noise Ratio (OSNR), which will give a reference of the amount of noise power present in the system for a defined bandwidth (which is 0.1 nm by convention, and translates to 12.5 GHz at 1550 nm).

This ratio will be the one defining the amount of noise in the system, and will be related to the Bit Error Rate results obtained in the simulations.

2.3.4 Other noise sources

The noise affecting the receiver has different sources with different statistics. They all contribute in the total noise variance of the system. Figure 2.26 shows the model location of all the noisy sources of the receiver.

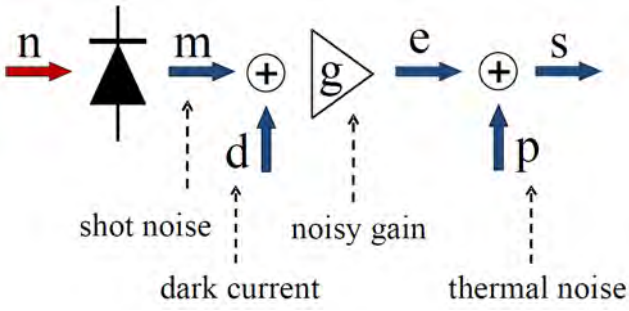

Figure 2.26: All the noise sources in the optical receiver. [2]

Shot noise refers to the random fluctuations in the photocurrent after the photodiode due to the light's inner randomness. Shot noise random contribution has Poison statistics, but can be approximated Gaussian for high average currents. Also, the so called dark current adds to the output current of the photodiode, and generates extra independent shot noise itself.

Thermal noise is caused by the motion of electrons in conductors when no voltage is applied, and happens at any temperature above 0 degrees Kelvin. It can be modeled as a stationary Gaussian process.

For this particular project only the noisy (ASE) gain is considered.

2.4 Multiple-Input Multiple-Output

The Multiple-Input Multiple-Output concept, or MIMO, refers to any communications system where the amount of transmitters and receivers are higher than one at each side. These systems are very common in radio channels [3], [7], where the use of multiple antennas to exploit spatial diversity is very common in technologies like Wi-Fi or GSM. The idea behind MIMO is to exploit the uncorrelated received signals that have crossed this linear multipath channel, and be able to recover all of the transmitted information separately.

Particular cases of MIMO are SIMO or MISO, where S stands for single, where only one transmitter (SIMO) or one receiver (MISO) is used.

The basic structure of a MIMO system is shown in figure 2.27. In this figure, the particular case of a MIMO wireless setup with M_T transmitting antennas and M_R receiving ones. For non wireless environments, the concept of antenna can be replaced with any other signal source.

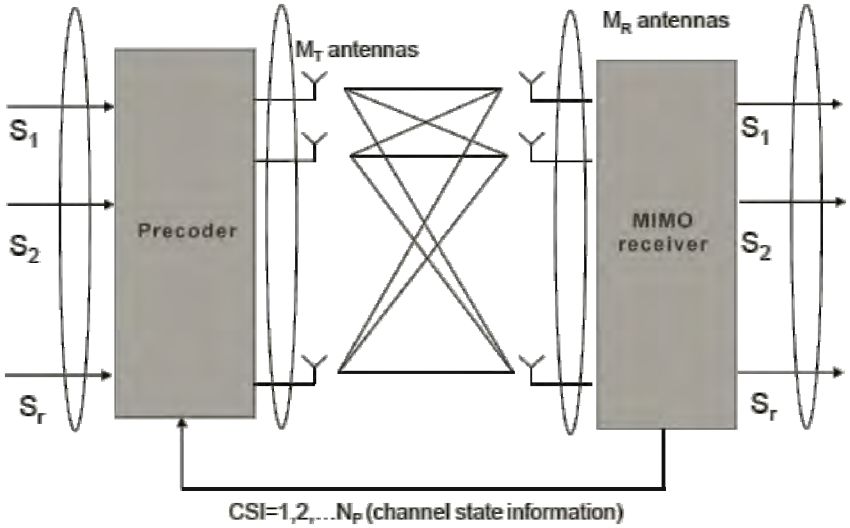

Figure 2.27: Common Multiple-Input Multiple-Output wireless scenario.

The MIMO system can be characterized as a linear equation system, where every output is a linear combination of the different inputs convoluted with the different channel paths.

$$y_i(t) = \sum_{j=1}^{M_T} h_{i,j}(\tau, t) * x_j(t) \tag{2.9}$$

Where $y_i(t)$ is the received signal on receiver *i*, $x_j(t)$ is the signal transmitted by transmitter *j*, and $h_{i,j}(\tau, t)$ is the impulse response of the channel between transmitter *j* and receiver *i*.

These equations can also be compacted in a matrix expression:

$$\mathrm{H}(\tau,\mathrm{t}) = \begin{bmatrix} h_{1,1}(\tau,t) & h_{1,2}(\tau,t) & \ldots\ldots & h_{1,M_T}(\tau,t) \\ h_{2,1}(\tau,t) & h_{2,1}(\tau,t) & \ldots\ldots & h_{2,M_T}(\tau,t) \\ \vdots & \vdots & & \vdots \\ h_{M_R,1}(\tau,t) & h_{M_R,1}(\tau,t) & \ldots\ldots & h_{M_R,M_T}(\tau,t) \end{bmatrix} \tag{2.10}$$

$$\boldsymbol{y}(t) = \mathrm{H}(\tau,\mathrm{t}) * \boldsymbol{x}(t)$$

In order to solve the equation and recover the original signal, MIMO systems need to invert the effect of the H matrix. There are different ways to accomplish this inversion. A very common solution is to estimate the channel matrix through the use of training symbols (or training sequences) which are known for both receiver and transmitter. These training symbols will estimate the channel response, thus estimating the matrix H. This estimation can also be adaptive (with algorithms like Least-Mean Squares,

LMS) and recursive (with algorithms like Recursive Least Squares, RLS [20]), some of them based on the Least Square Error (LSE) estimator [12]. Once this is done, the MIMO receiver uses DSP algorithms (like V-BLAST [10] or Zero-Forcing) to invert the matrix and solve the system. There are many other MIMO estimation algorithms in MIMO literature [9].

The Channel State Information (or CSI) refers to the channel properties information gathered during the transmission. If it is available to the transmitter, the pre-coder in figure 2.27 can adjust the transmitted power to each channel, optimizing the transmit powers and reducing the noise effect on every channel [11]. The CSI information has to travel back to the transmitter, which implies the need of bidirectional links. This might be a problem in optical links.

Other solutions do not require knowing the channel in advance, and simply start equalizing and adapting to the best solution through adaptive filters once enough symbols have been fed to the receiver [12]. The advantage of using adaptive filters is their ability to find good filter parameters by itself in an iterative way.

The use of MIMO to exploit space-division multiplexing (SDM) ideally multiplies the available capacity of a given link by a factor of M, where M is the number of parallel independent transmission paths (related with the number of transmitters and receivers in the system). This makes SDM highly scalable, since M could be chosen high.

2.4.1 MIMO on multimode fibers

The main idea of applying MIMO on multimode fibers is to use the different modes propagating through the fiber as if they were independent transmission links, and use their properties to exploit spatial division multiplexing. The equivalency with the wireless world would be considering multipath fading effects as the modal dispersion, allowing multiple simultaneous transmission channels through one single fiber. In fact, when assumed that the optical link is longitudinally invariant, the optical modes within will not exchange energy. This causes guided modes to become orthogonal eigenmodes, allowing their input fields to be decomposed by this base. However, in most scenarios effects like mode coupling and mode-dependent loss and gain will change this premise.

Spatial mode coupling effects happen when part of the optical signal transmitted in one mode is transferred to other modes along the propagation in the fiber. This crosstalk between modes can be very high for high-coupling fibers, or can be lowered by designing low-coupling fibers. The main consequence of mode coupling is the breaking of the orthogonality between modes. This could be seen as a huge problem, but in fact it is a requisite for MIMO processing, which is used to deal with heavy crosstalk in multipath links. In conclusion, MIMO should be used where the transmission suffers heavy penalties due to high level crosstalk [13]. Another kind of mode coupling is polarization mode coupling, and refers to the coupling between two orthogonally polarized modes due to phase effects.

Recently there have been many different solutions to mode division multiplexing in optical fibers. Some very early papers introduced the idea ([17], [18]). The capacities and outage probabilities of these systems are evaluated in [15]. In particular, some of these solutions uses coherent optics and MIMO techniques (COMIMO) to achieve multimode link transmission ([4], [5]).

The current project model is based in a COMIMO solution, with coherent detection and MIMO processing. The capacity enhancement provided by MIMO in this situation has been reported in [21].

2.4.2 Mode division multiplexing

In order to achieve an optical MIMO link, the first requirement is to set up some mechanism to excite and receive the different modes or channels in the fiber and make use of Space Division Multiplexing (SDM). There are different ways to achieve SDM:

- Multi-core fibers. Every core represents a different channel input. An example can be seen in [22].
- Multi-fiber links, where big amounts of single-mode fibers are put together in the same physical link.
- Using different techniques and signal shaping at the receiver and transmitter to achieve mode division multiplexing on multimode fibers.

Regarding mode division multiplexing, one possible solution consists in using spatial filtering techniques (masks) [17], with spatially resolved equalizers that subtract the photocurrents [23]. Another solution is based on offset launching techniques [24] [25], where selecting the proper launch offset will selectively excite the desired mode. Some recent papers consider designing laser and detector arrays located within the fiber's core [26] or by using phase controllers and mode couplers with SMF [27]. A very common solution requires the use of phase plates ([5], [28], [29]), as shown in figure 2.28. Other possibilities include mode shaping with spatial light modulators [30] or using complex integrated optical devices.

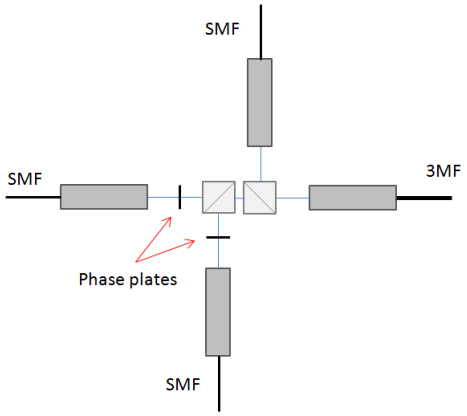

Figure 2.28: Example of a six-mode multiplexer for a Few-Mode fiber using phase plates.

For the current project, the way mode division multiplexing is achieved is not relevant and will not be considered. Mode splitting at the transmitter and at the receiver is assumed, allowing the use of N_T input mode signals and N_R output mode signals.

Mode-dependent loss and gain pose fundamental performance limitations in optical links and studies of their statistics and effects on mode-division multiplexing have been described [14], [16]. However, due to the increasing complexity of considering such effects in the current project, mode-dependent loss and gain effects have not been implemented.

2.4.3 Mode nomenclature

Regarding the description of the term ‘mode’ in this project, ‘mode’ will be considered any channel that allows independent communication propagated through optical fiber mode(s). The principal modes, defined by those modes not experiencing modal dispersion to first order frequency variations, are also referred as ‘modes’. In some situations, a group of modes with similar propagation properties (i.e. having nearly equal propagation constants) will fall into this description of ‘mode’. The use of different polarizations of the same mode(s) and spatial degrees of freedom can also be considered as different ‘modes’.

3
SYSTEM MODEL

In order to evaluate the performance of Multiple-Input Multiple-Output in multimode fibers, some simulation tool is required. This master thesis project has been focused on creating this simulation environment and verifying the capacities and possibilities of this method. The structure, behavior and implementation of the mathematical system model are described in this chapter.

The commercial software used to implement this project is Matlab®. The code functions that describe each one of the system blocks will be referenced here, but no actual code will be exposed. All the functions and scripts used in the system are found inside a ZIP file delivered with this thesis. The files also include more detailed explanations regarding inputs, outputs, algorithm flow and theoretical descriptions as comments within the code. These extra explanations are also part of the project description, but mostly offer specific programming details and thus will not be included in this chapter for practical reasons. Credits can found in the comments section of each file. The author only takes full credit for his files (marked with his name at the beginning).

The decision to use Matlab® for this project lies with the freedom it provides when designing your own model, allowing full control of the whole behavior of the system, and understanding all the theory and procedures behind every function. In other commercial developed software implementations some of the actual processing might remain hidden or not fully explained. In the particular scenario of multi-mode fibers, current software (like VPItransmissionmaker® or OptSim ModeSYS®) are being quickly updated to allow more precise and complete simulations considering the particular effects in this kind of fibers.

Most of the operations and procedures done by each of the components in the model have their theoretical background covered in the previous chapter. In this chapter the implemented solutions for the current project will be explained.

The formatting used in this chapter has specific rules. Words formatted with `this font` represent `functions` or `parameters` in the code, while *this font* is used for mathematical *expressions* and *results*.

3.1 System overview

The presented model follows the basic structure and includes all the components required in any communication system: transmitters, channel and receivers.

The transmitter block will generate N signal data (bit sequences) and will modulate them with an optical symbol modulator, feeding the channel with N mode division multiplexed optical signals.

A linear optical fiber model is used as the system transmission channel. The parameters that define it are: attenuation, mode delays, chromatic dispersion and mode coupling matrices. As stated in the theoretical overview chapter, some kind of Mode Division Multiplexing is required to be able to send every signal in every mode. In the current implementation, the multiplexing and demultiplexing processes are considered ideal, but their effects may also be accounted adjusting the mode coupling values of the fiber and group delays.

The receiver includes the optical amplifier (which will be the only considered noisy source), the optical (one) and electrical (N) filters, the coherent receivers, the MIMO equalizer (required for recovering the original signals) and the decision circuit.

To complete the system and make it valid for simulations, an extra error counting block is required to provide the total Bit Error Rate (BER) of the current transmissions.

Most of the parameters of the system can be controlled through an easy to use Graphical User Interface (GUI), allowing different setups and giving the option of many different output displays. The GUI is detailed in section 3.6.

A general overview of the model block structure can be seen in figure 3.1.

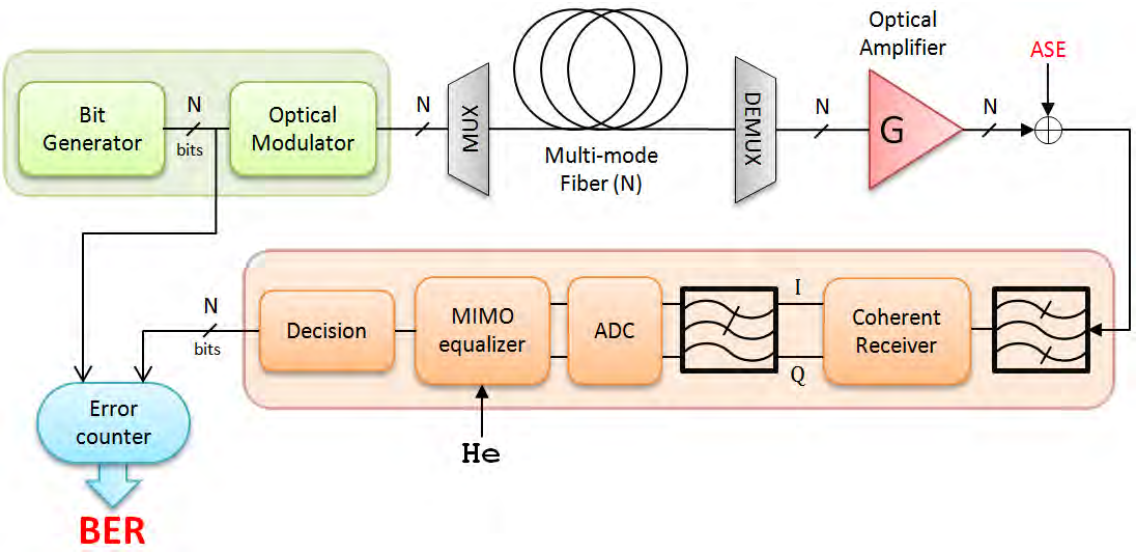

Figure 3.1: Block structure of the system model.

This general structure can be followed in the code with `simLink.m` file, which is the main function of the model. For every simulation, this function flows through the system as shown in figure 3.1.

3.1.1 Global settings

Function `simLink` defines some parameters that affect the whole system. In particular, the way signals are represented in Matlab®. Since working with computer software requires a digital environment, the analog signals simulated have to be sampled in the time domain by a higher sampling frequency than the digital signal symbol rate. This is what we call the sampling rate of the system (`Fs`), and it is proportional to the symbol rate of the system (`Rs`) by an amount we defined as samples per bit, or `sb`. The value of `sb` must be a power of two to avoid division errors. The default value chosen for this project is `sb=64`, and provides enough resolution (even with oversampling) without sacrificing processing speed and memory resources. We can see an example of a digital signal with this value in figure 3.2. Increasing this value will increase exponentially the length of the simulated signal vectors, thus limiting the maximum bit sequence length due to memory limitations and slowing down the simulations.

In reality, the working frequency values of all the optical signals are much higher than the symbol and sampling rates. All modern optical communication systems work at very high frequencies (hundreds of terahertz, the most common being at 193,12 THz). Due to implementation issues, we cannot sample these signals in Matlab® (it would require huge amounts of unnecessary samples). For this reason, all the signals in the optical domain have been moved to base band. This allows us to work with these signals as if they had symbol rate real frequencies, and sampling rate frequencies in our simulations are much shorter.

The time and frequency vectors for the current simulation are tied with the bit sequence length transmitted, which is the number of symbols `ns` times the samples per bit `sb`. The whole system will consider this transmitted sequence as a periodic signal. This way, the time vector will cover the whole length of the signal in portions of $1/Fs$. In similar fashion, the frequency vector will have as many samples as the time vector to allow Fourier transforms without padding or truncating the signal, and will have a resolution of Rs/ns.

The Fourier transform used to pass from time domain to frequency domain and vice versa is the commonly used Discrete Fourier Transform (DFT) and its inverse (IDFT). The Matlab® functions `fft` and `ifft` will be used for this purpose.

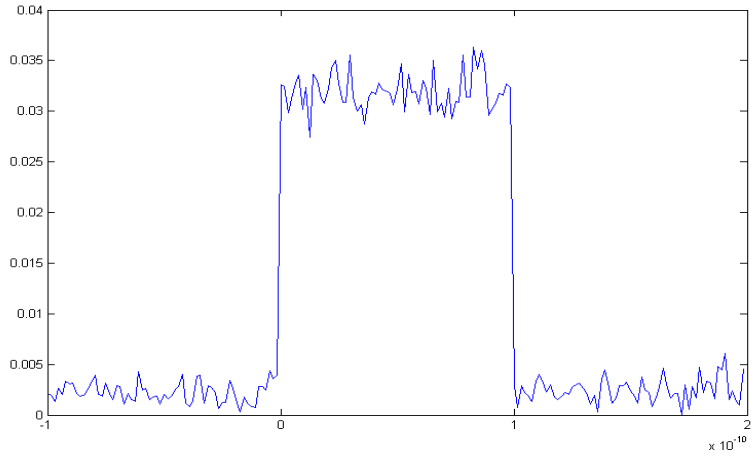

Figure 3.2: Representation of 3 noisy bits on a 10GHz signal with `sb=64` samples per bit.

