Università degli studi di Padova

Facoltà di Ingegneria

Tesi di Laurea Specialistica in

Ingegneria delle Telecomunicazioni

Digital Signal Processing Algorithms for High-Speed Coherent Transmission in Optical Fibers

Relatore

Prof. Andrea Galtarossa

Candidato

Marco Mussolin

Anno Accademico 2009/2010

Joy to the world!
Hunter "Patch" Adams

Acknowledgements

There are so many people I would like to thank. Since I think it is impossible to thank them all, I apologies with all of those that are not mentioned.

First of all, my acknowledgment to my family that prompted me to applied for this experience that turn off like one of the best experience in my life and supported me during these amazing six months, thank you very much.

Another special thank you goes to my supervisor Andrea Galtarossa for proposing me this great opportunity that was really interesting and formative. Thank also for your kind help before, during and after my stay in Stockholm.

I would like to thank my industrial supervisor Marco Forzati for guiding and helping me every time I was stuck with my work. I have always felt home with you and during these six months in Acreo I had a great time. I will always be indebted and it was a pleasure working with you.

Many thanks also to Jonas Mårtensson, who was actually my second supervisor after Marco Forzati and Sergei Popov that played an important role between my Italian university and KTH.

After that, my special thanks to:

My girlfriend Angela for all the romantic moments spent in Stockholm. I

will never forget the first time I hugged you in Skavsta.

All the people I've met in Stockholm that made this experience wonderful. A piece of me will stay forever in Sweden. All the Italian (and acquired) friends in Kista. Paolo, Luigi, Daniel, Elisa, Stefano, Jessie, Valeria, Antonello, every Tuesday I'll will think about you and *Forno Romano*. Thank you also for all the fun had together. A special thank you goes to Paola for putting me up in her place in my last Swedish week. All my Erasmus friends especially Bastien for all the football match played together (keep on training and maybe you will become stronger as me!); Alex, Arnau, Kathrin for all the dinner (in Kathrin's kitchen) in Sundbyberg and for helping me in my last days without accommodation; Kuba for his specialty from Poland; Alicia for the parties in Bromma. Steffi, Sarah and Martina you left in December but we spent three months of great fun. A special thank you goes to Federico. You helped me really a lot in my first Swedish week and without you I would never met such a wonderful persons. Thank you also for all the crazy moment all around Stockholm. *The Isotopes* for all the matches played together. Everybody in Acreo and the *students corner* for all the *fikas* made together.

All my friend in Vicenza especially to the SanLore group guided by our leader Lush. Every time I was back to Italy it was amazing to find friends waiting for me.

CDSA members for all the wonderful time during these years of university.

The clowns of V.I.P. Clown Vicenza, since 2003 you are a special part of my life.

All my friend that visited me in Stockholm. Thank you for all the funny moments spent together.

I also would like to thank KGB, Ambassadeur, Soap Bar, Marie Laveau,

DKV, Nymble, Blå Dörren, Ica, Coop, SL, Skype and RayanAir.

Introduction

Internship project was performed in the Transmission Group in Acreo AB under the supervision of Senior Scientist Marco Forzati.

Acreo AB

Acreo AB was formed in 1999 by bringing together Swedish research institutes in the microelectronics and photonics area together. It has more than 150 employees in headquarter in Kista (Stockholm) and in two other regional offices. Acreo is owned by the group Swedish ICT Research AB. 60% of the latter is owned by Swedish Ministry for Industry and the rest is owned by the industry through consortium.

Acreo fulfills its mission by carrying out contract research and technical development. Acreo's role is to bridge the gap between academic research and industrial commercialization.

EURO-FOS network

The internship project was done in scope of EURO-FOS project. EURO-FOS is a Network of Excellence project and it was funded by the European Commission under the 7th Framework Program (FP7), Information and Commu-

nications Technologies (ICT). The aim of EURO-FOS is to create European network of research groups, who has proved to be leaders in the design, development and evaluation of photonic subsystems. EURO-FOS Network has several *Centres of Excellence* from which the Transmission Group of Acreo is involved in *CE1: Digital Optical Transmission Systems*. From the objectives of the Centre of Excellence on Digital optical Transmission Systems, we can mention the followings, which somehow coincide with the objectives of current internship project:

- to study the benefits of advanced modulation formats in combination with coherent detection and electronic distortion equalization to increase spectral efficiency and to reduce system complexity;

- analysis of techniques for electronic dispersion compensation and electronic mitigation of linear and non-linear transmission impairments;

- to study concepts of transmitter and receiver subsystems for 100 Gb/s systems;

- to study and analyze the implementation of subsystems for optical clock recovery.

One of the efficient ways of collaboration in the scope of EURO-FOS network is joint experiments, when experiments and results are open for all the collaborating groups. Especially during 2008 the experiments on QPSK modulation format for the Centre of Excellence on Digital optical Transmission Systems were done in Heinrich Hertz Institute (Germany) and Politecnico di Torino (Italy). For the current project it was decided to develop DSP algorithms targeted on the experiments performed in Politecnico di Torino.

Abstract

Digital Signal Processing (DSP) is an indispensable technology for next generation 100 Gb/s optical coherent transmission system.

The aim of this thesis is to study DSP algorithms to compensate transmission impairments in a 100 Gb/s PolMux-QPSK coherent optical transmission system. The thesis can be divided in two parts: the first deals with testing different chromatic dispersion compensation techniques (Savory's method, adaptive filters and frequency domain method) while the second part is focused on nonlinear transmission impairment compensation using a multi-span backpropagation technique. Constant Modulus Algorithm butterfly structure and Viterby and Viterby methods are proposed for polarization de-multiplexingn and adaptive and for carries phase recovery. Results of the processing, presented in the last chapter, confirm that coherent detection at 100 Gb/s will become feasible in the future using DSP for transmission impairments compensation.

Contents

List of Figures

Chapter 1

Optical fiber communications

With the increasing of global information exchange it's becoming crucial to be able to transmit information over longer distance. An answer to this issue are optical fibers that are already used for most of the voice and data traffic all over the world.

Optical fibers are especially advantageous for long transmissions distance, because light propagates through the fiber with limited attenuation compared to electrical cables. The speed at witch the information in traveling along a single optical channel (bit rate) is limited by the speed of electronic components of the transmission system. However it's possible to transmit several channels, at different wavelength, on the same medium using wave division multiplexing (WDM). This allows to reach a capacity system of several Tbit/s[1].

This chapter introduces to optical fiber communications. Section 1.1 discusses the main transmission impairment in high-speed transmission. In section 1.2, the modulation format used in data analyzed in section 3 is presented. Section 1.3 briefly reviews a comparison between direct detection and coherent detection.

1.1 Signal propagation in optical fibers

The goal of a communication system is to propagate a signal $A(t,z)$ from the transmitter to the receiver so that signal at the receiver $A(t,z_{end})$ will be as similar to the transmitted one $A(t,0)$.

The propagation of a light pulse in a optical fiber is described by the *nonlinear Schrödinger equation* (NLSE)[2]:

$$j\frac{\partial A}{\partial z} = \frac{\beta_2}{2}\frac{\partial^2 A}{\partial t^2} - j\frac{\alpha}{2}A - \gamma|A^2|A, \qquad (1.1)$$

A is the electric field, α is the attenuation coefficient, β_2 is the dispersion parameter, γ is the nonlinear coefficient, z and t are the propagation direction and time, respectively[3]. Each of the three terms on the right side of Eq. 1.1 describe impairments that cause signal distortion and need to be compensated for if error-free transmission is to be achieved.

1.1.1 Attenuation

Attenuation causes the power level of the signal to decrease while propagating and optical amplifiers are used for compensate for this power loss. The price to pay is that this introduces noise and increases the system cost.