The definitions of the Fourier transforms used in this project are:

DFT (M samples):

$$X(k) = \sum_{n=1}^{M} x(n) \cdot e^{-j \cdot 2\pi \cdot (k-1) \cdot (n-1)/M} \quad , 1 \leq k \leq M \tag{3.1}$$

IDFT (M samples):

$$x(n) = \frac{1}{M} \sum_{k=1}^{M} X(k) \cdot e^{j \cdot 2\pi \cdot (k-1) \cdot (n-1)/M} \quad , 1 \leq n \leq M \tag{3.2}$$

Another important value in the system is the `N` parameter. `N` refers to the amount of modes that are able to be transmitted and received independently (read section 2.4.3 in chapter 2 for disambiguation of the term ‘mode’).

`N` determines the size of the MIMO system to $N \times N$. The main reason to use symmetric MIMO is practical, because in this situation all channel matrices are square, with all the good qualities inherited from them. In case any real experimentation could not receive or transmit all the `N` signals, there is always the possibility to operate with an oversized system and then ignore/compose the unavailable modes. An example of an asymmetric optical MIMO system (2×4) can be found in [31].

Once these general rules and definitions are clear, we can explain the functions and procedures further for each one of the blocks composing the system.

3.2 Transmission block

The transmission block is composed by two structures: the bit sequence generator and the optical signal modulator. They are implemented with `BitGen.m` and `BitMod.m` functions respectively. Their function is to generate an input data bit string and modulate it as an optical signal that could be sent to the multimode fiber. Since we might be able to send `N` mode multiplexed signals through the fiber, this process must be repeated for each one of them.

The number of bits depends on the modulation format selected. As stated earlier, the number of symbols `ns` determines the length of the optical signal. The total amount of samples per signal will be $ns \cdot sb$. The number of bits will be `ns` times the number of bits per symbol of the current modulation. Only two modulations have been implemented for the current version of the project: OOK-NRZ and DQPSK-NRZ. For OOK modulation the number of symbols and bits are equal, while for DQPSK this amount is doubled. New modulation formats can be added in the current project by adjusting the new values on `BitMod`, `BitxSym`, and controlling the switch scenarios for `bxs` (bits/ symbol) and modulation `type` values during `simLink` and related functions.

3.2.1 Bit generator

In order to evaluate the performance of the system, an appropriate symbol sequence is required. A good symbol sequence will include all the possible patterns for a certain amount of bits. An easy and practical way to generate this kind of sequences is using Pseudo-Random Binary Sequences (PRBS) implemented with Linear Feedback Shift Registers (LFSR). The total length of the sequence equals to $(2^{nb} - 1)$, and contains all the nb-bit patterns (except nb consecutive zeros). For implementation reasons, we add an additional bit to this sequence so the length becomes pair (and power of two).

The LFSR used for the current implementation is generated with the `commsrc.pn` Matlab® function [32] (this function requires the Communication System Toolbox for Matlab®). On `BitGen` we create this generator by seeding it with a polynomial generator that will provide the maximum sequence length mentioned in the previous paragraph. The polynomial generator represents the seed of the nb registers and sets their initial state. To avoid having the same signal in all the modes, we set up a random offset to this sequence so the signals for the various modes that travel through the fiber will not be the same. More information on generator polynomials can be found in [33]. The current code of `BitGen` supports lengths from 16 to 32768 bits.

While the sequences generated are perfect for evaluating modulations like OOK, they are not adequate for multilevel modulations like DQPSK. The sequences required should contain all the ns-symbol patterns. Symbol sequences like de Bruijn can accomplish this, but require very long bit sequences. For this implementation only

LFSR sequences will be generated, and DQPSK modulator will use two randomly offset LFSR sequences to generate the symbols. This does not provide with all the ns-symbol patterns, but at least generates pseudo-random symbol sequences that allow evaluating eye diagrams and modulation constellations.

3.2.2 Modulator

Once the bit sequences for all the modes have been generated, we need to modulate them as an optical signal with the current modulation. This function is done by `BitGen`. It runs a digital to analog conversion from the bit sequences to an electrical signal with `sb` samples per bit. Depending on the modulation type, a Mach-Zehnder modulator (for OOK) or an IQ Mach-Zehnder modulator (for DQPSK) will be required to modulate the electrical signal to an optical one. In addition, a low-pass Gaussian filter is used to apply the desired rise time to the ideal electrical signals generated, as seen in function `RiseTimeSignal`. The rise time signal value is set by the user as a percentage of the symbol period of the signal.

Finally, the signals are scaled to the current power output level selected.

At the output of the transmitter block we have `N` optically modulated signals, with sample length $ns \cdot sb$, represented in the time domain, and ready to be propagated through the multimode fiber.

3.3 The multimode fiber

The multimode fiber block receives all the optical signals generated on the transmitter block and delivers them to the receiver block. While propagating through the fiber, the signals are subjected to dispersion, coupling and attenuation effects set by the parameters of the simulation. The corresponding function to this block is `MMF.m`. Even though the name states multimode fiber, it could as easily be named Few-mode fiber. These few-mode fibers typically support two to four modes with suitably tailored dispersive properties [34].

The fiber is modeled as a cascade of small sections (or segments), each one of them modeled as a random multiplication matrix and assumed to be statistically independent with each other. The length of these sections should be slightly longer than their correlation length in order to consider the propagation characteristics invariant [36]. The combination of all these sections describe the transformation for the entire fiber. This is an approach that has been used in numerous papers ([35], [26], [36]).

In our current implementation, only the coupling effects (defined by the coupling matrices) and the modal dispersion (defined by the mode group velocities) have been applied to this section scheme.

The way `MMF` process the signals follows the code structure shown in figure 3.3.

```
1) FFT
2) Chromatic Dispersion
3) For every segment
      a. Input Coupling Matrix
      b. Modal Dispersion
      c. Output Coupling Matrix
4) IFFT
5) Attenuation
```

Figure 3.3: Code structure for `MMF.m`.

Most of the multimode fiber model effects are applied on the frequency domain. This is basically caused by the application of the chromatic dispersion effect, which makes it easier to work with frequency components. Therefore, the first step of the algorithm converts the input mode signals to the frequency domain using the Discrete Fourier Transformation (equation 3.1).

3.3.1 Chromatic dispersion

The chromatic dispersion implemented in this model considers up to the third order of the propagation constant, and its full contribution (for the whole length `L`) is applied to the signal before any other effect. If a value of β_2 is known, the system will apply the corresponding pulse spread for the current simulation parameters on B2Calc. However, code allows calculating β_2 and β_3 approximated values based on the zero-dispersion wavelength (λ_0) and the corresponding dispersion slope, as seen in `DefB2Calc` and `DefB3Calc` functions. It is worth mentioning that the value of `lambda` in the code will not be used if none of these two functions are called, since the rest of the system model will be working on base band and is unaffected directly by its value.

With the β parameters calculated, we can create a frequency vector that will multiply each one of the input mode signal frequency components. The script file `GaussianTest.m` validates the correct Gaussian pulse broadening caused by the chromatic dispersion, thus validating the correct behavior of this effect.

3.3.2 Coupling and modal dispersion

Once the chromatic dispersion has been applied, the optical fiber is virtually separated in `Nseg` segments. For every segment we can define a matrix multiplication in the frequency domain involving the mode coupling and the group delays.

$$\boldsymbol{M}^{(\boldsymbol{k})}(\omega) = \boldsymbol{A}^{(\boldsymbol{k})} \cdot \boldsymbol{D}(\omega) \cdot \boldsymbol{B}^{(\boldsymbol{k})}, \quad k = 1,..Nseg \tag{3.3}$$

The final contribution of mode coupling and modal dispersion at the end of the fiber link will be the product of all these segment matrices. This structure is analog to the one used in [36]. It is also worth mentioning that with this structure, the coupling caused by imperfect mode multiplexing and demultiplexing could also be considered by adding its contributions on the first and last coupling matrices of the fiber.

The code requires the definition of two unitary $N \times N$ coupling matrices per segment, $\boldsymbol{A}$ for the input (prior to applying the modal delay) and $\boldsymbol{B}$ for the output of each segment. These matrices are input parameters of the fiber, allowing the system to incorporate premade random matrices if only one segment is considered. For instance, we could incorporate coupling matrix measurements obtained in a lab for a specific fiber. An example of how these values could be acquired can be seen in [37].

Inside the delivered ZIP file there are two predefined coupling matrices files. These `.mat` files have two matrices defined, named A and B, which are the ones used for the model. File `Hc.mat` contains two 3×3 deterministic strong complex coupling matrices, and can be used to fix the coupling effect for different simulations and different parameters, eliminating the coupling randomness on the obtained results. The file `nocoupling.mat` contains two identity 3×3 matrices, and simulates the perfect no coupling scenario, useful when the coupling effect needs to be removed entirely.

If no predefined coupling matrices are selected, the system will generate random unitary matrices at `simLink`. This randomness will generate any kind of possible coupling situation, and will prove the strength of the MIMO system for any possible coupling. Since this is a coherent system (see 3.4.2), coupling models that describe redistributions of power among the modes are ineffective due to not considering phase effects, and so the matrices generated with our system will have to include them. And last, the unitary requirement is needed to prevent adding/removing overall power of the signal due to coupling effects (since mode-dependent loss and gain are not considered).

To achieve this unitary random matrix generation we have used the invariant properties of the Haar measure. The whole process, motivation and detailed mathematical definitions can be found in [38], and only the basic generation procedure will be exposed here. Basically, Haar Measure allows us to create a random matrix where the unitary conditions imposed imply that matrix elements are statistically correlated. The generation process follows these steps:

1. Generate an arbitrary complex full rank $N \times N$ matrix, whose elements are complex standard normal random variables.
2. Apply an Orthogonal-triangular (QR) decomposition routine (in this project we used `qr` Matlab® function). The resulting Q matrix will be unitary, and R will be upper triangular.
3. Multiplying the matrix Q with a diagonal matrix based on the diagonal elements of R divided by their modules, the resulting matrix will be distributed with Haar measure.

With these instructions we are able to generate unitary random matrices that can simulate strong mode coupling effects on every section of the optical fiber, as required in the model we are using (based on [36]). The code for this generation function can be seen in `haar_measure.m`, and the only input required is the size `N`.

Matrix $\boldsymbol{D}(\omega)$ contains the group delays for every mode in the current segment, and is defined as a diagonal matrix where each element is defined as an exponential delay, being τ_i the uncoupled group delays of every mode in the current section.

$$\boldsymbol{D}(\omega) = diag\left[e^{-j\omega\tau_1}, e^{-j\omega\tau_2}, \dots\ e^{-j\omega\tau_N}\right] \tag{3.4}$$

We have assumed that all the sections will have the same uncoupled group delays. Therefore the matrix $\boldsymbol{D}(\omega)$ does not depend on the current section k, and all the group delays τ_i will be $1/Nseg$ portion of the total group delay for the current length.

Due to implementation reasons, all the group delays are applied as Differential Group Delays (`DGD`) relative to the fastest mode. The reason behind this decision is originated by the need to shorten the time signals at the output to reduce the digital processing required and the memory usage. The values of `DGD` will also limit the maximum length of the link depending on the current duration of the signal. We have imposed the restriction that no mode can be delayed longer than the current signal period being simulated. This is in order to avoid dealing with delayed signals that might come from previous simulated periods, which could lead to the inability to distinguish different group delays that are delayed by a multiple amount of simulation periods. This imposition effect can be seen in figure 3.4.

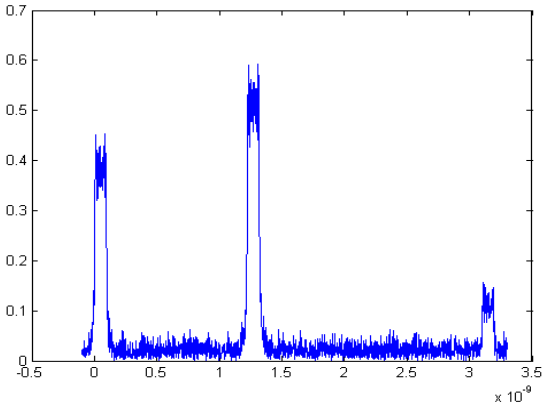

Figure 3.4: One of the $3x3$ channel impulse responses on a mode-coupling fiber (1 segment) with 32 bits delta @10GHz on a Max Length simulation. No coupled pulse is delayed outside the sequence period.

Using differential delays implies that the first mode (the fastest one) will have delay $\tau_i = 0$, and the slower modes will have the difference between the absolute delays of the current mode and the fastest. In order to calculate these `DGD`, `MMF` requires as

input the absolute Group Delays (`GD`). The way we have decided to incorporate this information into the model, considering the different sources that can be treated as 'modes', is feeding the system with the different group velocities of every mode. This will be input through an ASCII file (by default, `GroupVel.dat`) which includes a list of the group velocities of every mode (in meters per second). For obvious reasons, the length of the list needs to be equal or higher than the modes considered in the simulation. The group delays are then calculated for the current length of the fiber.

In all the simulations used on this project we have used a list of group of modes group velocities obtained with VPI® commercial software, by calculating the delay of different group modes in a simple simulation and dividing it over the simulation length. These values can be found in `GroupVel.dat` file. Special thanks to José Manuel Estarán for helping to obtain this data.

The matrix $\boldsymbol{M}^{(k)}(\omega)$ is not calculated as a whole. Instead, the signal applies the three effects multiplying matrix by matrix. This is mainly because the size of $\boldsymbol{D}(\omega)$ depends on the frequency vector ($f = 2\pi\omega$), and this blows the size of the matrix operations.

3.3.3 Attenuation

Once all the frequency dependant effects have been applied, `MMF` returns the signal to time domain with the Inverse Discrete Fourier Transformation (equation 3.2).

The attenuation is then applied as a matrix product. Since no mode-loss dependent effects have been considered, the matrix used for the current implementation (`Att`) only requires one value of attenuation (in dB/km), and all the input modes will be attenuated according to the current simulated link length.

3.4 Receiver block

The receiver block is the most important block of the system. Its purpose is to recover the original bit sequence for every mode from the strongly coupled and noise loaded signal that comes from the optical fiber and the amplification process. Within the receiver we find the optical amplifier, filters, the coherent receiver, an analog to digital converter, the MIMO equalizer and the decision circuit.

3.4.1 Optical amplifier

The optical amplifier is located right before the receiver (pre-amplifying setup). It is considered the only noise source of the system in our current implementation, and so the thermal and shot noises generated in the receiver can be neglected. This assumption can be considered when using high-gain amplifiers [39]. The amplified spontaneous

emission (ASE) complex noise generated has normal distribution, and is considered AWGN (additive white Gaussian noise).

The ASE noise variance of the signal in the modeled system can be defined as [1]:

$$\sigma_{ASE}^2 = \frac{N_0}{dT} \tag{3.5}$$

Where N_0 is the noise spectral density value and $1/dT$ (`Fs` in the code) represents the baseband frequency span used in the system. The relation between N_0 and OSNR is:

$$N_0 = \frac{\langle P \rangle}{\rho \cdot dT \cdot OSNR \cdot BW} \tag{3.6}$$

Where $\langle P \rangle$ is the average power of the signal, ρ is the number of polarizations used (for our model $\rho = 1$), and BW is the reference bandwidth for OSNR, defined by convention as 0.1 nm, which equals to 12.5 GHz bandwidth.

The function controlling the amplifier is `OptAmp.m`. The gain it provides comes from matrix `G`. In the current implementation, as we did with attenuation, mode-gain dependant effects have not been considered (no amplification figures have been considered). The gain matrix `G` will simply cancel the effect of the attenuation matrix `Att`, and its value is by default adjusted automatically to cancel the attenuation effect. The automatic refresh values function (`refresh_vals`) can be found in the function `MMFLink.m` (responsible of the graphical user interface).

The calculation of the current `ASE` level is done by `CalcOPn.m`. This noise level is defined by the value of the Optical Signal to Noise Ratio (OSNR), based on the average power of the current input signals (and output ones, once they are amplified). As stated in equation 3.6, the OSNR value can be related with the spectral noise density N_0, the OSNR bandwidth, and the sampling frequency of the system `Fs` that defines the sampling bandwidth. Once `ASE` value has been obtained, the function `addASENoise.m` will add a complex noise signal with the calculated power level.

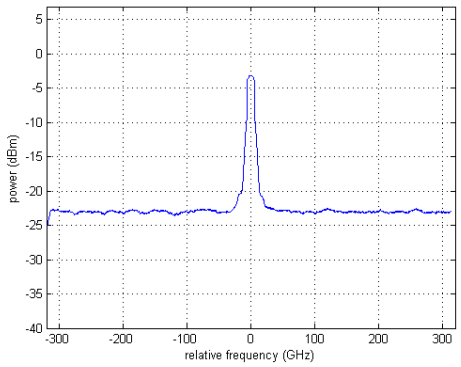

Figure 3.5: Spectrum analysis of a noisy signal with average power -3 dBm and OSNR 20 dB. The resolution bandwidth is the same as the OSNR Bandwidth (12.5GHz).

The reason of amplifying the optical signal before filtering is merely practical, and is known as pre-amplifying setup. Otherwise, the optical band pass-filter at the entrance of the receiver would not receive any noisy signal.

3.4.2 Coherent detection

The signal filtering, the coherent detection and the analog to digital conversion are all coded in the function `MIMORx.m`. As explained in chapter 2, the first step of the receiver is optically band-pass filter the signal to remove any frequency component outside the working bandwidth, and thus reducing the noise power. Since we work on base band, the centre frequency of the filter is $\omega_{ref} = 0$, so we are basically low-pass filtering. The other parameter required is the Full Width at Half Maximum (FWHM), referenced as `Fco` in the code, which by default is set to twice the symbol rate by the GUI. This bandwidth is converted to the equivalent electrical noise bandwidth (more details on `ElecOpt_NEB.m`). The filtering process is done by each of the `N` mode signals independently. The default optical filter has order 1 and is Gaussian type. A full list of available types can be seen in `Elec_ELPF_TF.m` function description.

Once filtered, we proceed to the ideal Coherent Detection. We have considered perfect homodyne coherent detection, with no need of carrier recovery and considering the local oscillator (at base band $\omega_{LO} = 0$) perfectly stable. We generate the In-phase (I) component as the real part of the incoming signal, and the Quadratic component (Q) as the imaginary one. Both signals are now in the electrical domain.

Both I and Q components are then low-pass filtered for every mode signal. The required electrical filter behaves in a similar way as the optical one, requiring cut frequency `Fce` instead. By default, the GUI sets this frequency value as the symbol rate. This filter works as an anti-aliasing filter, removing the alias located outside the working bandwidth. The default electrical filter has order 1 and is Gaussian type.

In order to digitally process the signal, we need to sample the filtered signal and quantify it. To implement this, an Analog to Digital converter is used, which will sample the I and Q components `sadc` times every symbol period. The value of `sadc` represents how many samples per symbol we capture. This oversampling function will be detailed in the MIMO equalization process in 3.4.3. No discrete quantification levels have been set for this converter (in other words, we have infinite quantification levels).

At the output of the coherent receiver we have obtained $2 \cdot N \cdot sadc$ digitally sampled signals ($sadc$ samples for every I and Q components for every mode), which are then fed to the MIMO equalizer.

3.4.3 MIMO equalization

The MIMO equalizer receives the digital signals of the modes and tries to compensate the MIMO channel effects.

When deciding how to implement the MIMO equalizer, the first solution attempted was building a $N \times N$ network of adaptive linear equalizers, like the one used in [28]. Each one of these filters of order L would then adaptive equalize the signals using Least Mean Square (LMS) algorithm. However, in order to set up an adaptive system, proper initialization of the equalizers and very long data-aided symbol sequences are needed (in [28] half a million symbols are used). In a practical situation, training sequences would be required (and sent as often as the optical channel might change its properties). Adaptive solutions are a great solution in reality thanks to their dynamic behavior. Different adaptive solutions were attempted, based on the options found in [12].