Removing the effect of chromatic dispersion and Kerr nonlinearity the Eq. 1.1 can be solved and gives:

$$|A(t,z)|^2 = |A(t,0)|^2 e^{-\alpha z}, \qquad (1.2)$$

α is the *attenuation constant* and it's measured in m^{-1} but usually it's referred as α_{dB} defined as:

$$\alpha_{dB} = 4.343 \cdot \alpha. \qquad (1.3)$$

2

There are different values of attenuation depending on the optical frequency range that can be divided in three window:
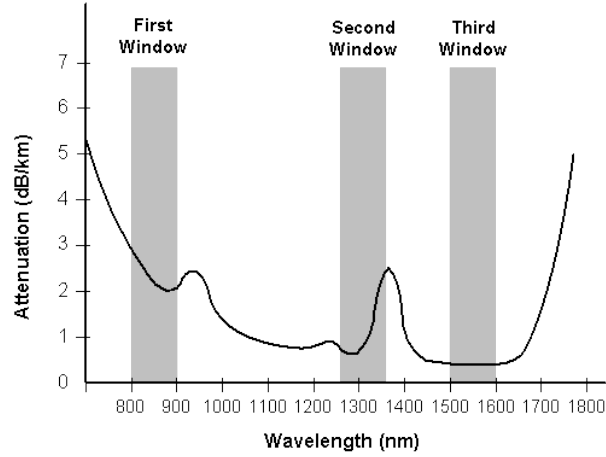
Fig. 1.1: Optical fiber attenuation

- first window centered around 850 nm with typical value of attenuation of 2 to 3 dB/km;

- second window centered around 1300 nm with typical value of attenuation of 0.4 to 0.5 dB/km;

- third window around 1550 nm with attenuation of 0.2 to 0.25 dB/km.

1.1.2 Chromatic dispersion

Chromatic dispersion (CD) is due to propagation speed of light being a function of wavelength, so that the different spectral components of signal show a relative delay causing signal distortion.

Ignoring the nonlinearity and the attenuation in Eq. 1.1 the resulting system becomes linear and we can write the expression that shows the effects of CD on the envelope $A(z,t)$:

$$\frac{\partial A(z,t)}{\partial z} = -j\frac{\beta_2}{2}\frac{\partial^2 A(z,t)}{\partial t^2}, \tag{1.4}$$

where z is the propagation distance, t is time variable, and β_2 is the *group-velocity dispersion* (GVD), often called simply dispersion.

The group-velocity is the traveling speed of a light pulse in the fiber and it is a function of ω therefore different spectral components of the signal travel with different group velocity.

Solving analytically Eq. 1.4 in the Fourier domain gives:

$$A(z,\omega) = A(0,\omega)e^{j\frac{\beta_2}{2}\omega^2 z}. \tag{1.5}$$

It can be seen that in the frequency domain the chromatic dispersion introduces a distortion on the phase of the signal spectrum without changing the spectral power distribution and at the end of the propagation the pulse result broadened.

The GVD parameter β_2 gives the time delay between two different spectral component separated by a certain frequency interval and it has units s^2m^{-1}. Usually the dispersion is measured with the dispersion coefficient D defined as:

$$D = -\frac{2\pi c}{\lambda^2}\beta_2, \tag{1.6}$$

where $\lambda = -2\pi c/\omega$ is the carrier wavelength and c is the speed of light. D gives the time delay between two different spectral component separated by a certain wavelength interval and it has units ps/nm/km.

It is possible to define a *dispersion length* defined as the propagation distance after a Gaussian pulse is broadened by 40%:

$$L_D = \frac{\tau_p^2}{|\beta_2|}, \tag{1.7}$$

where τ_p is the pulse half width[1].

1.1.3 Polarization mode dispersion

As in long-haul link standard single mode fiber (SSMF) are used, the propagating filed is described as a single mode consisting of two degenerate modes each corresponding of two orthogonal polarizations. The degeneration is due to the cylindrical symmetry of the optical fiber. However real optical fibers have a physical structure that is not perfectly cylindrical to core imperfections and perturbations due to mechanical tension, thermal gradients etc.
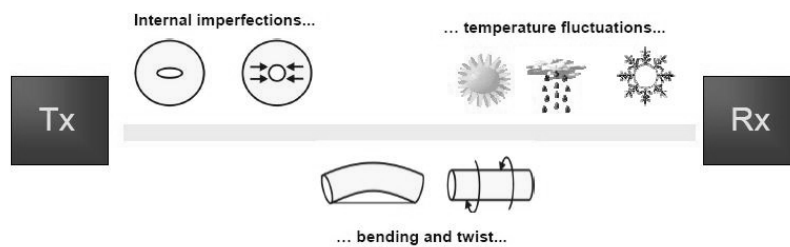
Fig. 1.2: Fiber imperfection[4]
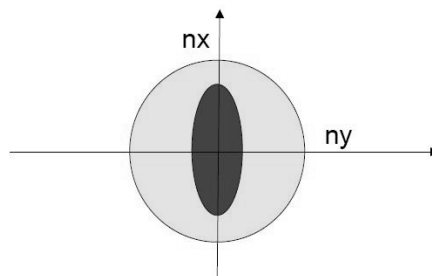
Fig. 1.3: Perturbated fiber core[4]

5

Due to these imperfections, the two fundamental modes see different effective indexes of refraction (Fig. 1.3) and optical fibers acquire birefringence meaning that the two polarization modes are propagating with slightly different group velocity. The state of polarization varies during the propagation. The polarization state is dependent on the wavelength and evolves with propagation at the end of the link, the two pulse components with different polarizations will arrive with a relative delay called *Differential Group Delay* (DGD) (Fig. 1.4).
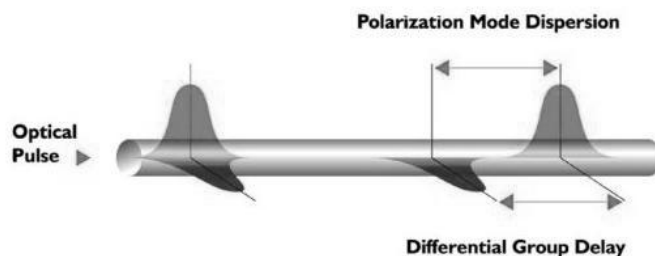
Fig. 1.4: Impact of PMD on the propagating pulse

This phenomenon is called *Polarization Mode Dispersion*(PMD). It's stochastic as a consequence of the random nature of its origin. The total DGD is given by:

$$\sigma(z) = D_g\sqrt{z}, \tag{1.8}$$

where D_g is the PMD coefficient and its typical value is in the range between 0.1 and 1 ps/km.

1.1.4 Nonlinear effect

The response of any dielectric to light becomes nonlinear with the increasing of the intensity of electromagnetic fields and optic fibers, made of silica,

behave this way. The origin of the non linear response comes from the interaction of the electromagnetic field with silica electrons.

Solving the 1.1 in absent of dispersion ($\beta_2 = 0$) we obtain:

$$A(t, z) = A(t, 0)e^{j\phi_{NL}(t,z)}, \tag{1.9}$$

$\phi_{NL}(t, z)$ the nonlinear phase shift is given by:

$$\phi_{NL}(t, z) = \gamma P_0 |A(t, 0)|^2 \frac{1 - e^{\alpha z}}{\alpha}, \tag{1.10}$$

where P_0 is related with the pulse peak. The Eq. 1.10 shows that the amplitude introduces a nonlinear phase shift. This phenomenon is called *self-phase modulation* (SPM) and, in frequency domain, it brings a widening of the spectrum while the shape pulse remain unchanged. If the system uses a Wavelength Division Multiplexing (WDM), several signals are transmitted at different wavelengths and the one wavelength signal $A(t, 0)$ is affected by other nonlinear shifts duded to the neighbor channels. This phenomenon is called *cross-phase modulation* (XPM). The presence of different wavelength channels can also generate new frequencies as the *four-wave mixing* (FWM). Generally these effects can be divided in two categories: intra-channel and inter-channel nonlinear effects. The first deals with the nonlinear effects of a single channel on itself while the second deals with the effects due to neighbor channels.

1.2   Data modulation

In modern telecommunications system information is stored and transmitted as digital data represented by symbols chosen from a finite alphabet. Digital data modulations consist in associating every symbols value to a particular signal state. In binary transmission only two symbol values are allowed, 0 and 1. In an optical transmission data modulation encode the information over an optical signals using intensity, phase or frequency[1]. In this section is a brief introduction the modulation formats.