However, due to the complexity and long sequences required for building simulations with adaptive filtering, the current project will equalize the MIMO channel by estimating the channel impulse responses for every possible path between modes in the fiber, and using that estimation to recover the original input by inverting the effect of the channel. This channel estimation process is run by the MIMO training module, and will be explained in 3.4.4.

Once the estimation of the channel matrix has been acquired, and since all the effects applied in the fiber are lineal, there are different ways to recover the original signals. One of them requires the calculation of the inverse of this matrix, and then multiply the input signal vector with it.

$$\boldsymbol{y} = \boldsymbol{H} \cdot \boldsymbol{x} \quad \rightarrow \quad \hat{\boldsymbol{x}} = \boldsymbol{H^{-1}} \cdot \boldsymbol{y} \tag{3.7}$$

This solution has the inconvenience of the inverse calculation itself: even when we restricted the channel to be square ($N \times N$), the inversion algorithm might fail for nearly singular channel matrix estimations. To avoid this, we decided to use a powerful Matlab® tool named `mldivide` (or \) [40]. This function performs matrix division as if we were multiplying by the inverse, but without the need to calculate it. Since $\boldsymbol{H}$ is a square matrix, the operation $\hat{\boldsymbol{x}} = \boldsymbol{H} \backslash \boldsymbol{y}$ returns the solution to the equation computed by Gaussian elimination with partial pivoting [40]. This makes the system faster, but still has problems if the matrix is badly scaled or nearly singular. For the experiments run in this thesis, the use of `mldivide` has been proven enough to solve the linear equation generated by the MIMO channel in all the simulated scenarios.

The MIMO equalization is done in the frequency domain, and is done at a frequency rate defined by the oversampling of the system. The channel estimation matrix $\boldsymbol{H}$ has as many digital samples as the output $\boldsymbol{y}$, thus fixing the order of the equalization. However, the actual filter order can be lower (see 3.4.4 for further explanation).

Since the channel state information (CSI) is not provided to the transmitter, no feedback is required (which is usually a good think in optical links), and no MIMO pre-coding block is used to optimize the transmitted power of each input signal to reduce the noise effect at the outputs. The OSNR observed at the output of every mode will be

dependent on the mode coupling effects. Noise power is calculated by the average power of the output signal once it has been amplified, (see 3.4.1) As a consequence, some mode signals might be more degraded than the specified OSNR. This can be seen when selecting “Spectrum” plot in the GUI, and is detailed in section 3.6.

3.4.4 MIMO training

Although very effective ways to estimate the MIMO channel have been described in wireless communications [41], we have decided to estimate the channel response in our optical system by sending ‘deltas’ in time domain and calculating their channel impulse responses.

The MIMO training block cannot be seen on the general scheme shown in figure 3.1, but it is an indispensable block for the current “MIMO channel estimation through training sequence” implementation. The code for this block can be followed in `MIMOtrain_t.m` and `MIMOtrain_f.m` files. Both files attempt to train the system in two different ways. The training code should run as often as the optical channel might change. For optical communications this should happen much less often than in wireless situations, where effects like Rayleigh fading change the channel response very quick, and the training must happen very often (for instance, in GSM this is done twice every second). For our implementation we have assumed that the channel properties and its estimation will remain constant through the current simulation.

The first solution, `MIMOtrain_t`, bases the training process in the time domain, and can be summarized in figure 3.6.

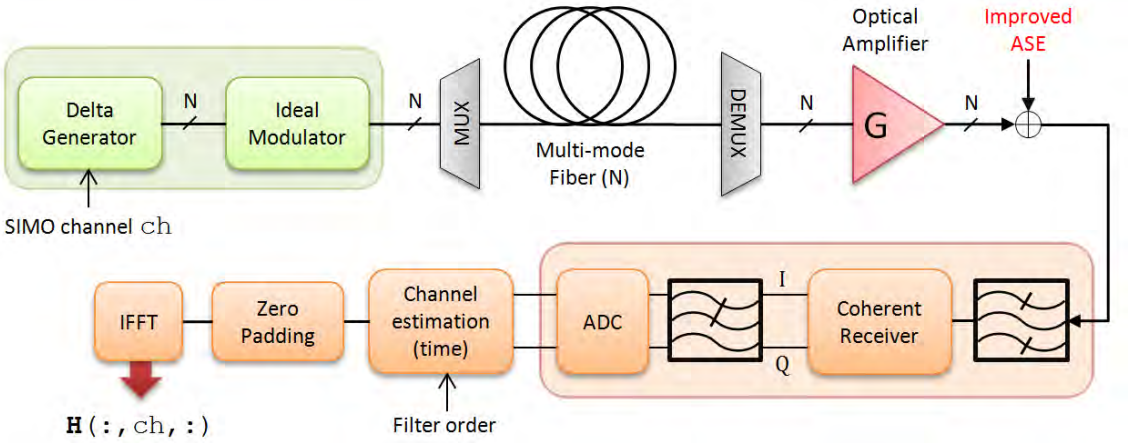

Figure 3.6: Structure of the MIMO training process. This process is repeated for every SIMO channel.

The process is divided into N Single-Input Multiple-Output (SIMO) scenarios. We accomplish this by sending one delta in the current SIMO channel (`ch`) and nil signals on the others. The deltas are modulated with an oversampled symbol frequency, and are as short as the minimum sampling period (set by the oversampling `sadc` and the

symbol rate `Rs`). The signal detected at the output of every mode will then be sampled with `sadc` samples per bit. Increasing this value will provide a better channel estimation (higher sampling rate), which will result in a more accurate equalization.

The real implementation difficulties for this process will not be considered. In a real world setup it would probably be much better to equalize the MIMO channel using adaptive digital filters. Examples of this implementations are exposed in [5], [28].

When using this SIMO structure, we can assure that the `N` output mode signals are the consequence of only the SIMO channel `ch` input signal. Since we have sent one delta, the received signals are the impulse channel responses of their respective channels $h(i, ch)$, where i is the current output mode and ch the current SIMO mode.

The noise is also affecting the estimation process. Since this effect can be critical (the error in an important response sample will affect all those where this sample is used to equalize), a noise improving system was designed.

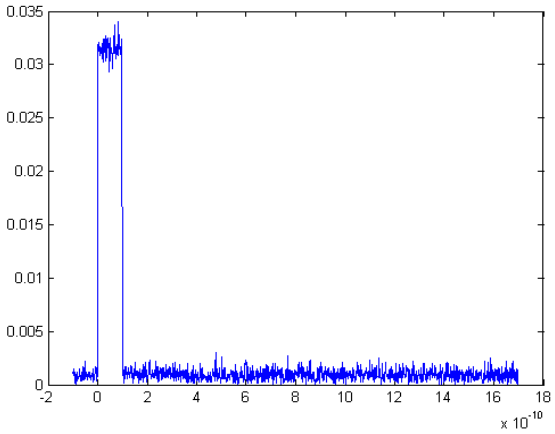

Figure 3.7: Training delta for a 10GHz signal (time in seconds).

Since the deltas are very short signals in time (one bit slot), the first noise improvement system works by repeating this training signals multiple times and averaging their outputs in order to decrease the noise effect (since the noise affecting the system has zero mean value). This can be seen in the code when setting `nT` higher than one. However, in the current implementation `nT` has been set to 1 by default (and not accessible by the GUI). Instead, another improvement system is being used: the noise has been improved by an amount equal to `impASE`. This parameter will decrease the effective ASE noise affecting the system (which is calculated from the average power of the signals that will be transmitted). This saves the necessity to run multiple training sequences (which slow down the simulations) while still lowering the noise effect on the training process. The value of `impASE` can be chosen by the user as long as it can be justified (i.e. with empirical results). No relation between this value and the amount of training sequences required to achieve same results have been studied in this project.

The time length of the channel estimation must be long enough to capture all the possible delays in the simulation. This minimum estimation duration can be obtained by calculating the worst possible delay in the current scenario (which happens if the signal always travels at the slowest transmitting mode). The function who calculates this minimum value is `refresh_vals` and can be seen in `MMFLink.m` file. This value will then set the minimum value of the filter order required to equalize the signal. The MIMO equalizer requires as many samples as the full sampled simulation signal period in order to perfectly recover the original signal. This maximum number of samples of the simulation period is the product of the number of samples per bit (the oversampling, `sadc`) and the current simulation symbol length (`ns`).

The equalization is done in the frequency domain, however the training is realized in the time domain because we can use zero-padding to fill the remaining samples once we are certain no more delayed signals will be found. If only the relevant time frame is simulated, the training sequences can be shorter, as well as the channel estimations. We can limit this duration by limiting the order of the filter. This provides two benefits: speeds up the training system and the overall simulations, and eliminates the noise that would affect the extra samples if no zero-padding is applied. These benefits will be increasingly worse when approaching the maximum length, requiring filter orders closer to the maximum order ($ns \cdot sadc$).

The oversampling is an attempt to increase the estimation resolution of the system. The resultant sampling frequency is a multiple of the symbol rate. Modal dispersion will translate the received deltas by amounts that are not multiple of the symbol period. This problem is shown in figure 3.8, where green deltas are the sampling instants.

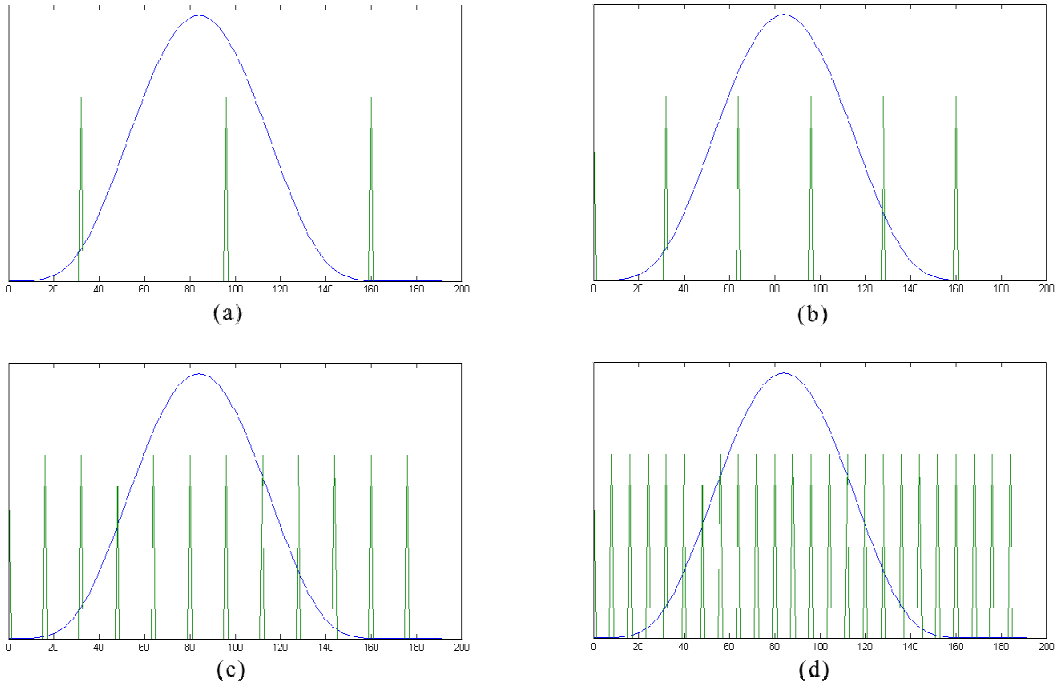

Figure 3.8: Training estimation samples when oversampling factor (sadc) is 1 (a), 2 (b), 4 (c) or 8 (d).

This means that for some scenarios with low oversampling the received pulse might fall between two decision instants, and thus failing to fully detect its magnitude. In other words, the oversampling frequency required to estimate this channel impulse responses should satisfy Nyquist criterion [1]. For multiple segments simulations, oversampling is even more needed to capture the channel responses with enough information to equalize the channel because the shape of the pulses disappear. This effect can be seen in experiment 4.8 next chapter. The required amount of oversampling is studied empirically in section 4.5.

Once the time estimations have been obtained, we transform them to the frequency domain, thus providing with the spectral behavior of the channels. These spectrums can be seen when selecting "Spectrum" output in the simulation (detailed in 3.6). The process is then repeated for the remaining SIMO systems, and $N \times N \times ns \cdot sadc$ full size MIMO channel matrix `He` is obtained for the MIMO equalization block.

The second solution, scripted in `MIMOtrain_f`, attempts to estimate the channel from the impulse response in the frequency domain. This was the first attempt on the training sequence, and it has been proven right for simpler systems (explained in experiment 4.3). However, the results obtained in its current version do not match the expected results and require further work and investigation.

3.4.5 Decision

The decision circuit implemented in this project is a simple threshold comparison. Depending on the oversampling, the decision circuit will choose the digital sample located in the middle of the symbol period and decide whether the signal is higher or lower than the defined threshold. The extra oversampled values are not used for the decision, and are only used to better recover the original signals in the MIMO equalization by providing higher resolution on the impulse responses.

A nice upgrade of the system would be introducing mode-dependent decision circuits, where every decision circuit can adjust the optimum sampling time (which is critical since mode dispersion affects this value by a large amount). This could be done by calculating the current optimum delay of the signal by calculating a fast correlation between the received and the training signal sent. This has not been implemented in this project and is left as a future work suggestion.

The thresholds applied in the decision depend on the modulation. For DQPSK, the threshold for both components is zero since the constellation is built so all the symbols have the same power (only phase changes). For OOK, the thresholds depend on the current OSNR. Even though some optimized OOK values have been calculated for some values of OSNR when validating the back to back transmission (see chapter 4), these are not used for normal simulations. Instead, the threshold for OOK is set to a non optimum level of half the average input power (as can be seen in `simLink.m`).

3.5 The error counter

The easiest and fastest way to compare the transmitted and the received bit sequences for each of the modes is calculating the XOR operation of the two strings. This task is done by `CalcErr.m` function, and also considers the demultiplexing used on the generated bits when there are more than one bit per symbol. This block also allows printing the position of the errors (details in 3.6).

The output of the block is the raw number of errors detected for the current sequence for every mode simulated. These values are then used to calculate the Bit Error Rate (BER) of each of the modes. The BER results for every channel will be printed in the command line, while the final averaged BER (or rather its logarithm) is shown in the GUI window, and it is considered the final qualification of the simulation. This means that the BER shown in the GUI is not the worst BER found in the system, but the average of the whole communication.

3.6 The Graphical User Interface

Since the complexity of the system and the amount of simulation parameters are quite abundant, the best way to run different simulations with different setups without the need to go deeper into the code and avoiding input errors is through a Graphical User Interface, or GUI. The project can be run by executing `MMFLink` on Matlab® command console (with all the files correctly located in the working directory). This will open the GUI and will allow any simulation to be run. Its task is to provide the best user friendly environment to change all the parameters required and set the different output options. Once the parameters have been set and the simulation is started, this function will adapt all the specified parameters into the `simLink.m` function.

This GUI has been designed with `guide`® [42], a graphical user interface design environment tool by Matlab®.

Most of the parameter values of the interface, like Maximum length, filter order and filter frequencies, will dynamically update when their depending parameters (like sequence length or symbol rate) are modified. This has been implemented to avoid setting up simulations with badly scaled parameters. This does not mean however that these values cannot be modified. For instance, when changing the symbol rate, both of the filter frequencies will adapt accordingly, but can be reset to a new desired value.

Figure 3.9 shows the main window of the user interface, where we can distinguish up to five different sections.

Figure 3.9: The simulation interface `MMFLink`.

The input panel includes all the parameters affecting the input signals: amount (thus fixing the MIMO system size), modulation, symbol rate, power, and their rise time.

The fiber parameters panel controls the way the multi-mode fiber behaves. We can specify the file where the mode group velocities are stored, the length of the link (limited by Max Length), the amount of segments to simulate (following the model we have implemented), and attenuation and chromatic dispersion values. We can also choose to use predefined coupling matrices by loading the appropriate file.

The receiver panel most important parameter is the OSNR of the output signals. Noise effects will be ignored if OSNR value is higher than 99 dB. We can also control the oversampling for the MIMO equalization and the MIMO equalization filter order, which is limited by default to the minimum amount of samples needed to sample the impulse response for the current simulation. The filter frequencies and the gain can also be set. Gain value will be automatically updated to cancel the attenuation caused by the current attenuation and fiber length parameters.

The simulation panel controls the amount of simulations, the amount of symbols per simulation, and the amount of maximum errors where we want the simulation to stop. This provides us a way to set up very large number of simulations and stop only when the desired amount of errors have been found. We can also control the training improvement explained in section 3.4.4, and various display options are available. We will discuss these display options in the next section (3.6.1).

And finally, the result panel will show the average Bit Error Rate for all the modes simulated with the current parameter setup. The result obtained here however must be treated carefully, since there might be modes where the BER result obtained (that can be seen in the command window output) could be significantly worse.

3.6.1 Display options

In order to display the result of our simulations we have many choices available. The most obvious one is the result tab in the GUI and the command line output generated by the system.

By default, the command line output (figure 3.10) is always enabled, and shows the simulation status and relevant results. Each of the system critical blocks also print the time spent in their processing, starting with the character "\". At the end of every simulation, a full summary with error results and time spent are shown.

If the option LOG file is chosen, the most relevant command outputs (system parameters, errors found, and final results) will be printed in a specially formatted text file (figure 3.11) and saved in the current working directory. The format for these file names can be seen in `FileName` variable in `simLink`.

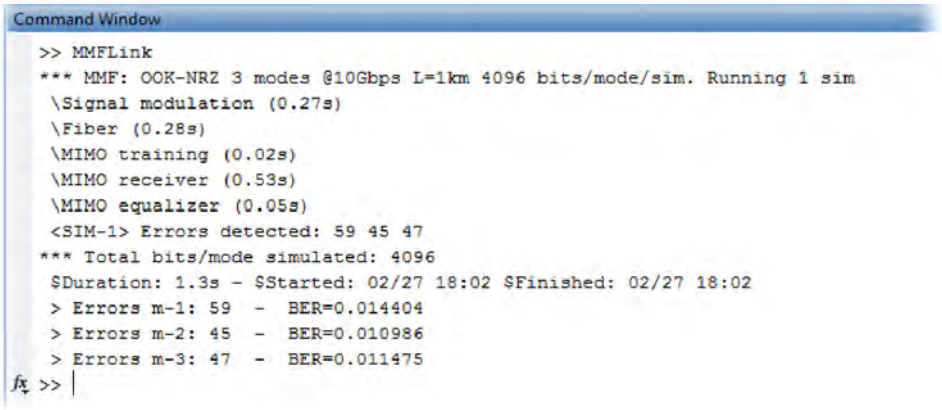

Figure 3.9: The command window output.

```
*** MMF: OOK-NRZ 3 modes @10Gbps L=1km 4096 bits/mode/sim. Running 10 sim
 <SIM-1> Errors detected: 2 0 0
 <SIM-2> Errors detected: 1 0 0
 <SIM-3> Errors detected: 0 1 0
 <SIM-5> Errors detected: 1 0 1
 <SIM-6> Errors detected: 0 0 1
 <SIM-8> Errors detected: 0 0 1
 <SIM-10> Errors detected: 0 0 1
*** Total bits/mode simulated: 40960
 $Duration: 14.1s - $Started: 02/27 18:05 $Finished: 02/27 18:05
 > Errors m-1: 4  -  BER=9.7656e-005
 > Errors m-2: 1  -  BER=2.4414e-005
 > Errors m-3: 4  -  BER=9.7656e-005
```

Figure 3.11: Log file example for 10 simulations with 14dB OSNR.

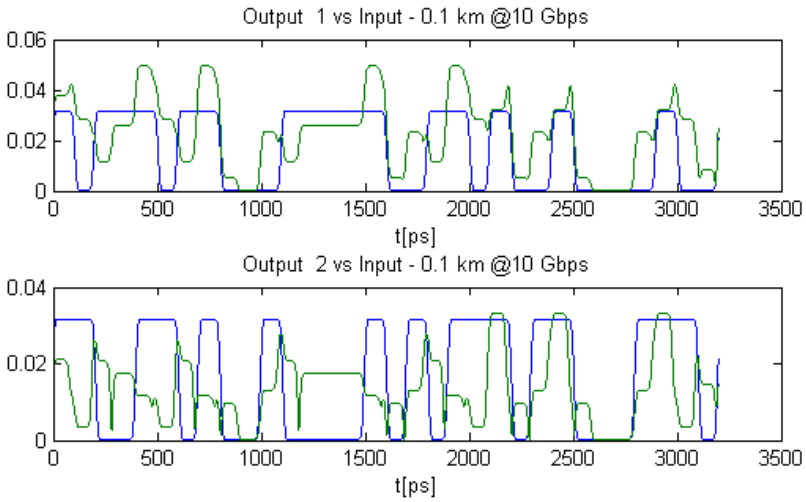

Figure 3.12: Fiber output for 100 meters link of a 2x2 coupled system.

The fiber output display (figure 3.12) plots the degradation of the signals caused only by the fiber effects (no noise). This is a nice tool to view the different effects of the fiber by carefully selecting the simulation parameters. For instance, we can remove the modal dispersion effect by setting the fiber length to zero, or we can remove the coupling effect in 3×3 systems when loading the `nocoupling.mat` predefined matrices.

The error location plot (figure 3.13) will show the exact position when this happens, and might be useful to check if the errors appear in bursts or are randomly spaced. This was used to verify there were no specific bits (like first or last) with higher error chances than others.

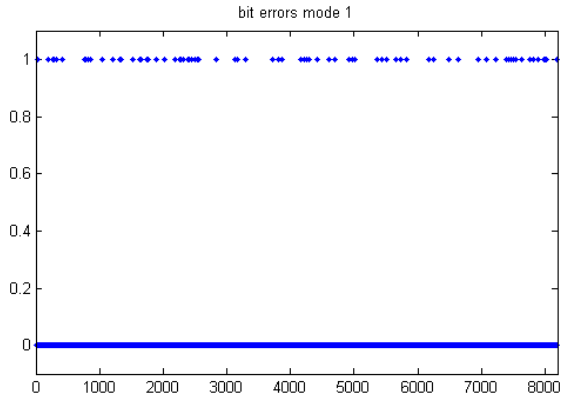

Figure 3.13: Error location plot for 8192 bits with BER=0.0097656.

The Spectrum option plots the optical spectrum analysis of one of the input signals (only one is relevant since the signals are the same, only shifted in time) and the output signals, allowing checking the defined OSNR value in the signal spectrum. The resolution bandwidth of these plots is the same as the OSNR bandwidth (12.5 GHz).

This option will also display the different channel estimation spectrums for the current MIMO system in a $N \times N$ grid plot. We can see an example in figures 3.14 and 3.15.

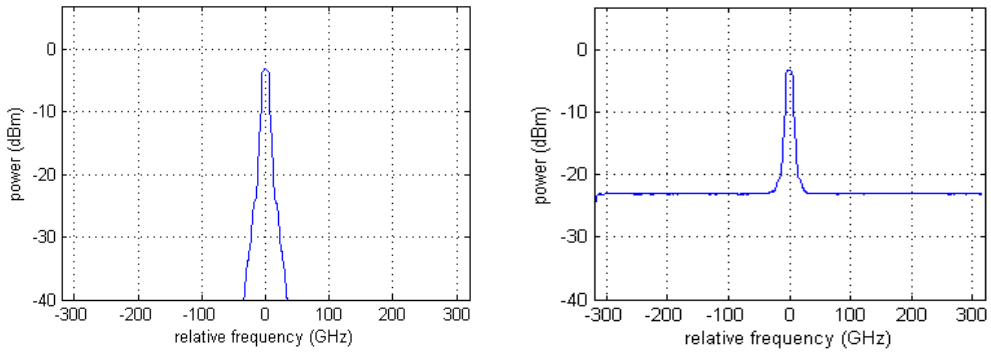

Figure 3.14: Input and output signal optical spectrum with OSNR=20dB.

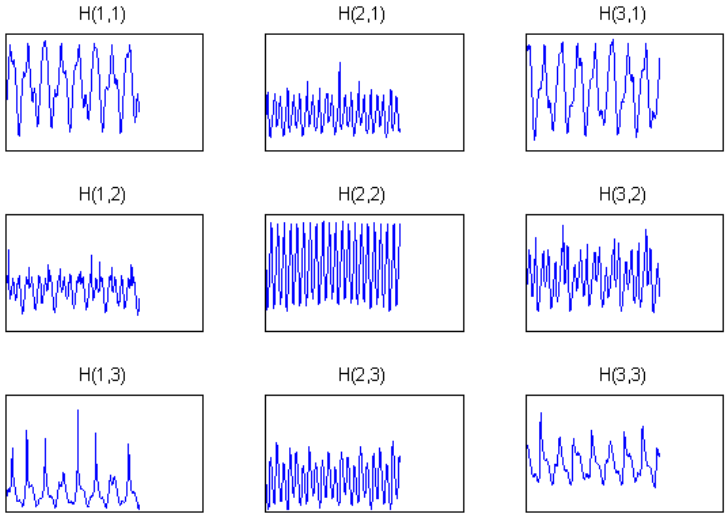

Figure 3.15: Example of 3×3 channel impulse response spectrums.

The signal evolution option is probably the most useful plot when simulating low amount of symbols. It displays for every mode signal the evolution in every step of the system: the original signal, the received one after the optical filter, the electrical signal before MIMO equalization, and the equalized MIMO signal compared with the original. When using OOK modulation the absolute value of the signal will be plot, and when using DQPSK only the phase signal will be shown. Figure 3.16 shows an example with OOK modulation.

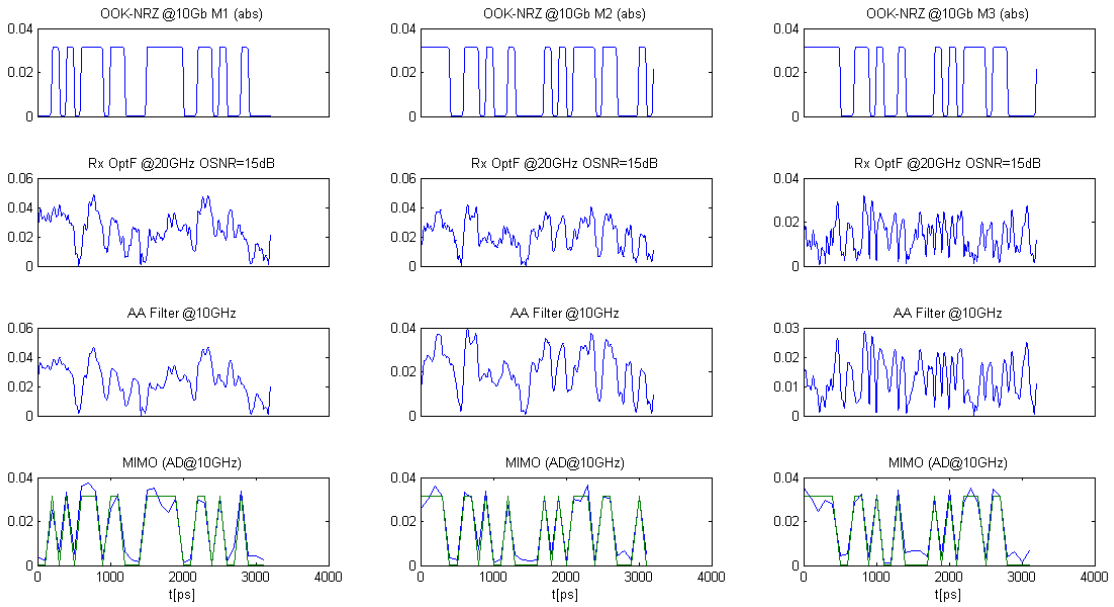

Figure 3.16: Signal evolution plot for 3×3 MIMO using OOK modulation.

The eye diagram plot displays the in-phase and quadrature components of the received signals. It works better for higher amounts of symbols. Figure 3.17 shows both OOK and DQPSK modulation eye diagrams for an OSNR of 15 dB.

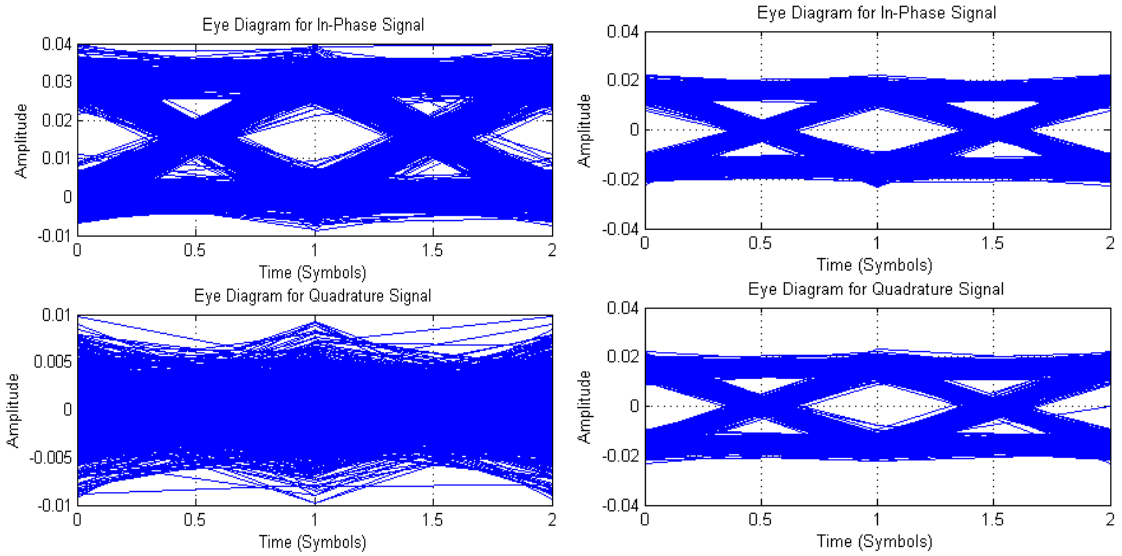

Figure 3.17: Eye diagrams for OOK (left) and DQPSK (right) with OSNR=15dB.

The constellation display (figure 3.18) will plot a grid containing the transmitted, received and equalized constellations for each one of the modes in the system. This provides very nice visual information to check the correct behavior of the system.

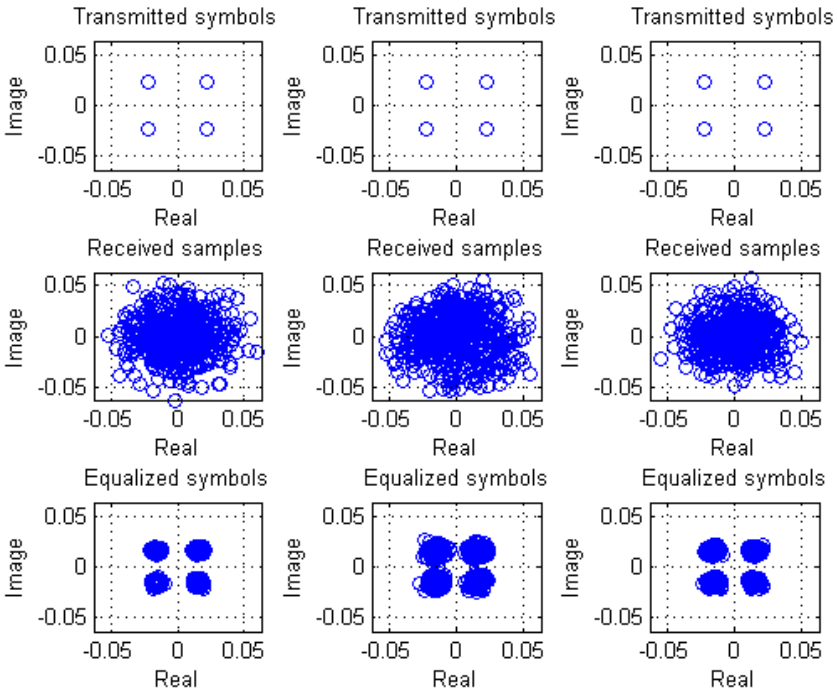

Figure 3.18: Constellation plot for DQPSK 3×3 MIMO after 3km @ 10GHz and OSNR=20dB.

And finally, the impulse response option will display the different channel estimations obtained after sending the delta in every SIMO system defined. Extra time periods (defined by the variable `T` in `MIMOtrain`) are displayed at the beginning and the end of the signal to provide extra representation margin. We can see one channel impulse response example for multiple segments fiber in figure 3.19. This display will be used on experiment 4.7 next chapter.

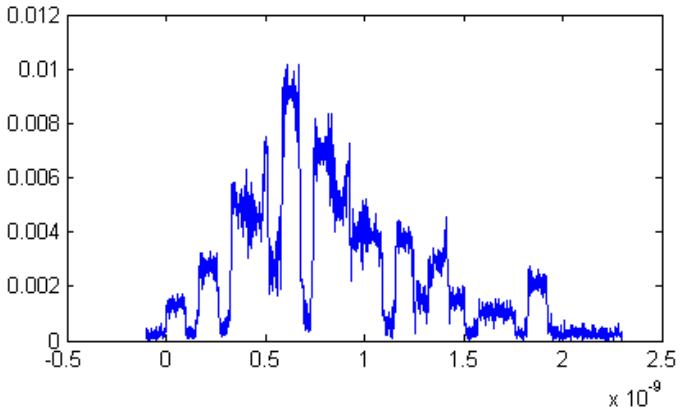

Figure 3.19: Impulse response plot on a 3-mode fiber at 10Ghz with 5 segments and OSNR=20dB.

3.6.2 Default values

For many experiments set up in next chapter, most of the parameters of the system are left by default to avoid listing the exact setup of the simulation. In this section, the default values of the GUI simulation tool are posted, and these are the values for the parameters in the simulations unless specified otherwise.

Parameter	**Value**	**Parameter**	**Value**
Lambda [nm]	1550	Mode velocities file	GroupVel.dat
MIMO size [NxN]	3	Fiber length [km]	0
Symbol Rate [Gbps]	10	# segments	1
Rise time [%]	25	Attenuation [dB/km]	0.28
Modulation	OOK-NRZ	Chromatic Disp. [ps/nm/km]	15
Power [dBm]	0	Predefined coupling matrices	no
Bit sequence length	32	Oversampling	1
# sims	1	OSNR [dB]	10
Max Errors	1000	FWHM Optical filter	20
Training improvement [dB]	10	F3dB Electrical filter	10
Display Options	(none)	Limit MIMO filter order	yes

4
SYSTEM SIMULATIONS

Once the simulation tool has been designed, system verifications are required to validate the model, and once verified, it can be used for simulations. In this chapter, all the validations and some experiments are detailed, and their results evaluated.

First we will validate the back to back transmission and reception system with direct detection setup. Then we will prove the correct behavior of our noise source depending on the optical signal to noise ratio (OSNR). To prove the MIMO system equalization, we will provide an equalization example with modal dispersion effects, and considering the different oversampling factors. Afterwards, the effect of the symbol rate and MIMO size on the final Bit Error Rate (BER) results is analyzed. And last, some real scenario approximation simulations are run to investigate its performance.

All the simulations have their scripts and code functions inside the ZIP file delivered, and can be run again to obtain the results exposed here. Some of the experiments are set up by running the default GUI of the model, `MMFLink`. The simulation possibilities of the presented tool are as large as all the possible combinations and setups imaginable.

It is important to note that the BER values obtained in the latter experiments should be read carefully. For OOK modulations the decision thresholds are not optimized for their respective noise levels. For DQPSK the sequences generated at the transmitter do not contain all the desired symbol patterns for bit error rate analysis. In addition, most of these simulations capture relatively low amounts of errors (from hundreds to thousands). Therefore, all BER values posted here should be considered with an inherited deviation from these effects. This deviation has been calculated from statistical BER results, and its estimated value is around ±0.1 dB. Nevertheless, the values obtained allow us to see the evolution of the system when changing its parameters, and also to provide an insight of what the system can achieve.

It is also worth mentioning that, for all the simulations exposed here, the error distribution among the channels follows normal distribution. No specific channel show higher statistical error count than any other channel of the MIMO system. This can be seen by looking at the command output of the simulations, or at the LOG files if such option has been chosen.

4.1 Noise validation

In order to count errors in a system like this, some sort of randomness source is required in the system. For this project, this source is the ASE noise that comes from the optical amplifier.

The best way to check if the produced noise has the specified optical signal to noise ratio (OSNR) is to observe the frequency spectrum of the signal. We use the function `Meas_Opt_OSA.m` to plot this spectrum, setting the resolution bandwidth 12.5 GHz, which is the one defined for OSNR.

The validation function `OSATest.m` can be modified with the desired peak power `P`, bit rate `Rb` and `OSNRdB` values to obtain the desired verification.

Figure 4.1 displays the spectrum analysis of a 10 GHz signal with 10 dBm average power and OSNR of 30 dB. We can perfectly see that the noise level sits 30 dB below this average.

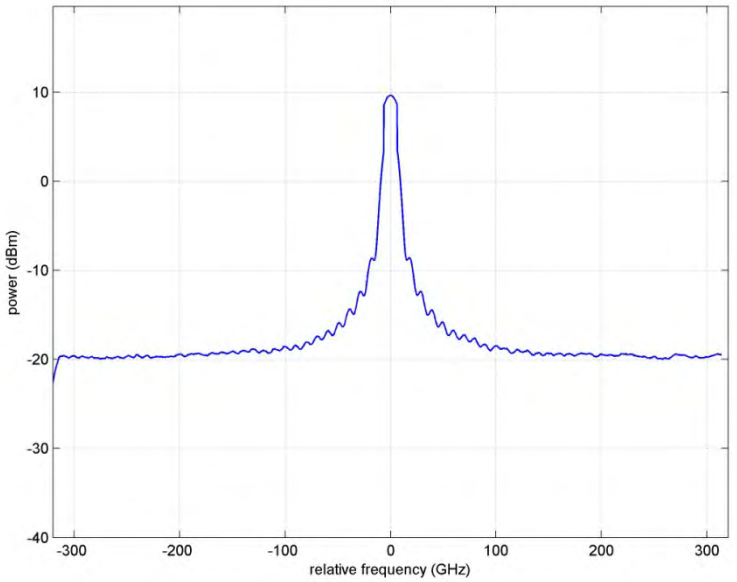

Figure 4.1: Optical Spectrum Analysis for 10GHz signal with OSNR=30dB.

Now that the noise source is validated, we can assume all the OSNR values are correct for the rest of the simulations.

4.2 Back to back system validation

The first step to validate the system before applying any effect is to check if the transmission and receiving processes are working as expected, and that the error counting algorithm is correct. To do this, the results exposed in [43] will be used to compare the performance of our system. In particular, we will use the values from figure 5, where the performance for non dispersive fibers with Gaussian filters is plot. In our scenario, no fiber is used, and the receiver works with direct detection scheme, order 1 Gaussian optical and electrical filters, a simple ideal photodiode, analog to digital conversion and the same decision circuit explained in section 3.4.5. The filter frequencies are set to match the ones in [43]. The performance curves are calculated with On-Off-Keying modulation with no return to zero.