Amplitude-shift keying

Amplitude-shift keying (ASK) is a form of modulation that uses the amplitude of the carrier wave to represent the digital data. While frequency and phase are kept constant, the amplitude varies according with the bit stream. For binary transmission the level of amplitude can be used to represent binary logic 0s and 1s (Fig. 1.5). We can think of a carrier signal as an ON or OFF switch. In the modulated signal, logic 0 is represented by the absence of a carrier, thus giving OFF/ON keying operation and hence the name On-Off-keying (OOK) given.
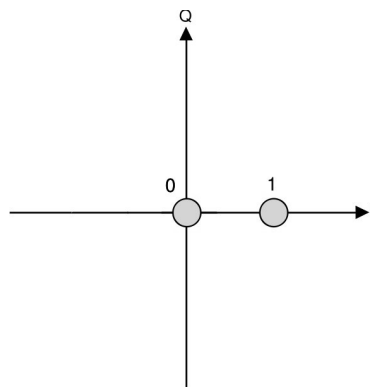
Fig. 1.5: On-Off-keying constellation

Phase-shift keying

Using phase *shift keying* (PSK) the information is stored on the phase of the transmitted signal. Some of the advantages introduced by the modulation format is a better tolerance to noise at the receiver (receiver sensitivity) as well as the increasing of the spectral efficiency and the dispersion tolerance compared to On-Off keying (OOK)[1]. Figure 1.6 shows a *Binary PSK* (BPSK) constellation where the phase is digitally mapped in the two allowed phase states $(0, \pi)$. The constellation can be obtained using a Mach-Zehnder EAM optical modulator that generates a PSK signal when driven by a single drive signal (Fig. 1.7).
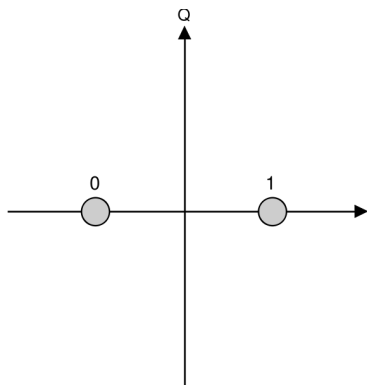
Fig. 1.6: Binary PSK constellation
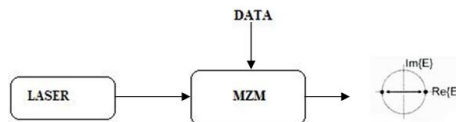
Fig. 1.7: PSK-OOK transmitter using a Mach-Zehnder EAM optical modulator

Quadrature phase-shift keying

Sometimes known as *quaternary or quadriphase PSK* (QPSK), QPSK uses four points on the constellation diagram, equispaced around a circle on four phase value $(0, \frac{\pi}{2}, \pi, -\frac{\pi}{2})$. With four phases, QPSK can encode two bits per symbol (Fig. 1.8). This may be used either to double the data rate compared to a BPSK system while maintaining the bandwidth of the signal or to maintain the data-rate of BPSK but halve the bandwidth needed. Usually QPSK is implemented using a nested Mach-Zender modulator (MZM) structure (Fig. 1.9) where complex field is modulated both in-phase and quadrature direction.
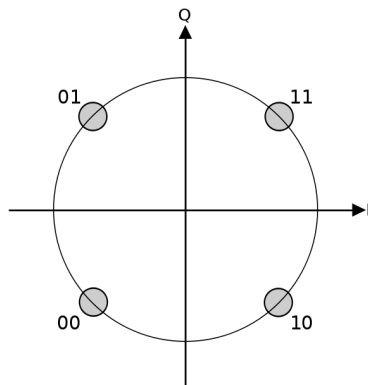
Fig. 1.8: QPSK Grey coded constellation

Polarization shift keying

In *polarization shift keying* (PolSK) the signal is encoded using the signal polarization state. A bit value 0 could be represented by a horizontal polarized signal and a bit value 1 by a vertical polarized signal. However, the polarization rotate randomly during the propagation of the optical field as explained in section 1.1.3. This problem limits the system performance especially for high power when non linear effects become stronger but can be
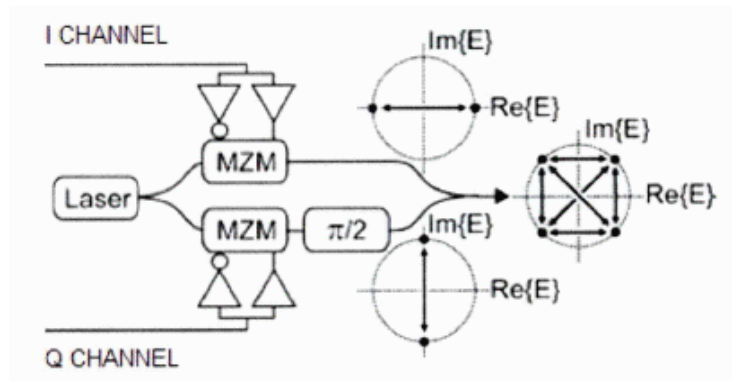
Fig. 1.9: QPSK transmitter using a nested MZM modulator

compensated electronically with proper algorithms. It makes more sense to use the polarization degree of freedom to transmit two orthogonal polarized channels. An example of this is PolMux-QPSK that will described below.

Polarization multiplexing of QPSK

Polarization multiplexing of QPSK (PolMux-QPSK) is the modulation format of the data analised in section 3. PolMux-QPSK is implemented using two MZMs and then the two QPSK signals are combined with a Polarization Beam Combiner (PBC) (Fig. 1.10). PolMux-QPSK helps to reduce the requirements on electrical and opto-electrical components because it requires half the baud rate needed by a QPSK (Fig. 1.11).
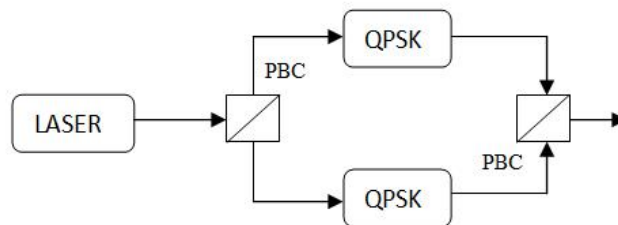
Fig. 1.10: PolMux-QPSK block scheme
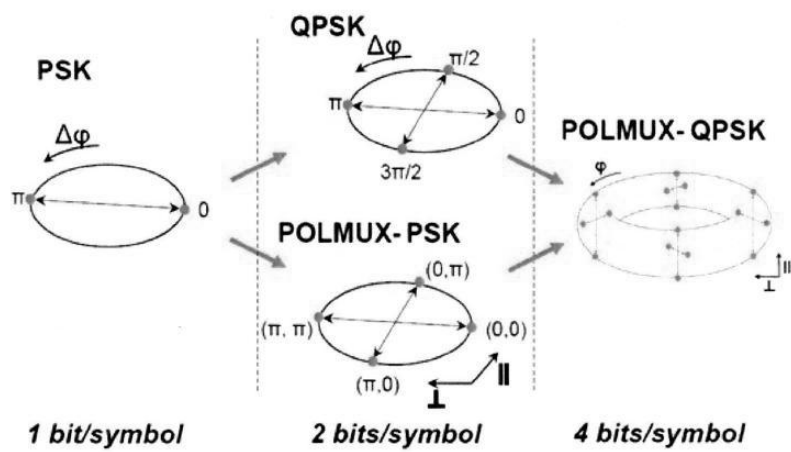
Fig. 1.11: Evolution from PSK to POLMUX-QPSK[5]

1.3 Direct detection versus coherent detection

In direct detection, in an optoelectrical photodetector (a photodiode) the light intensity $|E|^2$ is converted in an electrical signal and the phase information is totally lost (Fig. 1.12). An alternative way to detect the optical signal is coherent detection in which the received signal is mixed with local laser being detected in the photodiode, and two detectors and proper phase delays are used (as explained below), both amplitude and phase can be preserved. While coherent detection was experimentally demonstrated as early 1979, its use in commercial systems has been hindered by the additional complexity, due to the need to track the phase and the polarization of the incoming signal (Fig. 1.13). In a digital coherent receiver these functions are implemented in the electrical domain leading to a dramatic reduction in the complexity of the optical receiver[6].
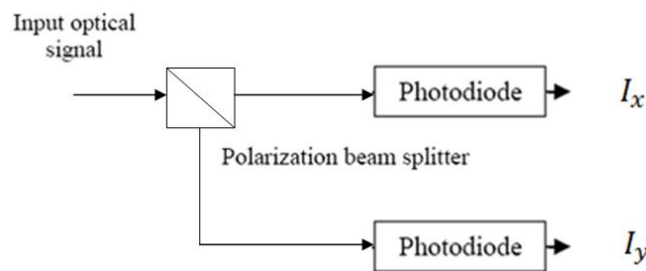
Fig. 1.12: Schematic direct receiver

Since coherent detection maps both the intensity and the phase of the optical field into the electrical domain therefore maintaining all the information it maximizes the effectiveness of the signal processing. This allows impairments which have traditionally limited 100 Gb/s single channel systems to be
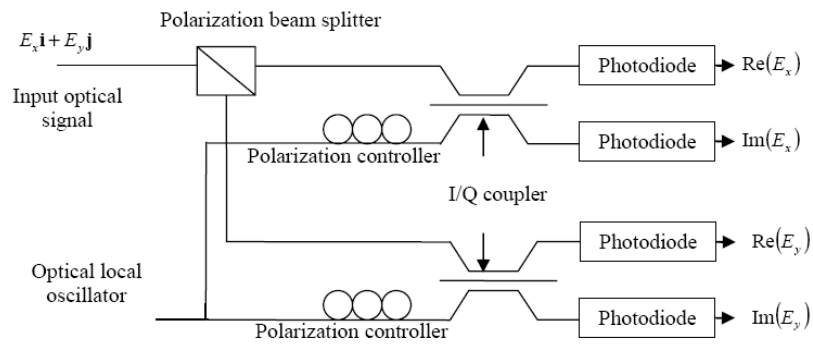
Fig. 1.13: Schematic coherent receiver[7]

overcome, since both chromatic dispersion and polarization mode dispersion may be compensated adaptively using linear digital filters.

Chapter 2

Digital Signal Processing aided coherent optical detection

This chapter introduces the concept of Digital Signal Processing in optical communications. Section 2.1 briefly explains why Digital Signal Processing it's becoming a need in long-haul optical fiber system. In section 2.2, 2.3 and 2.4 linear transmission impairments compensation techniques are presented, finally section 2.5 reviews a nonlinear effects compensator.

2.1   Why Digital Signal Processing

An important goal of a long-haul optical fiber systems is to transmit the highest data throughput over the longest distance without signal regeneration. *Digital signal processing* (DSP) is used at the receiver to remove the need for dynamic polarization control and also to compensate for linear (and some extent of non linear) transmission impairments[7].

An optical transmission system can be represented as shown in Fig. 2.1 where E_{TX} is the transmitted signal, $H(\omega)$ is the channel transfer function
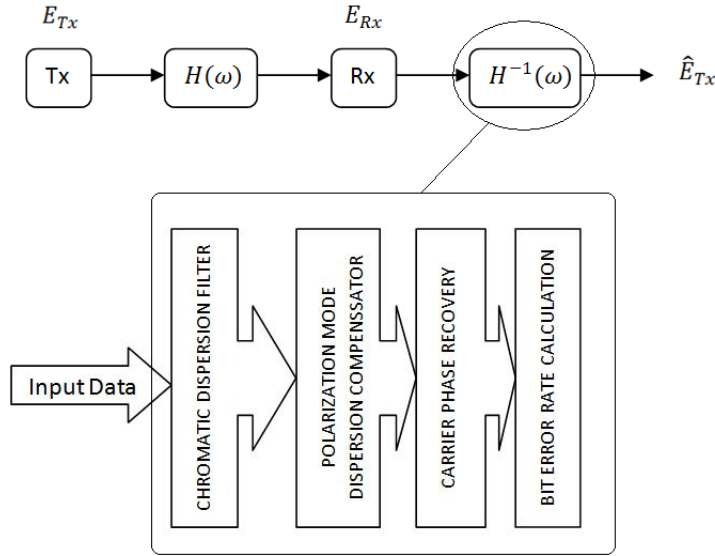
Fig. 2.1: Transmission and DSP block scheme

and E_{RX} is the received signal. The goal of DSP is to implement $H^{-1}(\omega)$, that can be interpreted as the combination of all the linear effects that affect the signal during the propagation, and estimate $\hat{E}_{TX}$ that represents the processed signal. In order to compensate all these effects, the received sampled electrical signal is elaborated with a series of algorithms in order to minimize the *bit error rate* (BER) that represents the main evaluation criterion for digital communication system quality. Another evaluation factor is the so called Constellation Plots or Diagrams (Fig. 2.2), a scatter plot of the modulated signal, amlitude and phase, in a two dimensional complex plane.

Fig. 2.2: QPSK constellation from a PolMux-QPSK (section 1.2): (a) received data constellation diagram without DSP, (b) received data constellation diagram with DSP

2.2 Chromatic dispersion compensation

Chromatic dispersion is one of the main limitation to long-haul direct detection optical fiber system. Its effect of broadening the pulse during propagation (see section 1.1.2) is extremely constraining if not compensated. A commonly adopted solution is the use of *Dispersion-Compensating Fiber* (DCF), however DCF introduces additional loss, therefore requiring additional optical amplifiers that increase noise and cost of the system. An alternative approach is to compensate entirely CD in the electric domain. At the beginning, equalization with training sequence was used but it was demonstrated that this method is good only for low dispersion system, as real system has a relative high CD, or system with DFC and residual CD[8]. A way to solve these problems is to deduce the needed equalized directly from the physics. As noted by Haykin "Signal processing is at its best when it successfully combines the unique ability of mathematics to generalize with both the insight and prior information gained from the underlying physics of the problem at hand"[9].

From Eq. 1.4 we can rewrite the expression of the effect of chromatic dispersion on the envelope A(z,t):

$$\frac{\partial A(z,t)}{\partial z} = j\frac{D\lambda^2}{4\pi c}\frac{\partial^2 A(z,t)}{\partial t^2},\tag{2.1}$$

Solving Eq. 2.1 in the frequency domain it gives:

$$A(z,\omega) = A(0,\omega)e^{-j\frac{D\lambda^2}{4\pi c}\omega^2 z}.\tag{2.2}$$

The frequency domain transfer function G(z,ω) can be obtained from Eq. 2.2[6]:

$$G(z,\omega) = e^{-j\frac{D\lambda^2}{4\pi c}\omega^2 z},\tag{2.3}$$

where ω is the angular frequency. Using Fourier transforms Eq. 2.3 the filter in time domain can be obtain[6]:

18

$$g(z,t) = \sqrt{\frac{c}{jD\lambda^2 z}} e^{j\frac{\pi c}{D\lambda^2 z}t^2} \tag{2.4}$$

From theory, filters for dispersion compensation in frequency and time domains can be obtained inverting the sign of the chromatic dispersion in Eq. 2.3 and Eq. 2.4. Fig. 2.3 shows the effect of CD on a QPSK constellation diagram.

In the next section time domain techniques will be presented then in section 2.2.2 frequency technique will be analyzed.
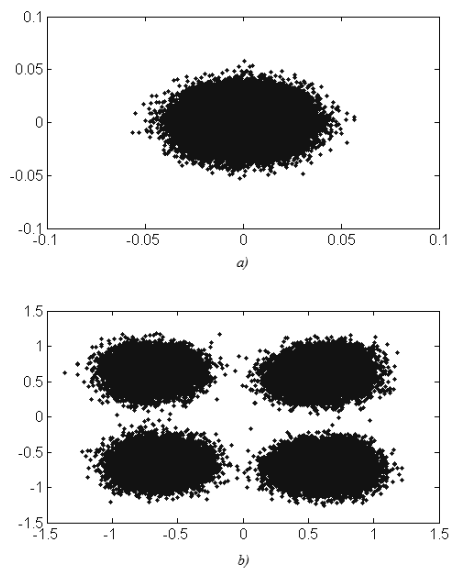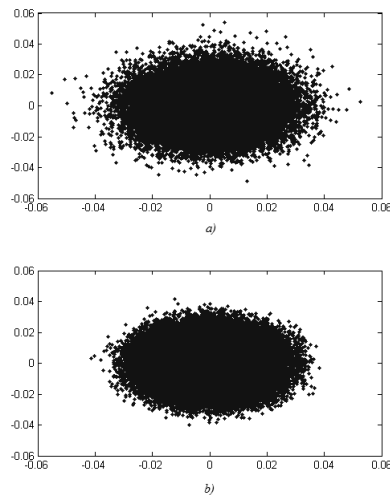

Fig. 2.3: QPSK constellation from a PolMux-QPSK (section 1.2): (a) received data constellation diagram without DSP, (b) received data constellation diagram with only CD compensation