The script corresponding to this experiment is `OOKBERT.m`, and every simulated scenario is run by the `b2btest.m` code function.

Each value of the Bit Error Rate is related to certain bit energy per noise power level (E_b/N_0), which can be related with the OSNR used in our system with an easy conversion calculated in `EbN02OSNR.m` function. This will allow us to simulate with the equivalent OSNR values for every E_b/N_0 in the plot. These plot values can be seen on `Forestieri_Gauss1_8B_Gauss1_0.7B.txt` file, and are obtained from figure 5 in [43].

The most important parameters for this verification are the threshold levels for every OSNR simulated. OOK optimum thresholds depend on the current noise level, and thus static threshold levels will provide worse results. For this reason, the script `optTH.m` has been designed to empirically sweep between threshold levels for every OSNR level, and choose the threshold level which provides the lower bit error ratio. This script saves the optimum levels (normalized to the input average power) in the text file `oth.txt`. These are the values used for the verification. Figure 4.2 shows its first values.

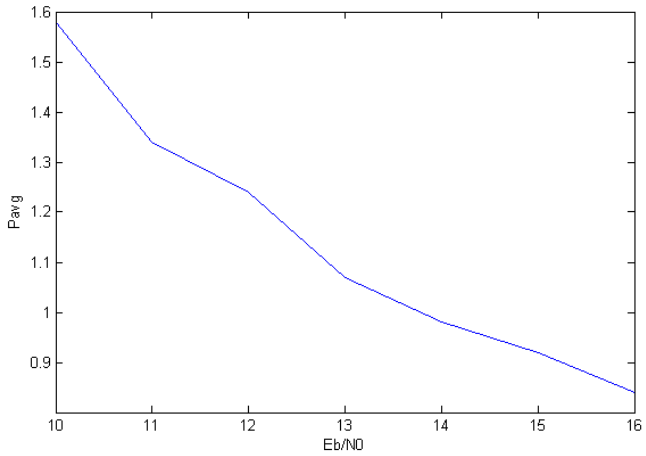

Figure 4.2: First seven values of the optimum threshold values for the OOK back to back test.

Notice that simulating this verification, as well as obtaining the optimum threshold levels, might take considerable amount of days. The simulation can be limited by setting `Nlim` lower (7 seems reasonable) and the threshold sweep range in `optTH` shorter (and adapted to every noise level). However, the time decision instants have not been optimized, and will decide at half bit period.

Figure 4.3 plot has been obtained after running OOKBERT with `Nlim=7` and optimized threshold levels from `oth.txt` up to E_b/N_0=16.

From the results obtained in this figure (can be found as JPEG file in the CD with the name `OOK_BERT_plot.jpg`), we can see that both lines are quite close, but green line (the simulated one) is slightly above the curve from [43]. This is to be expected since blue curve represents the best results possible in these conditions. The difference between both curves can be minimized by further increasing the resolution calculation of the optimum threshold levels (which in `optTH.m` is set to hundredth of the average input power) and optimizing the decision time.

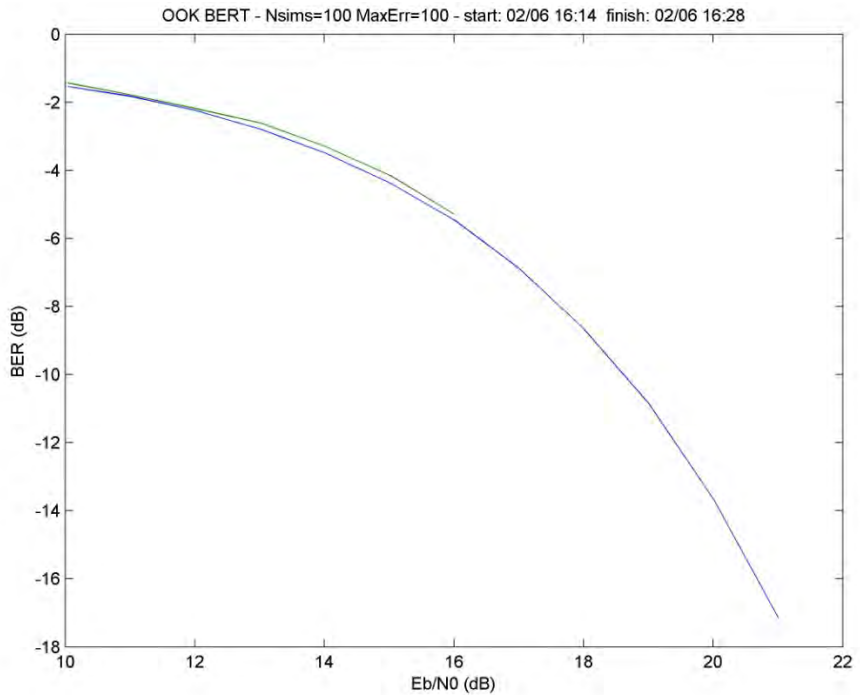

Figure 4.3: Back to back system verification.

After evaluating these results, we have concluded that the error counting algorithm of our system and the transmission and reception structures used are working as expected.

4.3 Frequency equalization test

For the implementation of our MIMO model we required an equalizer to solve the MIMO system. Before implementing the training required and the MIMO equalization process to the full model, a 'mini-model' was built. The code is found in `Feqtest.m` file, and has many similarities with `simLink.m` structure (the full model which is based upon).

This mini-model is a 3×3 MIMO system at 1 GHz, and simulates OOK sequences of 32 bits affected by modal dispersion effects (loading default calculated values from `GroupVel.dat`). No noise, chromatic dispersion or coupling effects are considered. The goal with this model is to verify that frequency channel estimation is possible while using delta sequences as training sequences to obtain the channel impulse responses.

The signal generation, the MIMO training, the modal dispersion effect and the equalization are performed. The output plot in figure 4.4 shows 3 columns, where first row represents the original and dispersed signals and second row shows the equalized signal. We can see that the equalized and the original signal are exactly the same under these circumstances.

This was the desired approach when designing the MIMO training block in the model, but after several attempts the results obtained in the current implementation were not satisfactory enough, and a time training method (already explained in 3.4.4) has been used.

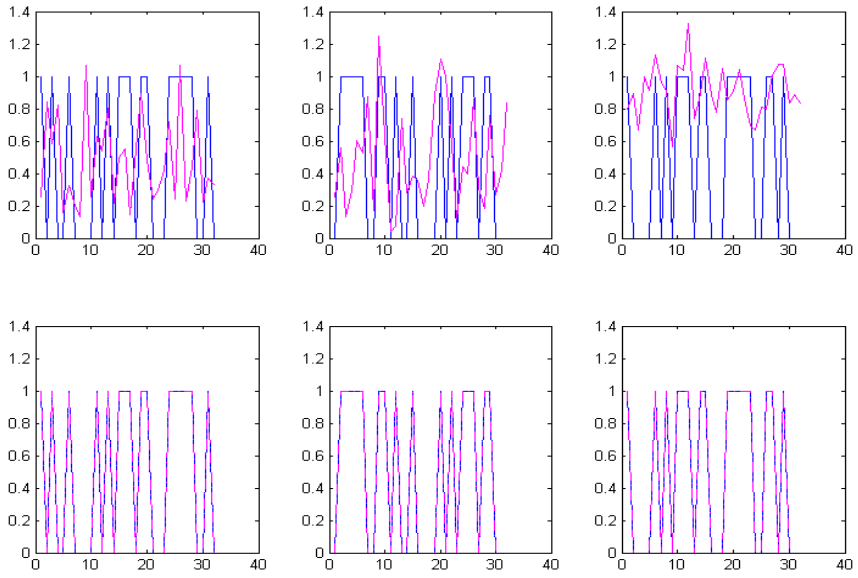

Figure 4.4: 3 × 3 MIMO frequency training and equalization process for 32 bits.

The simulations exposed from now on will use the full implemented model (`simLink.m`) with different simulation parameters.

4.4 Back to back system performance

This simulation is designed to see the overall performance of the final model (with coherent reception and MIMO equalization. Since fiber length is zero, modal and chromatic dispersions are not considered, and only noise and mode coupling effects are applied in the MIMO system.

The scripts `OOKsimBERT.m` and `DQPSKsimBERT.m` will plot the bit error rate curves for the selected OSNR range for each one of the modulations. By default they simulate 3×3 coupled 10 GHz signals with OSNR 10 dB. All their parameters can be adjusted by changing their values in the code. The script `simBERT.m` will simulate both modulations at the same time for the same ranges.

In figure 4.5 the BER results for OOK are shown. The OSNR range selected for this modulation plot goes from 8 to 16 dB with steps of 0.5 dB. The latter value has produced no errors for this simulation size.

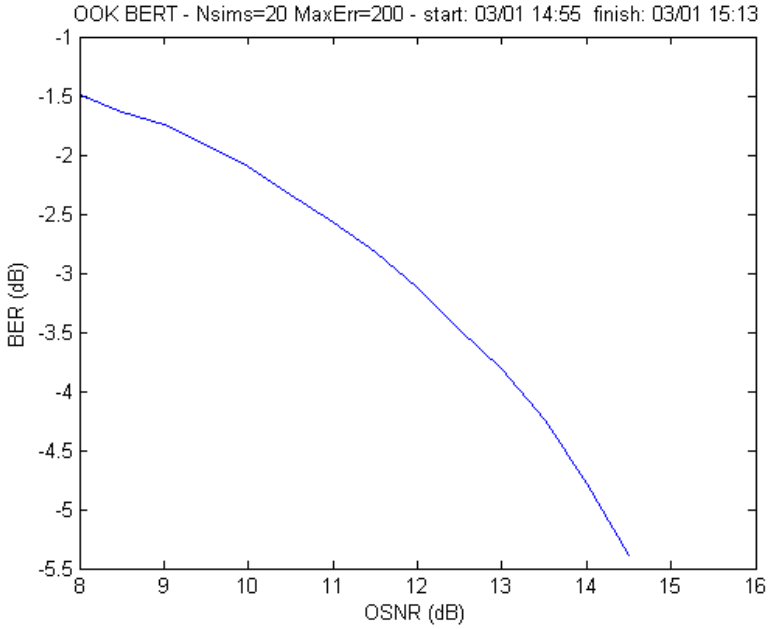

Figure 4.5: Bit Error Rate test for OOK modulation

Figure 4.6 shows the BER results for the DQPSK modulation. The OSNR range here is shorter, from 5 to 10 dB in steps of 0.5dB, because DQPSK has better performance than OOK [45], and higher values of OSNR require bigger simulation sizes in order to obtain enough errors and provide valid statistic results. This better performance is also increased due to the fact that DQPSK does not require optimized thresholds (which is zero in all cases), while OOK requires optimization to achieve the best performance.

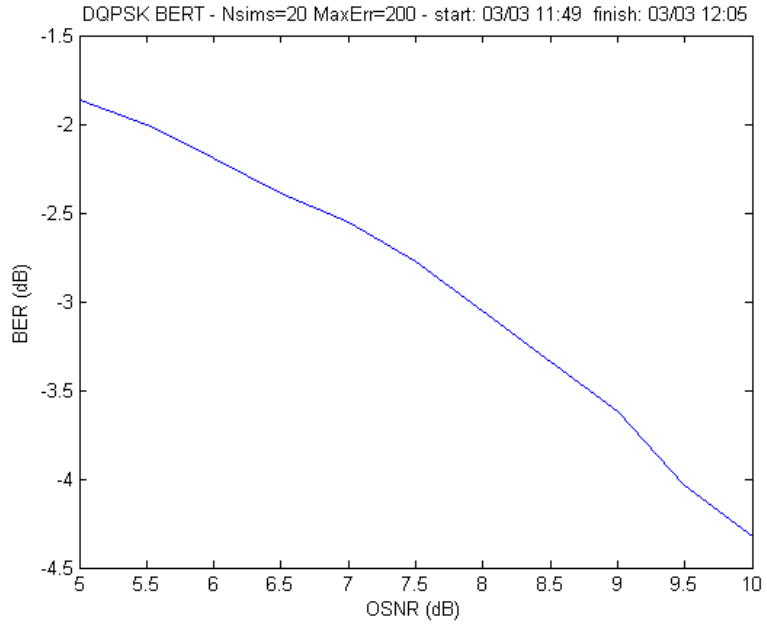

Figure 4.6: Bit Error Rate test for DQPSK modulation.

Figure 4.7 shows the result of plotting both signals with `simBERT`. The BER values obtained inferior than -4,5 dB are inaccurate (they are calculated with low amount of errors). In order to increase the resolution of this curve it is required bigger number of simulations, increasing the computational time.

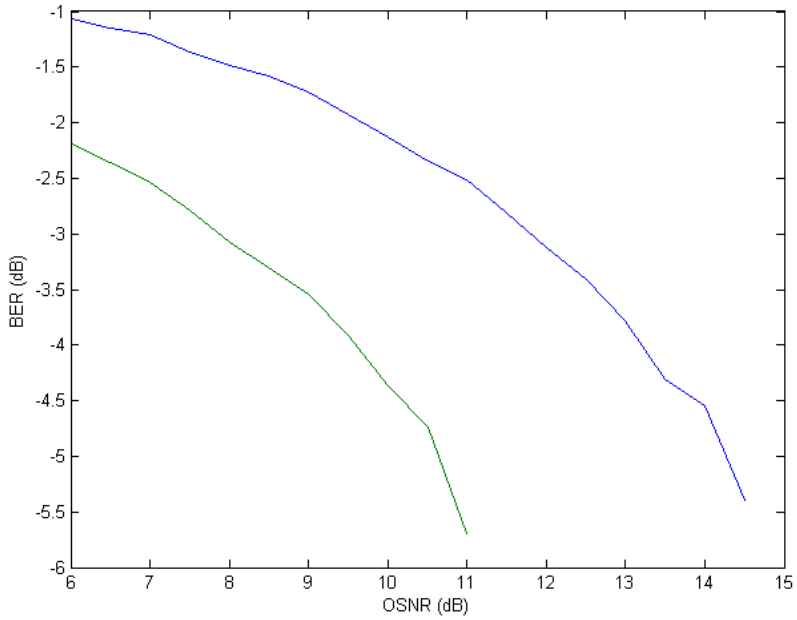

Figure 4.7: Bit Error Rate test comparison for OOK and DQPSK modulations.

4.5 Mode delays for multiple segments

The mathematical model of the multimode fiber implemented in this system is based on a cascade of coupling segments as explained in section 3.3.2. This experiment reproduces the scenario exposed in [36], where the statistics of group delays in multimode fibers with strong coupling are evaluated, and shares the same structure than the project's model.

The idea of this experiment is to see the effect in the simulation of the number of segments considered. When more sections are simulated, more homogeneous group delay distributions between all the modes are observed. This can be considered a positive effect: the overall delay is more predictable, and so it requires more concentrated predictions when training the channel (since the overall spread length is shorter), thus lowering the amount of filter taps required. It also causes synchronized arrivals of all the modes at the output, since the original mode where signals were introduced no longer determines their speed along the fiber, but only for the first segment.

To reproduce this effect, we have used `MMFLink` interface to set a 3×3 MIMO system with 10 Gbaud 30dBm signals (64 symbols) on a 2 km fiber with OSNR 10 dB. No oversampling is used for this experiment ($sadc = 1$). Each segment simulated will apply a random coupling effect (at the input and the output) with the proportional group delay for the current segment. The rest of the parameters are left by default (see section 3.6.2). The overall group delay values for each one of the channels can be seen when choosing the impulse response output.

Figures 4.8 to 4.16 show examples of the obtained plots at the output for the different amount of segments. The y-axis represents the amplitude of the OOK signal (taken as the square root of the signal peak power, which in this case is 30 dBm), while the x-axis are picoseconds (ps).

When looking at the evolution of the group delays, we can appreciate the sharpening of the received signals when increasing the number of segments. This segment number increase should be related with the link length (longer lengths should simulate bigger amount of segments). If we compare figures 4.8 and 4.16, we can see the transformation after 150 segments: when 1 segment was considered the deltas could be easily distinguished, and the overall signal had the widest time span; but after 150 segments, the deltas are no longer distinguishable, and the information of the signal is compressed in time to around 40% of the previous time window required.

In conclusion, these observations agree with the predicted results exposed in [36] and are in concordance with other published papers [35].

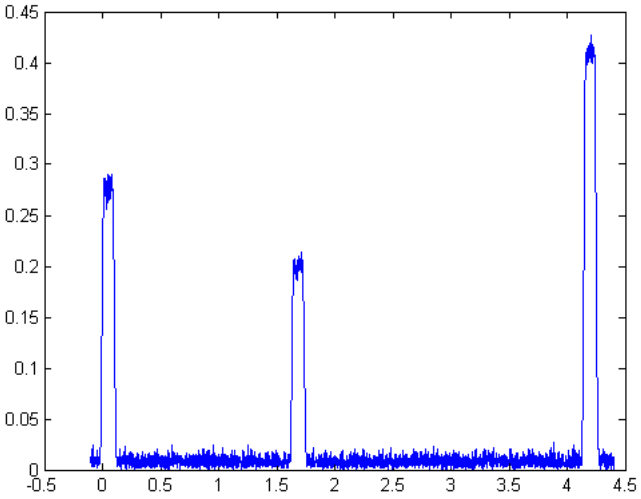

Figure 4.8: Group delays for 1 segment.

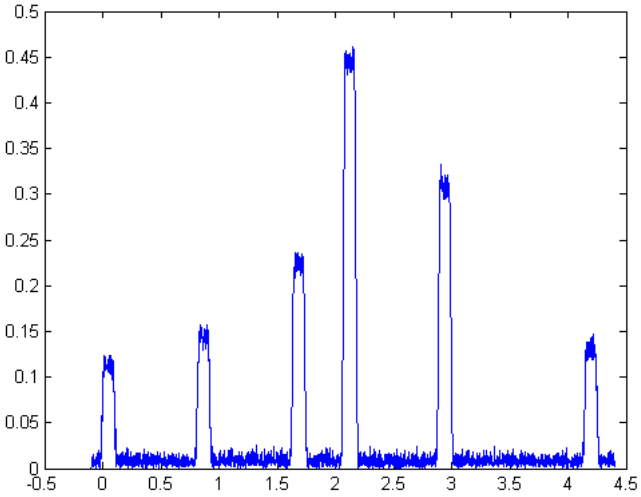

Figure 4.9: Group delays for 2 segments.

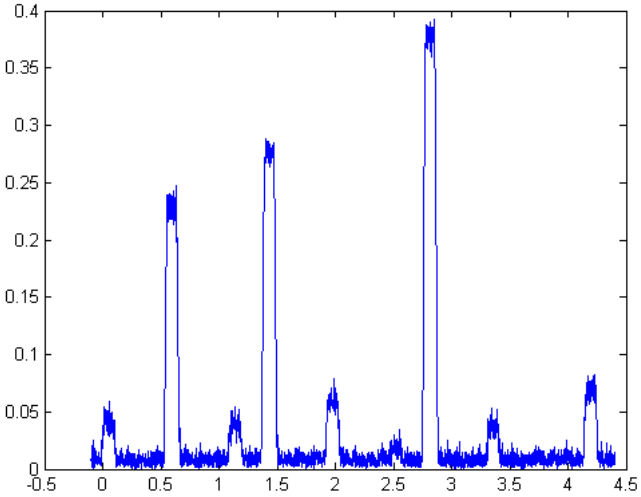

Figure 4.10: Group delays for 3 segments.