2.2.1 Time domain technique

In time domain filtering corresponds to convolution operation which can be performed in digital domain by Finite Impulse Response (FIR) filter[10]. The schematic representation of it is in Fig. 2.4 and the formula for FIR filter is:

$$y(n) = b_0 x(n) + b_1 x(n-1) + ... + b_N x(n-N) \tag{2.5}$$

where x is the input signal, b_i are the filter weights or as it is also called tap weights, N is the filter length and y is the output.
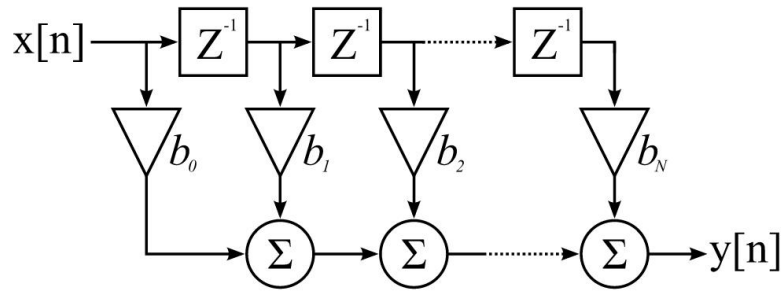
Fig. 2.4: Fir filter block

Two different time domain techniques for CD compensation were analyzed: Savory's method and blind adaptive equalizer such as *least mean square* (LMS) and *constant modulus algorithm* (CMA) algorithms.

Savory's method

In order to implement the CD filter in time domain (see the Matlab code in A.7), formulas should be derived for tap weights. As S.Savory demonstrated[6], expressions for tap weights and filter length come from Eq. 2.4 by truncating the impulse response to a finite duration. Since it is finite we can implement this digitally using a FIR filter where the number of the taps and the amplitude can be calculated using the formula:

$$a_k = \sqrt{\frac{jcT^2}{D\lambda^2 z}} e^{-j\frac{\pi cT^2}{D\lambda^2 z}k^2}, \tag{2.6}$$

with

$$-\left\lfloor \frac{N}{2} \right\rfloor \leq k \leq \left\lfloor \frac{N}{2} \right\rfloor, \tag{2.7}$$

20

and

$$N = 2 \times \left\lfloor \frac{|D|\lambda^2 z}{2cT^2} \right\rfloor + 1, \tag{2.8}$$

where N is the number of taps, T is the sampling period and $\lfloor x \rfloor$ is the integer part rounded toward $-\infty$.

Adaptive filters

An adaptive algorithm incorporates an iterative procedure (Figure 2.5) that makes successive corrections to the weight of the filter in order to minimize the mean error between the output $\hat{d}(\text{n})$ and the desired symbol $d(\text{n})$. An extremely important parameter is the step size μ that is linked with the filter taps weight uploading. High value of μ means faster convergence but lower precision. On the other hand low value of μ bring to slower convergence but higher precision. A good compromise is the use of high value of μ to initialize the filters taps and then switch to a low value to get an accurate precision. Least mean square (LMS) algorithm and constant modulus algorithm (CMA)[11] will be presented in the next sections.
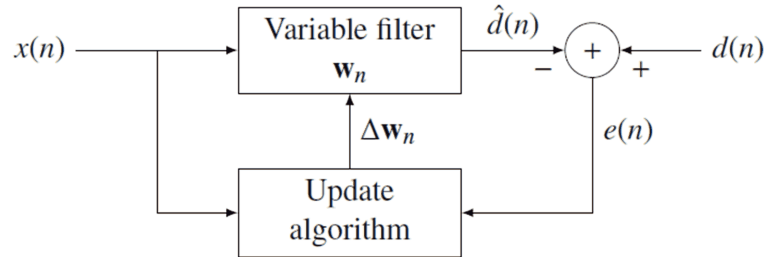
Fig. 2.5: Adaptive filter scheme

LMS method

Notation:

$$x(n) \quad \text{input} \tag{2.9}$$

$$w(n) \quad \text{filter tap weights} \tag{2.10}$$

$$d(n) \quad \text{desired response} \tag{2.11}$$

$$y(n) \quad = \quad \sum_{K=0}^{M-1} w_k^* x(n-k) \tag{2.12}$$

$$= \quad w^H(n)x(n) \tag{2.13}$$

$$\text{filter output}$$

$$e(n) \quad = \quad d(n) - y(n) \tag{2.14}$$

$$\text{error estimation}$$

$$w(n+1) \quad = \quad w(n) + \mu * x(n) * e^*(n) \tag{2.15}$$

$$\text{uploading equation}$$
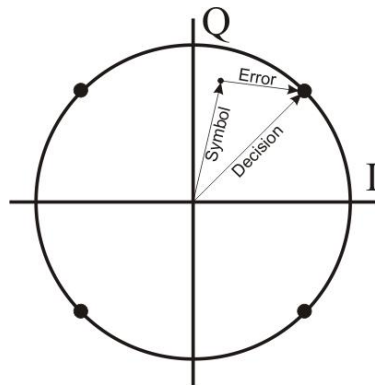
Fig. 2.6: LMS error evaluation

At every step the algorithm (see the Matlab code in A.4) uploads the filter weight following the Eq. 2.15 and it estimates the error comparing the

output at the step n with the desired response $d(\text{n})$ using a directed decision method (Fig. 2.6). The filter tap weights should be initialized. A solution is to set all the taps weight to zero except the central one that is set to unity corresponding to no filtering[6].

CMA method

Notation:

$$x(n) \qquad \text{input} \tag{2.16}$$

$$w(n) \qquad \text{filter tap weights} \tag{2.17}$$

$$y(n) \;=\; \sum_{K=0}^{M-1} w_k^* x(n-k) \tag{2.18}$$

$$ \;=\; w^H(n)x(n) \tag{2.19}$$

$$ \quad \text{filter output}$$

$$e(n) \;=\; 1 - |y(n)|^2 \tag{2.20}$$

$$ \quad \text{error estimation}$$

$$w(n+1) \;=\; w(n) + \mu * x^*(n) * e(n) * y(n) \tag{2.21}$$

$$ \quad \text{uploading equation}$$

After CD compensation for phase modulation signals the amplitude of the symbols is constant. For this reason the error is calculated imposing a constrain on the intensity (Eq. 2.20) and not using direct decision as in LMS case. As CMA doesn't take any decision, it brings better performance in terms of convergence than LMS as results will show in section 3. The taps weight are initialized as the LMS case.

23

2.2.2 Frequency domain technique

Fast Fourier Transform method

In frequency domain the filtering of the input stream is obtained by multiplication operation. The formula for the digital filter weights in frequency domain can be obtained from Eq. 2.3:

$$f_k = e^{j \frac{D\lambda^2}{4\pi c} z \omega_k^2} \tag{2.22}$$

where $\omega_k = \frac{2\pi}{T L_{FFT}}$, L_{FFT} is the *Fast Fourier Transform* (FFT) length and T is the sampling period (see the Matlab code in A.2). As the frequency domain
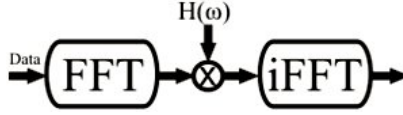
Fig. 2.7: FFT CD compensation algorithm scheme

compensation uses the FFT function it is not advantageous working with long sequence of data because it increases the complexity of the algorithm. The received data should be split in smaller windows but connecting directly different blocks after the compensation results in errors at the edges. For that reason the overlap-add method is used. The overlap-add method has this procedure[10]:

- decompose the received signal into smaller windows;

- each window is padded on the both side (grey part in Fig. 2.8) with a sequence of zero (*zero-padding technique*);

- the portion of the signal enters in the scheme shown in Fig. 2.7. FFT, multiplication with the filter and *inverse Fast Fourier Transform* (IFFT) is made;

24
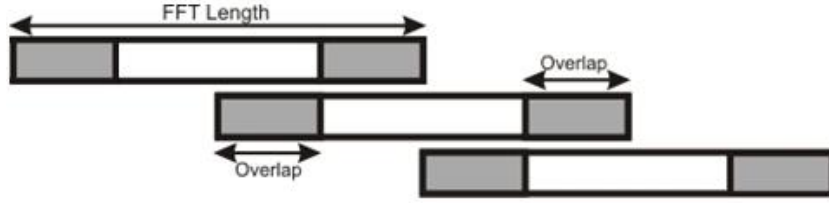
Fig. 2.8: Overlap add method

- the windows are recomposed overlapping the grey part as showed in the Fig. 2.8.

There are two crucial parameters in this method: the FFT length (that is equal to the filter length) and the zero padding length. The FFT length must be long enough to allow the compensation of the chromatic dispersion accumulated during the fiber propagation and, as FFT algorithm is defined, it must be a power of two. The zero padding length is related to the dispersive effect on the samples at the edge of the windows in which the signal is split. If the padding is not long enough the filtered portion of data will exceed the time window and this will rise to aliasing. If L is the number of samples of each window and G is the total overlap length (G/2 at the beginning of the window and G/2 at the end), we fix $L = FFT\ length - G$ with $G = \frac{FFT\ length}{2}$ (see matlab code in A.2).

One issue of our investigation was the calculation of the FFT length and the zero padding length that guaranties a right value of BER. The results will be presented in section 3.1.

2.3 Polarization Mode Dispersion compensation

Polarization tracking (section 1.3) can be archive using an adaptive CMA butterfly structure (see the Matlab code in A.6)[6][7]. The structure is com-
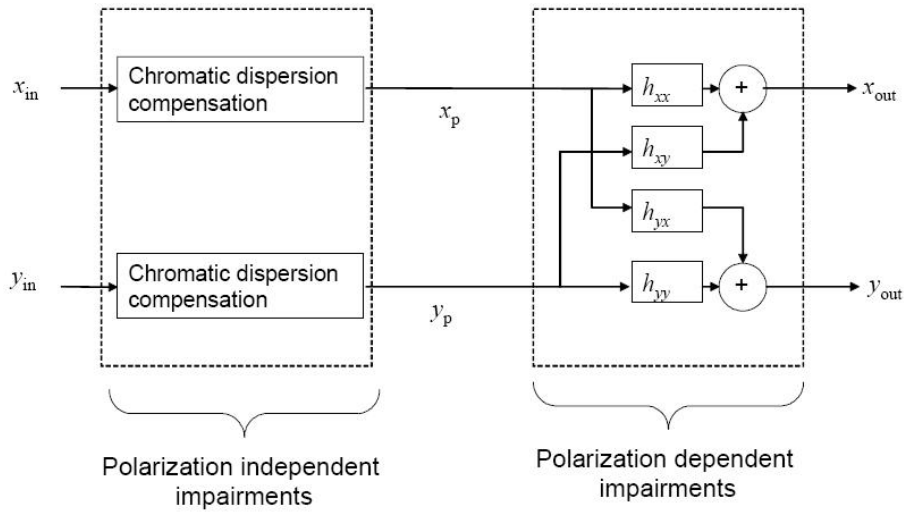
Fig. 2.9: Functionality of the digital filtering stage[6]

posed by four CMA adaptive filters (h_{xx}, h_{yx}, h_{xy} and h_{yy}) related each other by these formulas:

$$h_{xx} = h_{xx} + \mu \cdot e_x \cdot x_{out} \cdot x_p^* \qquad (2.23)$$

$$h_{xy} = h_{xy} + \mu \cdot e_x \cdot x_{out} \cdot y_p^* \qquad (2.24)$$

$$h_{yx} = h_{yx} + \mu \cdot e_y \cdot y_{out} \cdot x_p^* \qquad (2.25)$$

$$h_{yy} = h_{yy} + \mu \cdot e_y \cdot y_{out} \cdot y_p^* \qquad (2.26)$$

where x_{out} and y_{out} are given by:

26

$$x_{out} = h_{xx}^T \cdot x_p + h_{xy}^T \cdot y_p \qquad (2.27)$$

$$y_{out} = h_{yx}^T \cdot x_p + h_{yy}^T \cdot y_p \qquad (2.28)$$

e_x and e_y are the errors defined as:

$$e_x = 1 - |x_{out}|^2 \qquad (2.29)$$

$$e_y = 1 - |y_{out}|^2 \qquad (2.30)$$

Filters are initialized with all the tap weights to zero except for the central tap of h_{xx} and h_{yy} that are set to the unity[6]. As the equalizer is free from any constrain and it could happen that the equalizer converge to the same output (*singularity problem*). To avoid this it is possible to initialize h_{xx} and h_{xy} and run the algorithm then h_{yy} and h_{yx} can be set using the result obtained[12]:

$$h_{yy}(t) = h_{xx}^*(-t) \qquad (2.31)$$

$$h_{yx}(t) = -h_{xy}^*(-t) \qquad (2.32)$$

As during the first stage of CD compensation some residual CD could not be compensated this algorithm takes care of it. Fig. 2.10 shows the differences, on a QPSK constellation, with and without PMD compensation after the first stage of CD compensation.

Fig. 2.10: QPSK constellation from a PolMux-QPSK (section 1.2): (a) received data constellation diagram before PMD compensation, (b) received data constellation diagram after PMD compensation

2.4 Carrier phase recovery

The local oscillator at the receiver runs at a frequency which is only nominally equal to the transmitter laser. The relative small offset in frequency causes the detected signal to accumulate a phase error. This is taken care in the DSP by carrier phase recovery or phase noise compensation. During this step frequency and phase offset between local oscillator and signal is compensated using a "Viterby-and-Viterby" carrier phase estimation (CPE) method (see Matlab code in A.8) as it has been used by Chris R. S. Fludger[8]. Power of four is used to remove the quaternary phase modulation and for each symbols the phase shift value is averaged with the neighbors on both side using a sliding window of length N. The formula for the phase shift is:

$$\hat{\phi}(k) = arg\left(\sum_{n=-N}^{N} x^4(k+n) \right) \tag{2.33}$$

where $\hat{\phi}(k)$ is the phase correction that has to be applied to the symbol k.

When the spontaneous emissions (ASE) are dominant in the system, N should be higher to average the Gaussian noise and optimize the phase shift estimation. On the other hand in presence of higher phase noise a smaller number of N is preferred as it permits to follow change in phase with more precision. In all the simulations an optimization of N was performed.

Fig. 2.11 shows the differences, on a QPSK constellation, with and without carrier phase recovery after CD and PMD compensation.
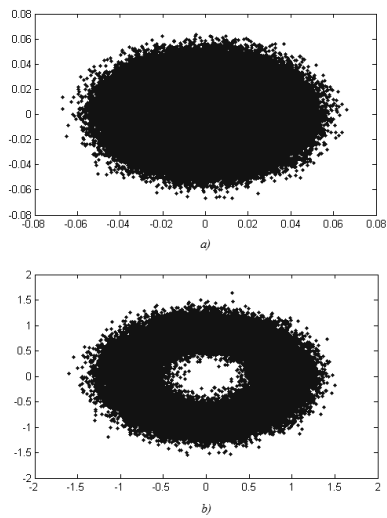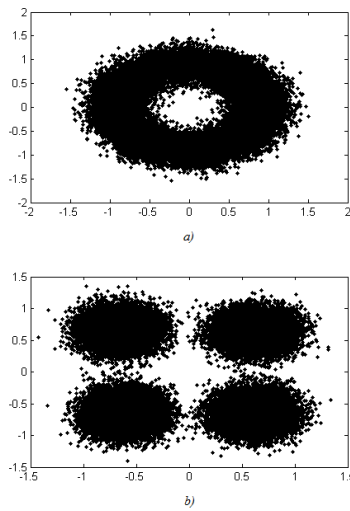
Fig. 2.11: QPSK constellation from a PolMux-QPSK (section 1.2): (a) received data constellation diagram before carrier phase recovery, (b) received data constellation diagram after carrier phase recovery

2.5 Nonlinear compensation

As explained in section in section 1.1.4, nonlinear effects become an important constrain as power increases. Eq. 1.9 and 1.10 show that the nonlinear phase shift is proportional to the signal intensity.