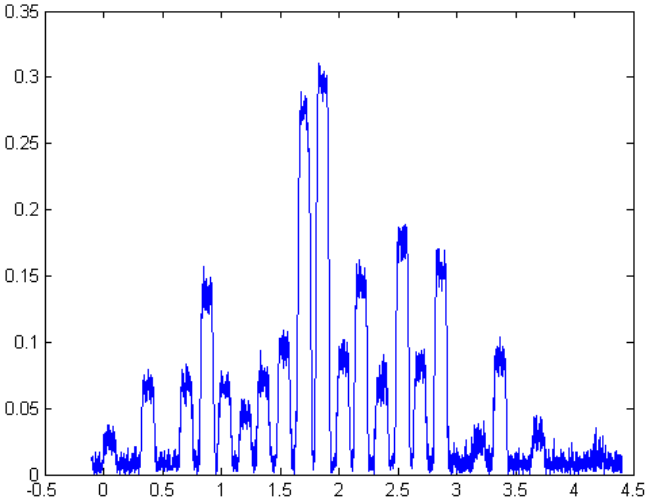

Figure 4.11: Group delays for 5 segments.

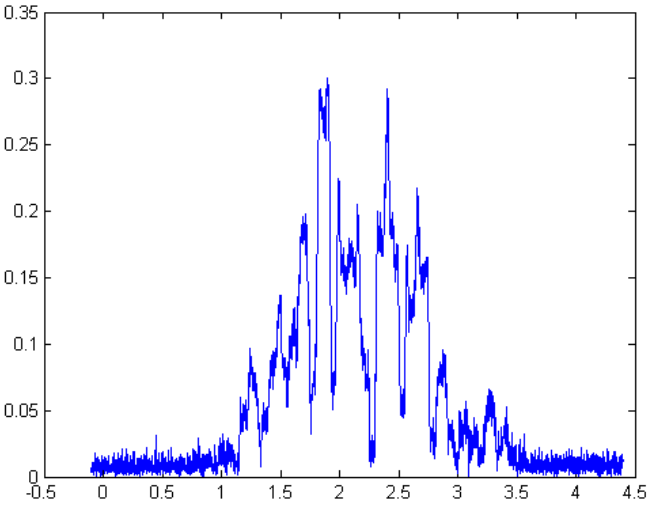

Figure 4.12: Group delays for 10 segments.

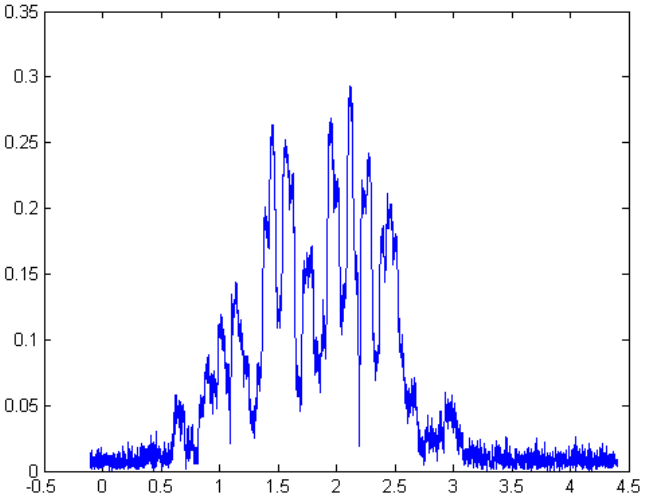

Figure 4.13: Group delays for 15 segments.

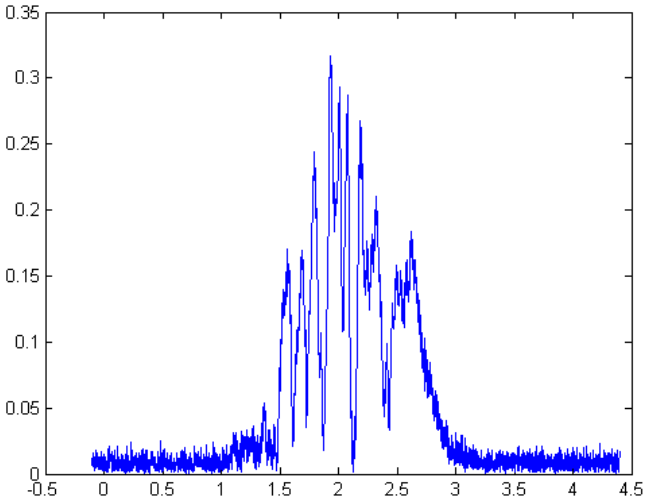

Figure 4.14: Group delays for 50 segments.

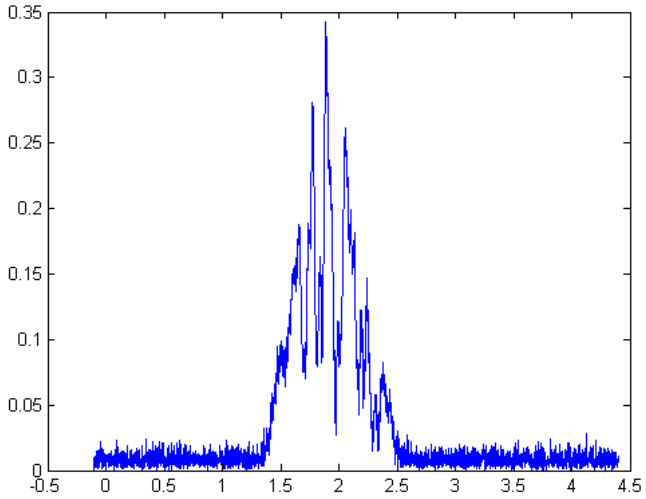

Figure 4.15: Group delays for 100 segments.

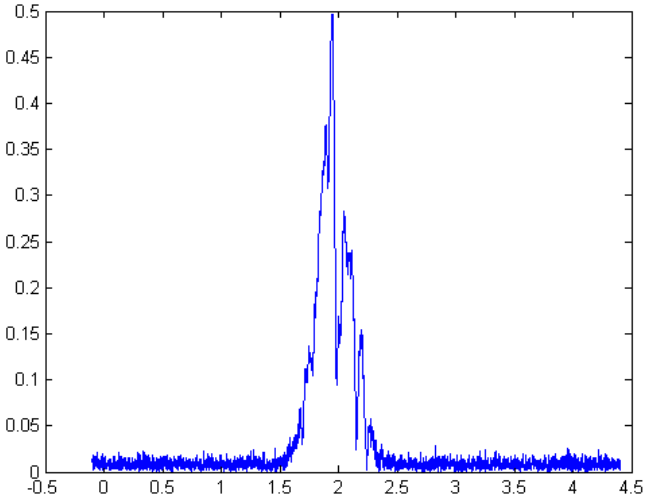

Figure 4.16: Group delays for 150 segments.

4.6 Link length effect and oversampling

This experiment evaluates the degradation of the calculated BER for increasing multimode fiber lengths at different oversampling rates of the training impulse response. The idea behind this experiment is trying to find out how many samples per symbol should be enough to estimate the channel impulse response in the time domain for any length and group delays. `MMFLink` will be used to run this simulation.

The experiment setup consists of a 3×3 MIMO system for three 10 GHz signals with rise time of 25 ps (25%), sequence length of 16384 symbols each, and an OSNR of 10 dB. It must be remembered that the OSNR is the noise ratio calculated from the average output signal power (once amplified – see 3.4.1). Both OOK and DQPSK modulations have been simulated. Max Errors is set to stop the simulations once 200 errors for OOK or 100 errors for DQPSK have been reached in one mode. The rest of the parameters (filter frequencies, training improvement, chromatic dispersion, etc) are left with default values (see section 3.6.2).

On the first part of the experiment, the link performance is evaluated with OOK modulation every 5 meters, up to a maximum of 200 meters. To reduce the randomness caused by the mode coupling effect on these simulations, only one segment with premade coupling matrices `Hc.mat` are loaded for all the simulations. For the current group velocities simulated (default ones from `GroupVel.dat`) at 10 GHz, 200 meters allow to see the transition when slower received pulses fall between different time-slots, thus causing low-oversampled simulations to be inefficient to detect these variations. We can see an example of this effect in figures 4.17 and 4.18 for 5 and 25 meters respectively.

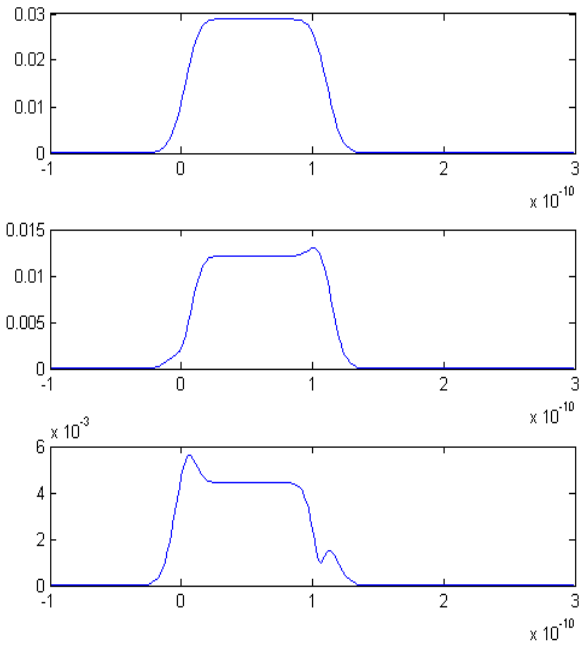

Figure 4.17: 10GHz deltas for 5 meter link on a 3 × 3 MIMO (time in seconds).

From 4.17 we can already see the delaying effect produced on the slowest mode, invading around 25% of the neighbor time slot. For this particular case, this invasion will not be detected with oversampling lower than $sadc = 4$ (sampling every 25% of the bit slot, which is 100 ps at 10 GHz), and even then it might not be accurate enough.

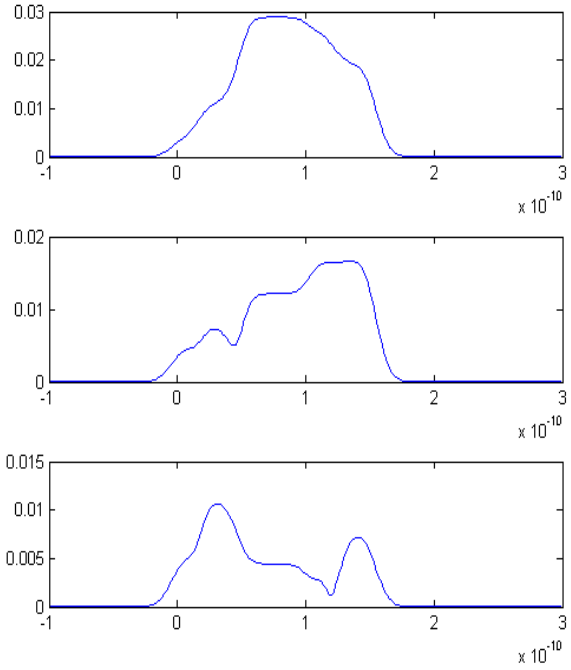

Figure 4.18: 10 GHz deltas for 25 meter link on a 3 × 3 MIMO (time in seconds).

In figure 4.18 we can see that at 25 meters the slower modes invade the neighbor time slot, and high enough oversampling frequencies are required to capture this behavior.

The equalization filter order is limited to the minimum order required for the current length, which in this case is $nmax = (5, 11, 23, 47)$ for the different oversampling $c = (1, 2, 4, 8)$. The thresholds used for OOK are the default ones (not optimized).

To obtain these results, up to 10 #sims for every scenario are run. For the current OSNR this amount of simulations provide enough errors to obtain statistically significant values. The overall BER result shown by the GUI has been taken as the final performance.

Figure 4.19 shows the measured performances with OOK and different oversampling during the first 200 meters. It is easy to notice that when no oversampling is performed or it is very low ($sadc$ 1 and 2) link performance is highly dependent on the current length. Lengths which cause the received deltas to fall between the sampling times of the MIMO training process will cause bad channel estimations, as explained earlier with figures 4.17 and 4.18. This translates in loss of performance observed in the BER

values. When higher oversampling is applied ($sadc$ 4 and 8), this bad equalization still happens but the loss of performance is reduced in most cases.

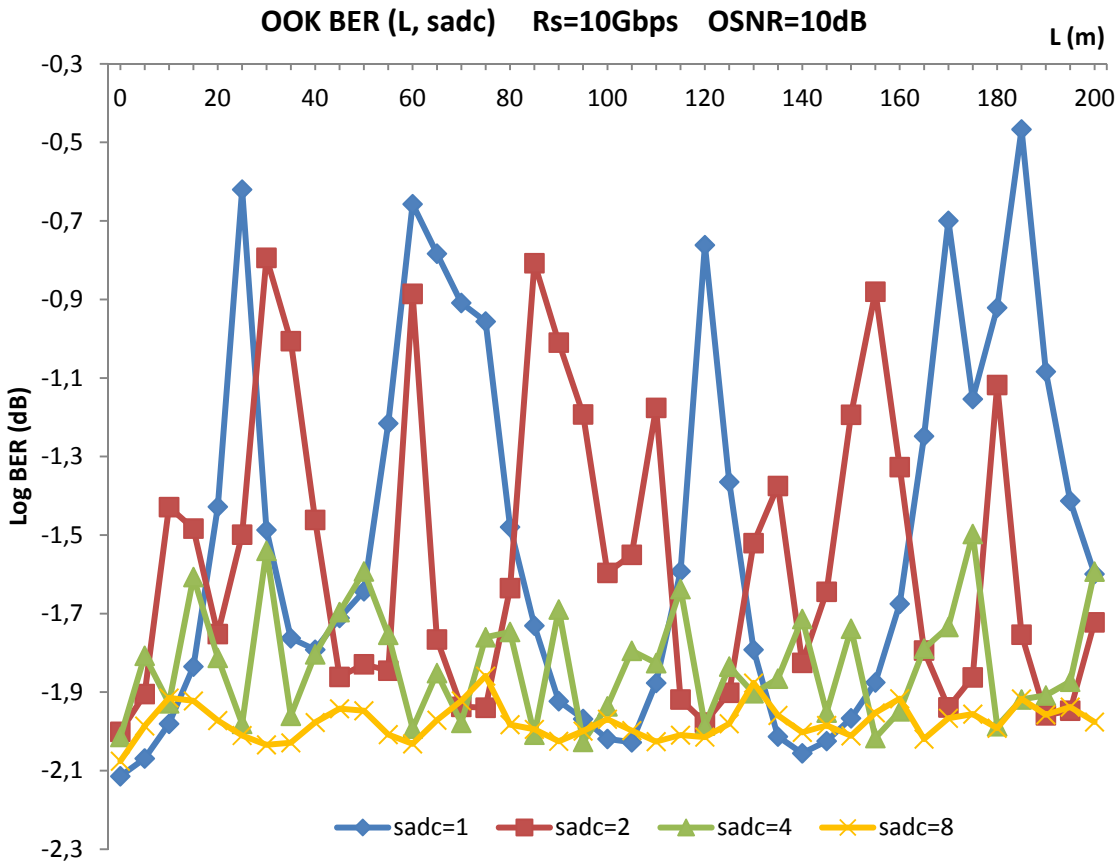

Figure 4.19: OOK BER performance for the first 200 meters with different oversampling.

In fact, when 8 samples per bit are taken (which means 80 GHz sampling rate) the effect has mostly disappeared. This demonstrates that taking more samples per symbol when estimating the channel impulse response will improve the system performance. However, this technique seems to require high amount of samples, which translate into higher filter equalization order, and higher processing speed requirements.

In a real world implementation we could consider that the benefits of this technique do not seem to compensate the hardware and computing needs, and other solutions like adaptive solutions with independently synchronized receivers to optimize the sampling times would probably work much better.

For further experiments in this chapter, only MIMO equalization systems with oversampling of 8 times the symbol rate ($sadc = 8$) will be considered because of their moderate performance dependence with simulation link length.

In addition, we will set up the segment threshold to 200 meters. In other words, links will be divided into segments of 200 meters each, and thus simulated with the

corresponding number of segments, each one of them with their own random coupling matrices. This corresponds to 5 segments per kilometer, and match the result obtained after 2 kilometers seen in figure 4.12.

The deviation of the BER results obtained for curve $sadc = 8$ in figure 4.19 has been calculated from the 10 simulations run for every length, and its value is estimated to be around ±0.1 dB.

In the next part of the experiment, we plot the performance of the system in steps of one segment (200 meters) for the first 4 kilometers for 3×10GHz signals, and additional values for every kilometer until 10 are shown (with 5 segments per kilometer). The simulation parameters are the same as before, using 80 GHz oversampling ($sadc = 8$), and adding extra segments every 200 meters. Results obtained are shown in figure 4.20.

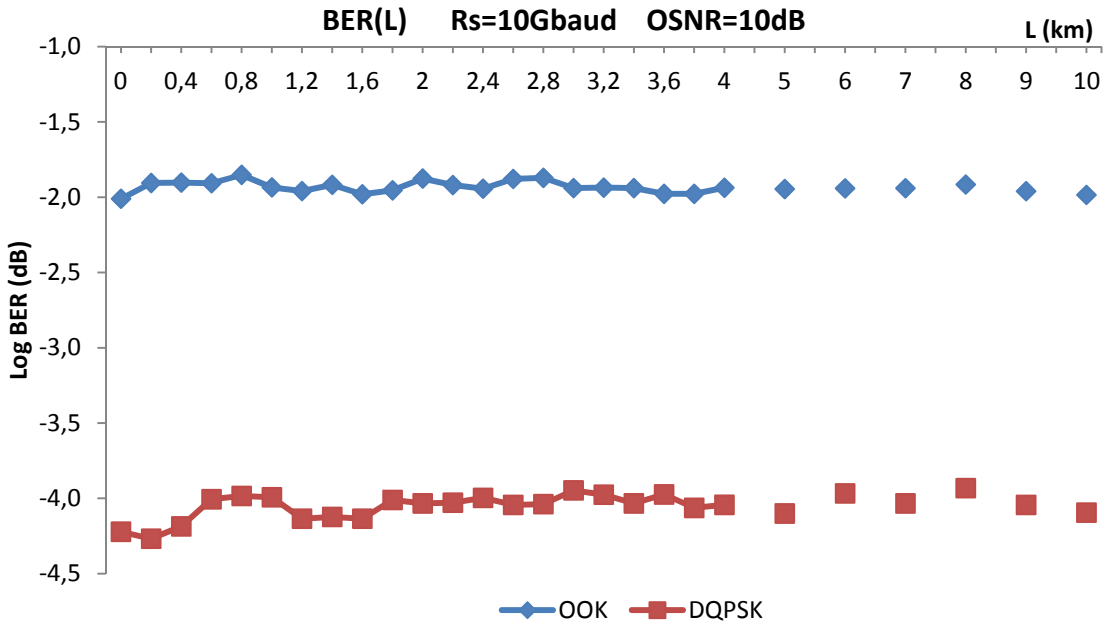

Figure 4.20: OOK and DQPSK BER performance experiment for the first 10 km.

These results indicate that system performance is stable with length increase. Since the gain is automatically set up to compensate for the losses, and the OSNR is the same at the output for all the simulations, the only effect that should be degrading the BER with the increase of length is the chromatic dispersion (default value of 15 ps/nm/km). However, we can see that this negative effect is not affecting the system. This is because the MIMO channel estimation process also captures the broadening of the deltas from the training sequence, thus estimating the impulse response including the effect of chromatic dispersion. This is a beneficial side effect of the MIMO equalization process.