Nonlinear effects can be compensated using the received intensity. The results that will be presented in section 3.2 were obtained using a nonlinear compensator (NLC) based on the technique of multi-span back-propagation (see Matlab code in A.9)[13]. In long-haul transmission systems the signal is periodically amplified throughout transmission to compensate for fiber loss. The optical link is therefore composed by a number of *span* (Fig. 2.12) made by an optical amplifier, Erbium Doped Fiber Amplifier (EDFA), followed by an optical fiber. It can be approximated that nonlinearities are concentrated just after the optical amplifiers where the power is higher. In this method we simplify things further by assuming that the nonlinear shift takes place instantaneously at the fiber input. Therefore the method consists in compensating for the CD accumulated during the span first and then treat the nonlinear effects.
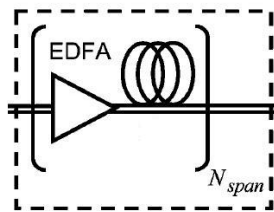
Fig. 2.12: Span[13]

These steps are then repeated for each span. The structure of the filter is a FIR filter for the CD compensation followed by a butterfly NLC structure called *NLC core* as showed in the figure 2.13 a).
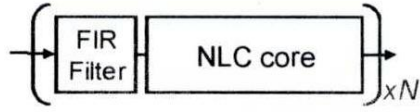
(a)
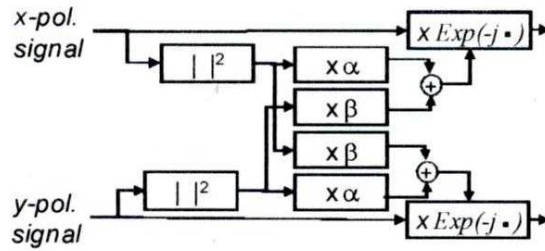
(b)

Fig. 2.13: Nonlinear compensator[14]

The NLC, presented by T. Tanimura [14], introduces a phase shift on both of the digital streams according to these formulas, as represented by Fig. 2.13 b):

$$E_x^{out} = E_x^{in} e^{-j(\alpha I_x + \beta I_y)} \tag{2.34}$$

$$E_y^{out} = E_y^{in} e^{-j(\alpha I_x + \beta I_y)} \tag{2.35}$$

where $I_x = |E_x|^2$, $I_y = |E_y|^2$, α is the intra-polarization nonlinearity parameter and β is the inter-polarization nonlinearity parameter and have to be optimized. In section 3.2 optimization result will be presented.

Chapter 3

Algorithm testing and results

In this chapter results of the DSP algorithm tests will be presented. It is divided in two part. The first part is focused on the testing of CD compensation methods presented in section 2.2 using optical data generated with the simulation tool VPItransmissionMaker. The second part deals with nonlinear compensation using the result obtained in the first section on optical data generated in a laboratory experiment.

3.1 Chromatic dispersion compensation

3.1.1 System setup

The setup of the 112 Gb/s PolMux-QPSK coherent optical transmission system established using VPItransmissionMaker is illustrated in Fig. 3.1. The electrical data from four 28 Gb/s pseudo random bit sequence (PRBS) generators are modulated into two orthogonally polarized QPSK optical signals by two Mach-Zehnder modulators, which are then integrated into one fiber transmission channel by a polarization beam combiner to form the PolMux-QPSK optical signal. The received optical signals are mixed, with an optical

local oscillator (LO) in the coherent receiver, to be transformed into four electrical signals by balanced photodetectors. In the dual polarization QPSK
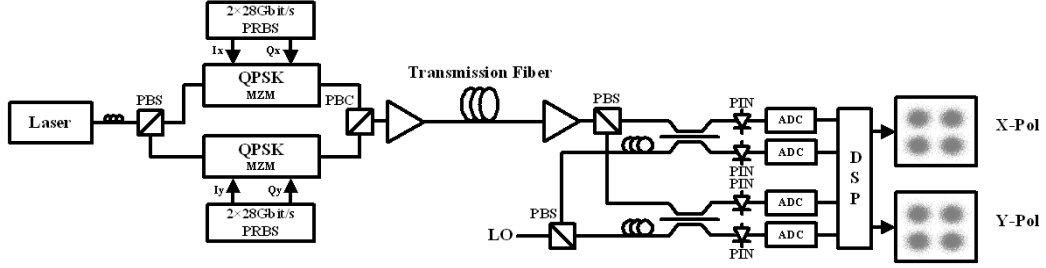
Fig. 3.1: Schematic of 112 Gbit/s PM-QPSK coherent transmission system

transmitter, the number of bits output from the PRBS generator in each polarization is 65536 (2^{16}), and the number of symbols in four components (I_x, Q_x, I_y, Q_y) is 32768 (2^{15}). The linewidth of the transmitter laser and the LO laser as the the offset between the transmitter laser and the LO laser are set to be 0 Hz (i.e. no phase noise). In the fiber, the attenuation is set to 0 as well as PMD and the nonlinear index. The chromatic dispersion coefficient is set to be 16 ps/nm/km. In the coherent receiver, the 2×4 90° hybrid structure is adopted to demodulate the received optical signal, which consists of 3 dB 2×2 fiber couplers and a phase delay components of $\pi/2$ phase shift in one branch. The goal of these simulations is to understand how the different CD compensation methods behave and a good procedure is to neglect all the effects except for CD. The back to back (transmitter and receiver, without any transmission line) OSNR level required for a bit error rate (BER) of 10^{-3} (limit value that allow full error recovery using forward error correction techniques) is 14.8 dB. As our goal is to fully compensate for CD, this value of required OSNR is expected for all the propagation distances.

34

3.1.2 Time domain methods

Savory's method

Savory's filter is implemented using equation 2.6, 2.7 and 2.8 (section 2.2.1). Fig 3.2 and Fig. 3.3 show respectively OSNR required for a BER of 10^{-3} versus propagation distance and number of taps required. As Savory's formula is an approximation of equation 2.4 for short distance there is a penalty and the algorithm is not performing totally the CD compensation (Fig. 3.2). For longer distance it start to perform perfectly but we still have a penalty for 1000 and 5000 km as a matter of fact our simulations have only chromatic dispersion and it is expected to be fully compensated. The number of taps increase linearly as the taps equation (Eq. 2.8) is linearly dependent on the propagation length.

Adaptive filters

Using a LMS adaptive filter it is possible to reach some expected OSNR level required for BER of 10^{-3} as for back to back. This algorithm needs a time to converge and for longer distance it becomes a problem as a higher number of taps is needed. The issue was solved for 200 km using CMA algorithm that has better performance in convergence (section 2.2.1) and increasing the number of received symbols replying the received sequence. The longer the input sequence is, more convergence time the adaptive algorithm has. For 200 km it is enough to repeat the received data three time (65536×3 samples) but increasing the distance to 1000 km, the filter length required is too long and even CMA convergence is not reached.
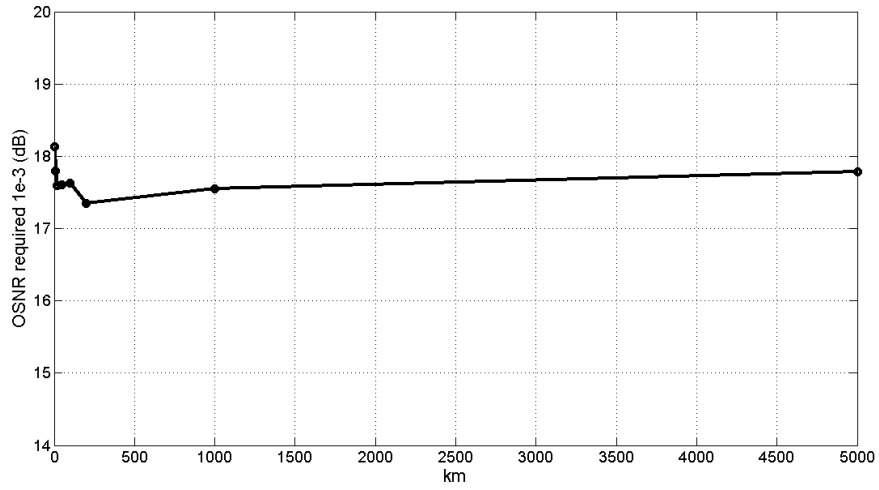
Fig. 3.2: OSNR required for BER of 10^{-3} versus the propagation distance using the Savory's method
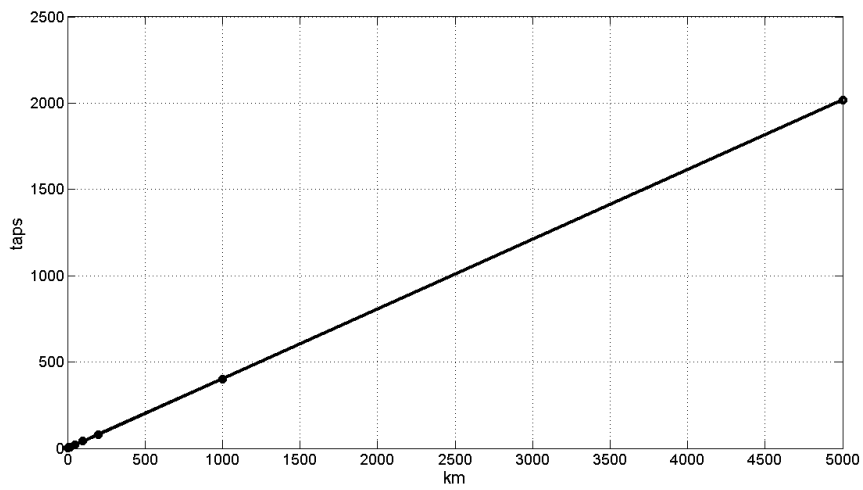
Fig. 3.3: Number of taps required using the Savory's method