All the BER values used for this experiment plots can be found in appendix A.

4.7 Symbol rate effect

After the results obtained in experiment 4.6, only MIMO equalization systems with an oversampling of 8 times the symbol rate will be considered ($sadc = 8$), basically for their stability for different link lengths. In this experiment the effect on the BER performance when increasing the symbol rate for a specified link length is evaluated. A 3×3 MIMO setup will be considered for a total of 3 input signals with OSNR 10 dB.

These simulations are run with the `MMFLink` interface. The same rise time is set for all the different symbol rates, to a value of 25% percentage the symbol period. The simulation link length has been set to 2 kilometers (typical local environments, such as local area networks). For this length link has been simulated as a cascade of 10 segments (5 per kilometer). A total of 10 simulations with 16384 symbols are run. The rest of the parameters are left by default. Results are plot in figure 4.21.

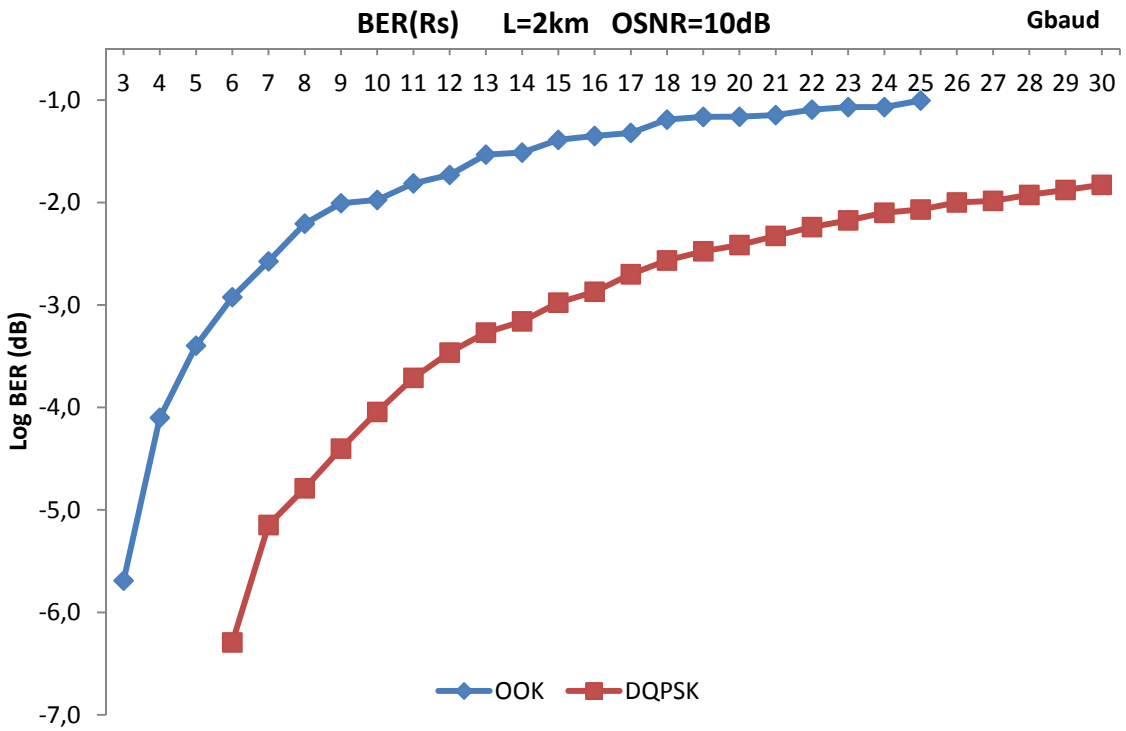

Figure 4.21: BER performance results for different symbol rates.

First thing to notice is that for lower symbol rates good performances are obtained. We should also keep in mind that according to the results of 4.6, this graph should show similar results for other lengths (as seen in figure 4.20). Performance decays quick at the beginning, and steadily afterwards (9 Gbps for OOK and 12 Gbaud for DQPSK). This behavior is to be expected, since the optical and electrical filters are increasing their bandwidth accordingly with the symbol rate, thus allowing more noise into the system. We can rule out the effect of the chromatic dispersion according to the results of experiment 4.6. The BER values used for this experiment are found in appendix A.

4.8 MIMO size effect

This experiment evaluates the effect on the MIMO size for a specific fixed setup. The system setup consists of N signals at 10 Gbaud for both modulations, at a length of 2 kilometers (10 segments) for an OSNR of 10 dB. MIMO equalization is set at 80 GHz ($sadc = 8$). `MMFLink` interface has been used for this experiment, with 10 #sims of 16384 symbols each. Other parameters are left to their default values (see section 3.6.2). Range values of N have been chosen from 2×2 to 10×10 MIMO. The results are shown in figure 4.22.

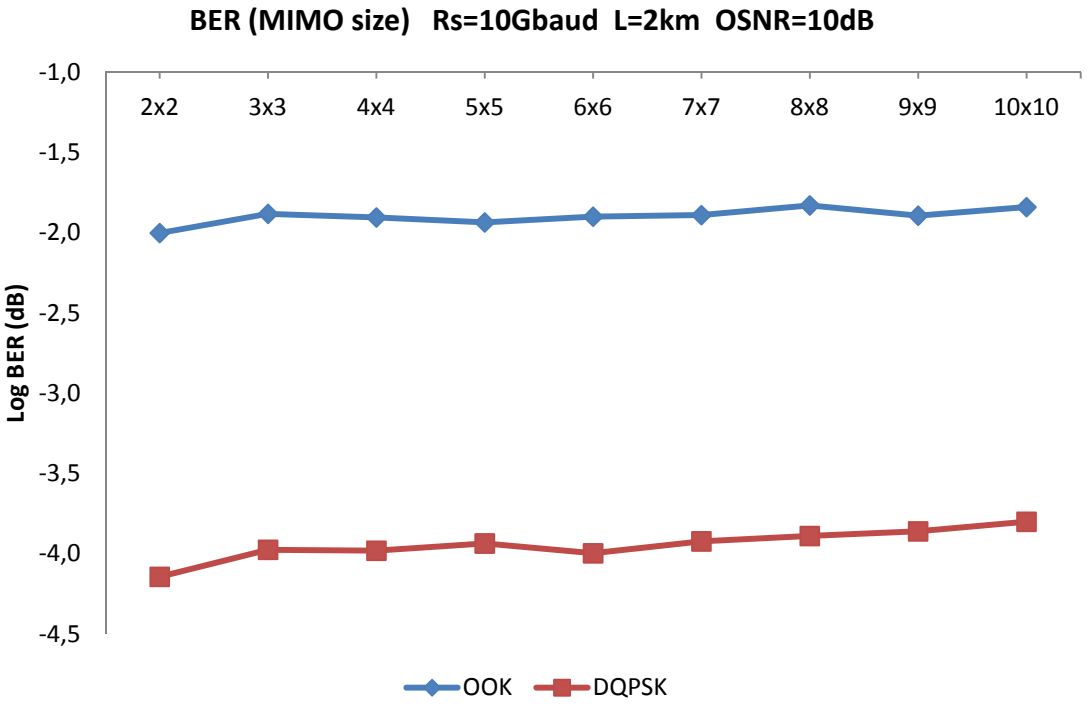

Figure 4.22: BER performance when increasing MIMO size.

Before evaluating the obtained results, an important issue must be discussed: the bigger the MIMO system is, the bigger the equalization filters and the processing speed need to be. This is because the more modes are considered, the more spread they are in time due to the new modes speed (magnitude order of ns/km). For example, with current mode group velocities used in this setup, up to 3000 filter taps in 10×10 MIMO for 2 km were required for one single channel. Therefore, this means that the overall system for 10×10 MIMO up to $10 \times 10 \times 3000 = 300.000$ filter taps are required. These are very big values in terms of digital processing, so the performance obtained for high MIMO sizes has to be look carefully, considering the processing required. This fact becomes a limiting factor in real-life implementations, and some balance is needed between MIMO size and required digital processing speed to achieve a desired BER.

Once this consideration is taken into account, we can observe that both modulation formats perform very similar for all the different MIMO sizes. A slow decaying can be observed for DQPSK at 9 and 10 sizes, and it could be a consequence of the increasing penalties caused by the imperfection of the system. When increasing the MIMO size, more MIMO equalizations per bit are required, thus multiplying the imperfections of this equalization by a proportional amount of the size increase. MIMO capacity and outage probabilities are fully detailed in [15].

The results prove that when high mode coupling fibers are treated with MIMO, increasing the number of modes affected by the coupling does not substantially degrade the information capacity. An agreement between the MIMO size and the available processing speed must be reached.

4.9 Real scenario approximation

In these experiments we want to set the system to be as close as possible to a real life link in order to provide an example of the expectations of this model.

4.9.1 Standard scenario

A 3×3 MIMO system on a random coupling three-mode fiber of 10 kilometers has been simulated for this scenario. The fiber is modeled as a cascade of 50 segments (5 per kilometer). The input signals are modulated at 10 Gbaud each, and the MIMO equalizer works at 80 GHz ($sadc = 8$). The OSNR of the average output is 13dB. A total of 20 simulations with 16384 symbols each have been run with `MMFLink`.

First, the performance for an OOK modulation is evaluated. This same experiment has already been evaluated with the same parameters, but with OSNR of 10 dB. For that value, the BER obtained was around $-2\ dB$ (see figure 4.20). The BER obtained after increasing the signal noise ratio by 3 dB is $-3.4\ dB$. An output example is shown next:

```
*** Total bits/mode simulated: 163840
 $Duration: 228.3s - $Started: 03/06 22:17 $Finished: 03/06 22:21
 > Errors m-1: 74  -  BER=0.00045166
 > Errors m-2: 70  -  BER=0.00042725
 > Errors m-3: 57  -  BER=0.0003479
```

BER= -3.3883 dB

An eye diagram and constellation plots for OOK are shown in figures 4.23 and 4.24 respectively, considering only 2048 bits (Q eye plot for OOK should be ignored).

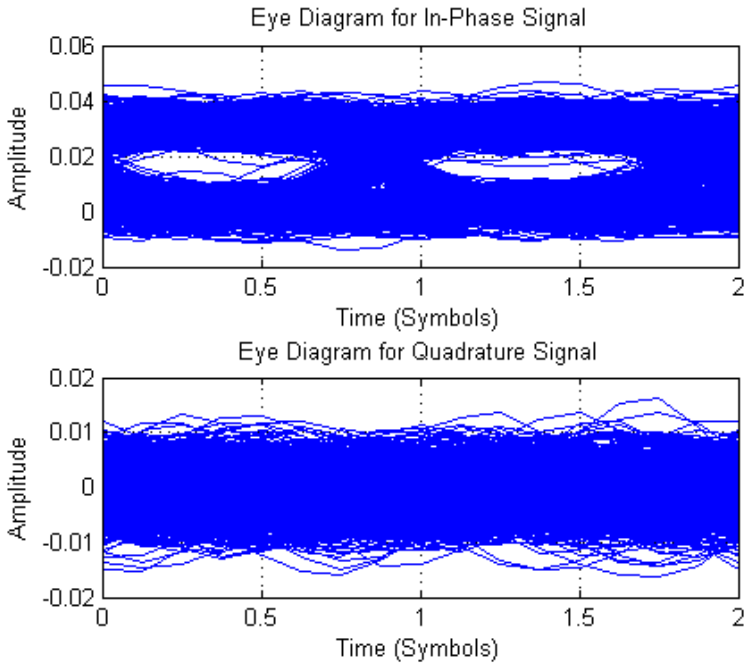

Figure 4.23: OOK eye diagram in one channel for experiment 4.9.1

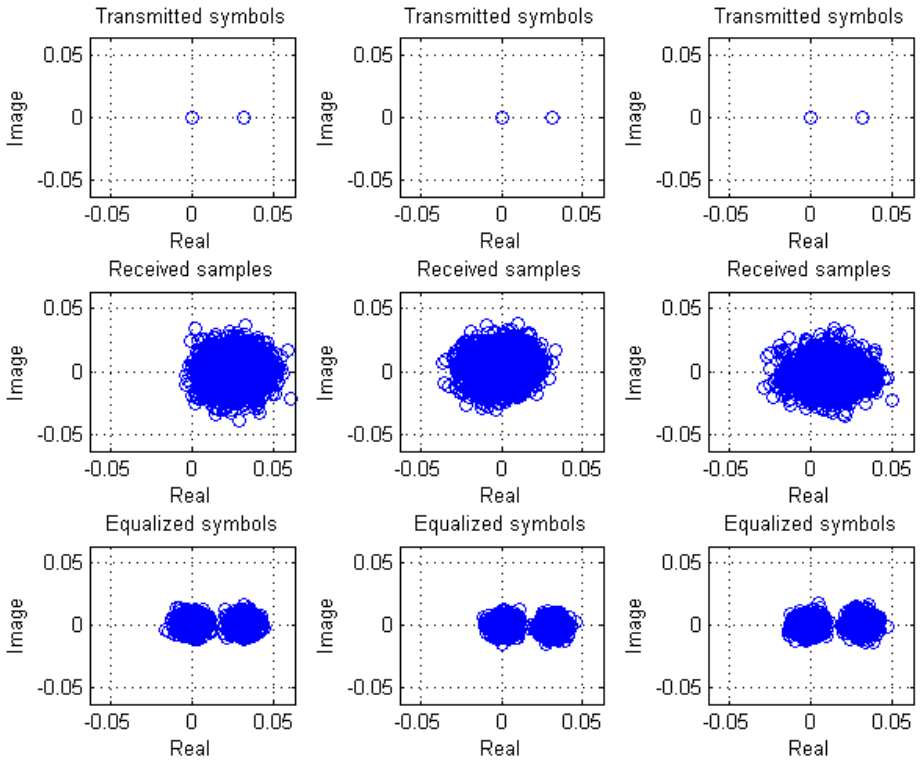

Figure 4.24: OOK constellation plot for experiment 4.9.1

The eye opening in 4.23 can be easily distinguished. When looking at the constellation plot in 4.24 we can see the frontiers between the two symbols, but the separation is very thin. It can also be seen that mode 1 (left) shows more degradation than mode 3 for this particular coupling scenario. We could conclude that for symbol rates close to 10 GHz and OSNR bit higher than 10 dB, OOK could be considered as a modulation format for this system . However, since the MIMO equalizer requires phase components for mode coupling estimation, direct detection schemes cannot be used with current setup, and coherent receiver is still needed. Under the current project premises, the use of OOK instead of phase modulation formats is unadvised.

For DQPSK modulation we have set a lower OSNR value of 10 dB. the BER results obtained after several simulations fluctuate around $-4\ dB$. A simulation output example is shown next:

```
*** Total bits/mode simulated: 655360
 $Duration: 494.1s - $Started: 03/04 13:52 $Finished: 03/04 14:00
 > Errors m-1: 65  -  BER=9.9182e-005
 > Errors m-2: 65  -  BER=9.9182e-005
 > Errors m-3: 71  -  BER=0.00010834
```

BER= -3.9904 dB

This value is consistent with the BER results obtained at figures 4.20 and 4.21 An eye diagram and constellation plots for the DQPSK simulation are shown in figures 4.25 and 4.26 respectively, when considering only 4096 bits.

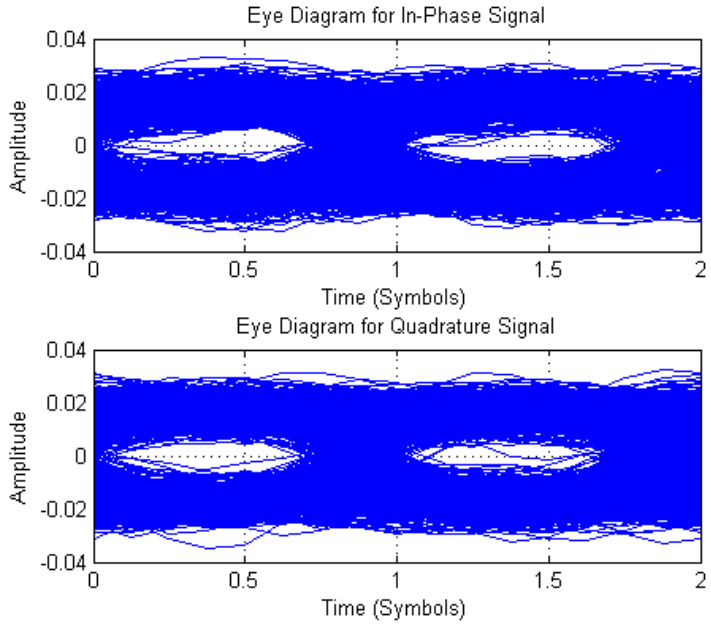

Figure 4.25: DQPSK eye diagram in one channel for experiment 4.9.1.

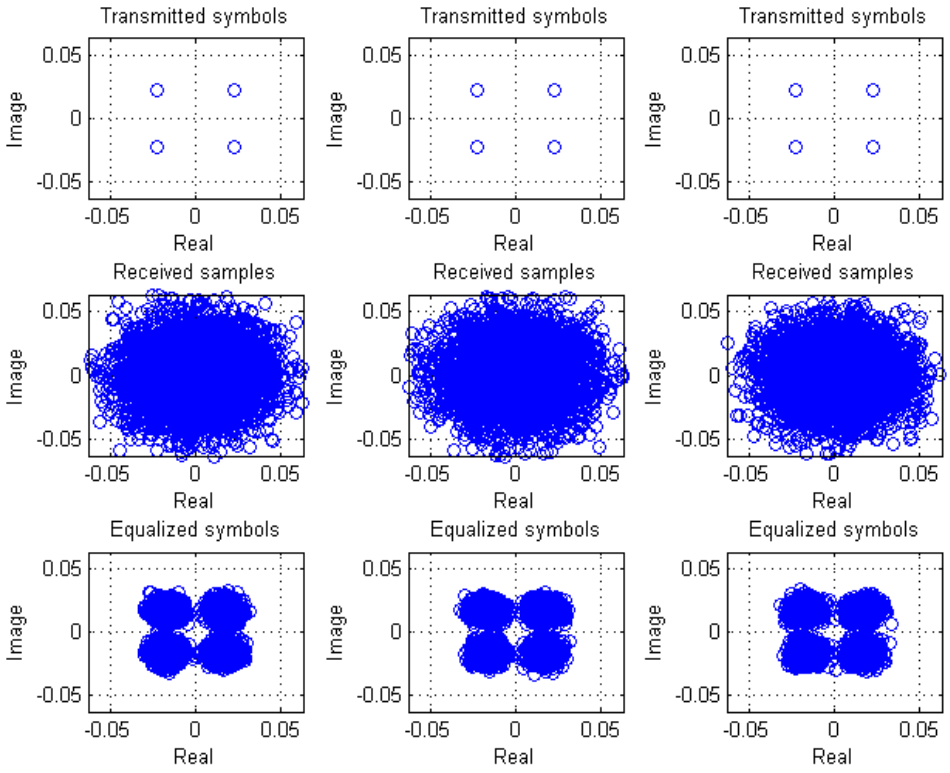

Figure 4.26: DQPSK constellation plot for experiment 4.9.1.

The eye openings for DQPSK and the constellation frontiers are quite clear. These results encourage testing the system using this modulation with tougher conditions.

4.9.2 Tough scenario

As a response of the obtained results on the previous standard scenario, an experiment with increased symbol rates, MIMO size and fiber length is simulated. This simulation runs 8 DQPSK signals at 20 Gbaud, 25% rise time, 8×8 MIMO and 20 kilometers of multimode optical fiber, modeled with 100 segments (5 per kilometer). The MIMO equalization is realized at 160 GHz (8 samples per symbol). OSNR is set to 10 dB. The rest of the parameters are left by default (see section 3.6.2). This experiment can be run with `MIMO8x8.m` script, but might not work in older computers (requires lots of memory).

The link's total bandwidth is: $8 \times 20\ Gbaud \times 2\ (bits\ per\ symbol) = 320\ Gbps$

This scenario uses filters with 42247 taps, for a total amount of a little bit more than 2.7 million taps. This is a very huge number, and some sort of improvement should be implemented. As explained in experiment 4.5, due to the effect on the sharpening of the group delays, there is room for lowering the amount of required samples to estimate the channel responses.

The output of one of the simulations is posted next:

```
*** Total bits/mode simulated: 32768
 $Duration: 1621.9s - $Started: 03/05 09:52 $Finished: 03/05 10:19
 > Errors m-1: 166  -  BER=0.0050659
 > Errors m-2: 150  -  BER=0.0045776
 > Errors m-3: 143  -  BER=0.004364
 > Errors m-4: 160  -  BER=0.0048828
 > Errors m-5: 170  -  BER=0.005188
 > Errors m-6: 145  -  BER=0.004425
 > Errors m-7: 149  -  BER=0.0045471
 > Errors m-8: 164  -  BER=0.0050049
```

BER= -2.323 dB

If we consider that all the processing and sampling requirements are met (which is not a light consideration) and the parameter values of the fiber can be achieved, the results obtained are acceptable. A final average bit error rate of -2.32 dB has been accomplished after several simulations. This translates to one bit error every 200 bits.

The performance has dropped 1.7 dB compared to the experiment in 4.9.1. If we consider that the MIMO size and the link length have not affected the system performance, as we have seen in figures 4.20 and 4.22, the performance decay comes from the increasing of the symbol rate (caused by the increase of noise bandwidth). We can see that similar performance drop was already expected in figure 4.21.

In conclusion, with some improvements in the system and more realistic simulation parameters, this system could prove to be enough to implement short-haul multimode optical links, which could be useful for metropolitan, access or local networks.

5 CONCLUSION

A particular solution to achieve mode division multiplexing with coherent Multiple-Input Multiple-Output (MIMO) processing has been reported. The system implemented in this thesis has demonstrated the correct behavior of this technology, supporting the idea that multi-mode fiber singular properties can be used to enhance the capacity output of the optical transmission.

Evaluating the quality of the system implies evaluating the MIMO receiver. In the implemented solution, the MIMO training is realized estimating the MIMO channel matrix by sending deltas in all the modes and sampling their impulse responses in time. This technique has proven to be a valid option. The equalization process chosen has proven to be enough for the simulated environments. However, the sampling frequency required to estimate the impulse responses and obtain these results seem to be quite high. Sampling rates of 8 times the symbol rate have been used in order to test the performance of the system.

The MIMO implementation is the main challenge of this system. With the sampling requirements obtained, using adaptive filter grids to equalize the MIMO channel are a wise and realistic option [28]. The digital signal processing needed is another challenge, especially for big MIMO sizes and fast transmission rates. It might become a limiting factor, also in terms of power consumption.

When comparing the results obtained in the simulations for both modulations, it is fair to advise the use of multi-level formats with phase modulation (unless non-linear effects are considered [45]). This enforces the use of coherent receivers, which nowadays can be practically implemented with help of simple to recover carrier phase. Since DSP is already required for MIMO, this duo of coherent receiver and MIMO (COMIMO) is a clever approach.

The graphical user interface implemented with the simulation tool developed has allowed quick and easy experimental setup. The numerous display outputs have allowed checking the correct behavior of the system after adding new parameter or effects during the realization of the project.

With the results of experiments 4.9, theoretical simulations of the system for short-haul transmissions for various mode multiplexed channels has been proven.

Future improvements of this project should focus on the addition of new multi-level modulation formats and the generation of full pseudo-random sequences with all symbol patterns. System could also be improved by changing the MIMO training process, either by improving the training sequences (like robust training sequences [46]) or by changing the estimation method entirely. The implementation of an adaptive filter instead of a channel estimation algorithm would be another important upgrade to the system. And finally, taking further considerations regarding nonlinearities, phase noise in the coherent detection, and the mode-division multiplexing and demultiplexing process would increase the accuracy of the simulations and approach to more realistic environments.

As experimental work, a real laboratory setup could be prepared with state of the art tailored few-mode fibers to check system's performance and analyze the different mode division multiplexing techniques and the different MIMO equalization techniques developed.

While there are still many challenges to be solved, mode division multiplexing could become an important part of the increasing capacity solution. With optimized Optical MIMO processing techniques, the possibilities for multimode fiber links should grow substantially.

The work presented in this thesis is another step forward towards this goal.

REFERENCES

[1] G. P. Agrawal, *Fiber-Optic Communication Systems*. New York: Wiley Series in Microwave and Optical Engineering, John Wiley and Sons, 3rd ed., 2002.

[2] Joan M. Gené (jgene@tsc.upc.edu), *Comunicacions Òptiques: apunts de classe*. ETSETB (UPC), departament de Teoria del Senyal i Comunicacions, 2010.

[3] Hernando, J.M. *Transmisión por radio*. Madrid: Centro de Estudios Ramón Areces, 6th ed., 2008.

[4] A. R. Shah, R. J. Hsu, A. Tarighat, A. H. Sayed, B. Jalali, *Coherent optical mimo (COMIMO)*, Journal of lightwave technology, vol. 23 No. 8 August 2005.

[5] R Ryf, S. Randel, A. H. Gnauck, C. Bolle, R.-J. Essiambre, P. J. Winzer, D. W. Peckham, A. McCurdy, and R. Lingle, Jr., *Space-division multiplexing over 10 km of three-mode fiber using coherent 6×6 MIMO processing*, OSA OFC/NFOEC 2011.

[6] J. G. Proakis, M. Salehi, *Communication systems engineering*. 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, 2002

[7] A. Goldsmith, *Wireless Communications*. Stanford University, Cambridge University press, 2005.

[8] K. Okamoto, *Fundamentals of Optical Waveguides*. Elsevier, 2nd edition. 2006.

[9] N. Benvenuto and G. Cherubini, *Algorithms for Communications Systems*, New York: Wiley, 2002.

[10] P. W. Wolniansky, G. J. Foschini, G. D. Golden, R. A. Valenzuela, *V-BLAST: An Architecture for Realizing Very High Data Rates over the Rich-Scattering Wireless Channel*. Bell Laboratories, Lucent Technologies, 1998.

[11] D. Jiang, H. Zhang, D. Yuan, *Joint precoding and power allocation for multiuser transmission in MIMO relay networks*. New York: Wiley Online Library, 2011.

[12] A. Maleki-Tehrani, B. hassibi, J. M. Cioffi, *Adaptive Equalization of Multiple-Input Multiple-Output (MIMO) Frequency Selective Channels*. Bell Laboratories, Lucent Technologies, 1999.

[13] P. J. Winzer, A. H. Gnauck, A. Konczykowska, F. Jorge, and J.-Y. Dupuy, *Penalties from In-Band Crosstalk for Advanced Optical Modulation Formats*. Bell Labs, Alcatel-Lucent. ECOC Technical Digest 2011 OSA.

[14] K.-P. Ho and J. M. Kahn, *Mode-dependent loss and gain: statistics and effect on mode-division multiplexing*, OSA Optics Express 16612, 2011.

[15] P. J. Winzer and G. J. Foschini, *MIMO capacities and outage probabilities in spatially multiplexed optical transport systems*. Bell Labs, Alcatel-Lucent, OSA Optical Express 16680, 2011.

[16] M. J. Yadlowsky and A. R. Mickelson, *Distributed loss and mode coupling and their effect on time-dependent propagation in multimode fibers*. Applied Optics 6664 Vol. 32 No. 33, 1993.

[17] S. Berdagué and P. Facq, *Mode division multiplexing in optical fibers*, Applied Optics Vol. 21 No. 11, 1982.

[18] B. Scmauss, *All-fiber mode-multiplexed transmission of two channels in a single step-index multimode fiber*. Microwave and Optical Technology letters, Vol. 6 No. 2, 1993.

[19] P. J. Winzer and R.-J. Essiambre, *Advanced Modulation Formats for High-Capacity Optical Transport Networks*. Journal of lightwave technology, Vol. 24, No. 12, 2006.

[20] A. H. Sayed and T. Kailath, *A State-Space Approach to Adaptative RLS Filtering*. IEEE Signal Processing Magazine, p. 18-60. 1994.

[21] R. C. J. Hsu, A. Tarighat, A. Shah, A. H. Sayed, and B. Jalali, *Capacity Enhancement in Coherent Optical MIMO (COMIMO) Multimode Fiber Links*. IEEE Communications letters, Vol. 10 No. 3, 2006.

[22] B. Zhu, T. F. Taunay, M. Fishteyn, X. Liu, S. Chandrasekhar, M. F. Yan, J. M. Fini, E. M. Monberg and F. V. Dimarcello, *112 Tb/s Space-division multiplexed DWDM transmission with 14-b/s/Hz aggregate spectral efficiency over a 76.8 km seven-core fiber*. Optics Express 16665, Vol. 19 No. 17, OSA 2011.,

[23] K. Balemarthy and S. E. Ralph, *MIMO Processing of Multi-Mode Fiber Links*, Lasers and Electro-Optics society annual meeting-LEOS, p. 639-640, 2007.

[24] J. Xu and C. Peucheret, *Two-Mode Multiplexing at 2×10.7 Gbps over 7-Cell Hollow-Core Photonic Band Gap Fiber*, DTU Fotonik, ECOC 2011.

[25] Y. Wang, Y. Shao, and N. Chi, *Multiple-Inputs Multiple-Outputs Combining Center Launch and Ring Launch for High-Speed Transmission in Multimode Fiber Links*. Lecture notes in Electrical Engineering, Vol. 143 p. 353-360, 2012.

[26] K. Appaiah, S. Vishwanath, and S. R. Bank, *Device Design and Signal Processing for Multiple-Input Multiple-Output Multimode Fiber Links*. Proc. Of SPIE Vol. 8267, 2012.

[27] C. Yu, J. Liou, Y. Chiu, and H. Taga, *Mode multiplexer for multimode transmission in multimode fibers*. Optics Express 12673 Vol. 19 No. 13, 2011.

[28] S. Randel, R. Ryf, A. Sierra, P. J. Winzer, A. H. Gnauck, C. A. Bolle, R.-J. Essiambre, D. W. Peckham, A. McCurdy, and R. Lingle Jr., *6×56-Gb/s mode-division multiplexed transmission voer 33-km few-mode fiber enabled by 6×6 MIMO equalization*, Optics Express 16697, Vol. 19 No. 17, 2011.

[29] S. Randel, R. Ryf, A. Sierra, P. J. Winzer, A. H. Gnauck, C. A. Bolle, R.-J. Essiambre, D. W. Peckham, A. McCurdy, S. Mumtaz, M. Esmaeelpour, E. C. Burrows, and R. Lingle Jr., *Mode-Division Multiplexing Over 96 km of Few-Mode Fiber Using Coherent 6×6 MIMO processing*. Journal of lightwave and technology, Vol. 30 No. 4. 2012.

[30] C. Koebele, M. Salsi, D. Sperti, P. Tran, P. Brindel, H. Mardoyan, S. Bigo, A. Boutin, F. Verluise, P. Sillard, M. Astruc, L. Provost, F. Cerou, and G. Charlet, *Two mode transmission at 2x100Gb/s, over 40km-long prototype few-mode fiber, using LCOS-based programmable mode multiplexer and demultiplexer*. Optics Express 16593, Vol. 19 No. 17, 2011.

[31] B. Franz, D. Suikat, R. Dischler, F. Buchali, H. Buelow, *High Speed OFDM Data Transmission Over 5 km GI-Multimode Fiber Using Spatial Multiplexing with 2×4 MIMO Processing*, ECOC, 2010.

[32] commsrc.pn, Matlab® Help reference: http://www.mathworks.se/help/toolbox/comm/ref/commsrc.pn.html

[33] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw Hill, 1995.

[34] S. Ramachandran, *Dispersion_tailored Few-Mode Fibers: A Versatile Platform for In-Fiber Photonic Devices*. Journal of litghtwave technology, Vol. 23 No. 11, 2005.

[35] M. B. Shemirani, W. Mao, R. A. Panicker, and J. M. Kahn, *Principal Modes in Graded-Index Multimode Fiber in Presence of Spatial- and Polarization- Mode Coupling*. Journal of lightwave technology, Vol. 27 No. 10, 2009.

[36] K.-P. Ho, and J. M. Kahn, *Statistics of Group Delays in Multimode Fiber With Strong Mode Coupling*. Journal of lightwave technology, Vol. 29 No. 21, 2011.

[37] K.-I. Kitayama, S. Seikai, and N. Uchida, *Impulse Response Prediction Based on Experimental Mode Coupling Coefficient in a 10-km Long Graded-Index Fiber*, IEEE Journal of Quantum Electronics, Vol. QE-16, No. 3, 1980.

[38] F. Mezzadri, *How to Generate Random Matrices from the Classical Compact Groups*, Notices of the AMS, Volume 54 Number 5. 2006.

[39] P. S. Henry, *Error-rate performance of optical amplifiers,* Tech. Dig. OFC, Paper THK3, 1989.

[40] mldivide, Matlab® Help reference: http://www.mathworks.se/help/techdoc/ref/mldivide.html

[41] M. Morelli, L. Sanguinetti, and U. Mengali, *Channel Estimation for Adaptive Frequency-Domain Equalization*, IEEE Transactions on wireless communications, Vol.4 No. 5, 2005.

[42] guide, Matlab® Help reference: http://www.mathworks.se/help/techdoc/ref/guide.html

[43] E. Forestieri, *Evaluating the Error Probability in Lightwave Systems with Chromatic Dispersion, Arbitrary Pulse Shape and Pre- and Postdetection Filtering*, Journal of lightwave technology, Vol. 18 No. 11, 2000.

[44] A. A. Amin, A. Li, S. Chen, X. Chen, G. Gao, W. Shieh, *Dual-LP11 mode 4x4 MIMO-OFDM transmission over a two-mode fiber*, Optics Express 16672 Vol. 19 No. 17, 2011.

[45] R. A. Griffin and A. C. Carter, *Optical differential quadrature phaseshift key (oDQPSK) for high capacity optical transmission*, OFC, paper WX6, 2002.

[46] N. Shariati and M. Bengtsson, *Robust Training Sequence Design for Spatially Correlated MIMO Channels and Arbitrary Colored Disturbance*, IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, 2011.

A. EXPERIMENT DATA

In this appendix the values for all the graphical plots in chapter 4 are detailed. Refer to each one of the experiments for further details. They refer to the BER result obtained by the simulation interface (in dB). `Rs` refers to symbol rate, `L` to length, `sadc` to the oversampling used, and `OOK` and `DQPSK` the modulation types.

A.1 Experiment 4.6 figure 4.20

L (km)	OOK	DQPSK
0	-2,010	-4,222
0,2	-1,906	-4,268
0,4	-1,903	-4,186
0,6	-1,907	-4,006
0,8	-1,851	-3,984
1	-1,935	-3,993
1,2	-1,959	-4,135
1,4	-1,918	-4,123
1,6	-1,981	-4,135
1,8	-1,954	-4,010
2	-1,876	-4,034
2,2	-1,919	-4,029
2,4	-1,943	-3,997
2,6	-1,879	-4,043
2,8	-1,871	-4,038
3	-1,940	-3,947
3,2	-1,936	-3,976
3,4	-1,940	-4,034
3,6	-1,978	-3,974
3,8	-1,978	-4,063
4	-1,937	-4,043
5	-1,946	-4,101
6	-1,941	-3,967
7	-1,940	-4,034
8	-1,916	-3,932
9	-1,960	-4,043
10	-1,985	-4,095

A.2 Experiment 4.6 figure 4.19

L (m)	sadc=1	sadc=2	sadc=4	sadc=8
0	-2,114	-2,001	-2,014	-2,076
5	-2,069	-1,905	-1,807	-1,986
10	-1,981	-1,429	-1,928	-1,915
15	-1,835	-1,483	-1,607	-1,922
20	-1,428	-1,752	-1,812	-1,972
25	-0,620	-1,499	-1,981	-2,010
30	-1,487	-0,794	-1,540	-2,034
35	-1,763	-1,006	-1,960	-2,029
40	-1,792	-1,461	-1,803	-1,977
45	-1,711	-1,861	-1,696	-1,942
50	-1,643	-1,829	-1,593	-1,947
55	-1,215	-1,845	-1,754	-2,008
60	-0,657	-0,886	-1,993	-2,032
65	-0,783	-1,766	-1,851	-1,971
70	-0,909	-1,937	-1,978	-1,922
75	-0,956	-1,940	-1,760	-1,859
80	-1,479	-1,635	-1,747	-1,983
85	-1,731	-0,807	-2,008	-1,995
90	-1,923	-1,010	-1,689	-2,025
95	-1,968	-1,193	-2,026	-1,999
100	-2,019	-1,596	-1,936	-1,969
105	-2,028	-1,550	-1,795	-1,998
110	-1,877	-1,176	-1,826	-2,026
115	-1,592	-1,919	-1,637	-2,009
120	-0,761	-1,977	-1,985	-2,014
125	-1,365	-1,901	-1,835	-1,980
130	-1,792	-1,520	-1,903	-1,877
135	-2,013	-1,375	-1,866	-1,959
140	-2,056	-1,825	-1,713	-2,002
145	-2,025	-1,644	-1,952	-1,985
150	-1,966	-1,194	-1,738	-2,011
155	-1,875	-0,880	-2,017	-1,953
160	-1,675	-1,327	-1,949	-1,917
165	-1,248	-1,794	-1,789	-2,019
170	-0,699	-1,939	-1,734	-1,966
175	-1,154	-1,862	-1,497	-1,956
180	-0,921	-1,118	-1,987	-1,992
185	-0,467	-1,754	-1,919	-1,917
190	-1,084	-1,959	-1,908	-1,958
195	-1,413	-1,947	-1,873	-1,938
200	-1,599	-1,722	-1,593	-1,976

A.3 Experiment 4.7 figure 4.21

Rs (Gbps)	OOK	DQPSK
3	-5,690	
4	-4,101	
5	-3,397	
6	-2,924	-6,294
7	-2,574	-5,148
8	-2,207	-4,789
9	-2,007	-4,402
10	-1,975	-4,043
11	-1,814	-3,712
12	-1,730	-3,464
13	-1,532	-3,270
14	-1,513	-3,159
15	-1,388	-2,977
16	-1,350	-2,871
17	-1,321	-2,700
18	-1,191	-2,566
19	-1,164	-2,476
20	-1,163	-2,415
21	-1,146	-2,326
22	-1,093	-2,240
23	-1,069	-2,174
24	-1,068	-2,100
25	-1,005	-2,068
26		-2,000
27		-1,984
28		-1,925
29		-1,878
30		-1,828

A.4 Experiment 4.8 figure 4.22

MIMO	OOK	DQPSK
2x2	-2,002	-4,144
3x3	-1,882	-3,976
4x4	-1,905	-3,981
5x5	-1,936	-3,936
6x6	-1,900	-3,997
7x7	-1,890	-3,923
8x8	-1,830	-3,890
9x9	-1,894	-3,860
10x10	-1,840	-3,800