36
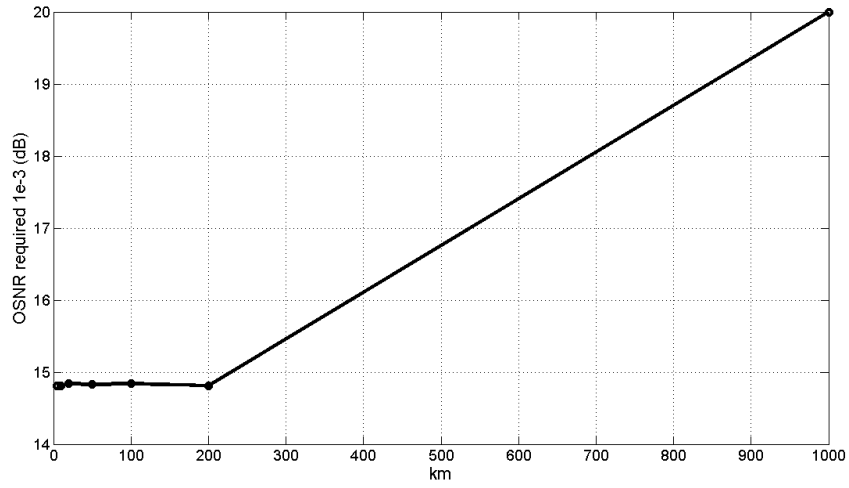
Fig. 3.4: OSNR required for BER of 10^{-3} versus the propagation distance using CMA adaptive filter
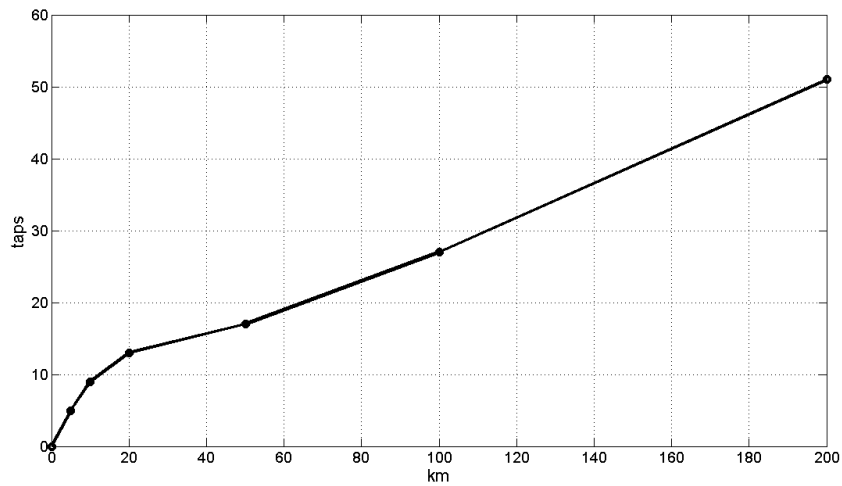
Fig. 3.5: Number of taps required CMA adaptive filter

Savory + adaptive filter (CMA)

In this section CD compensation is performed with a combination of Savory method and CMA adaptive filter. First Savory's filter is used to compensate most of the dispersion and then CMA adaptive filter with few taps is used for the residual dispersion. This structure is closer to a real system where a fixed long filter is used for the first stage of CD compensation and then the residual CD is treated by a shorter one. As in VPItransmissionMaker system the signal is sampled twice the symbol rate at the receiver, the algorithm has to chose the best sample to be used for the BER calculation. This decision is taken after the first CD compensation stage by plotting all the first and the second samples separately and choosing the one with the highest mean intensity. The best sample is passed to the CMA routine so that the filter weights are uploaded every 2 samples and downsampling is performed.
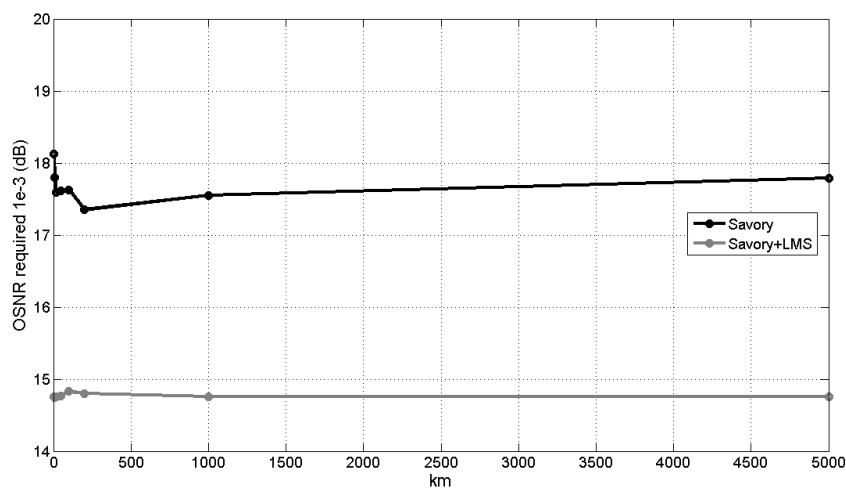
Fig. 3.6: OSNR required for BER of 10^{-3} versus the propagation distance using Savory's and CMA filters

Using Savory + CMA method leads to a required BER equal to back to

38

back. The number of taps required for the adaptive filter is related to the Savory's filter behavior. As for short distance Savory's method approximation is not accurate (Fig. 3.2), CMA required more taps while for more the 200 km only 7 taps are needed. This problem is avoided, as will be showed in the next section, using the frequency domain compensation.

3.1.3 Frequency domain method

As explained in section 2.2.2 an investigation on the length of the frequency filter was made to understand the behavior and the performance of this algorithm. The overlap-add method was firstly tested over three different propagation distance for a fixed OSNR level. FFT length was change to find the minimum required length to get the convergence of the BER value. As Fig. 3.7 shows for 20, 100 and 200 km of propagation FFT length of 32, 128 and 256 is required.
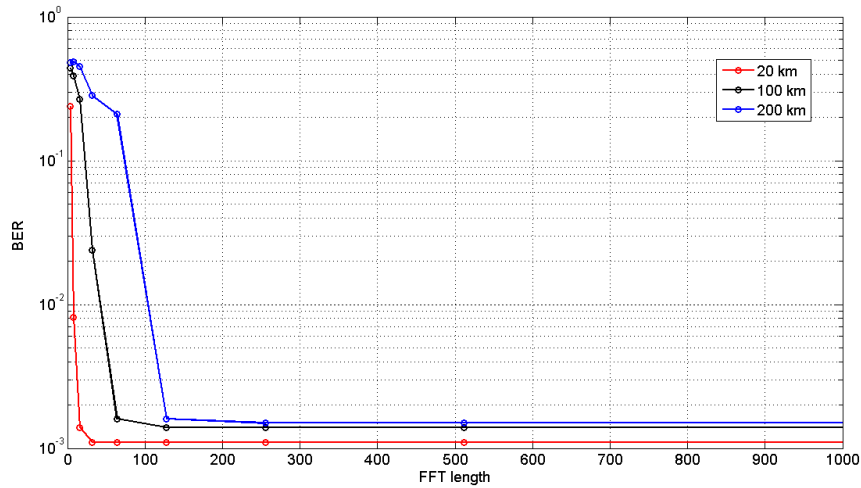
Fig. 3.7: FFT length in overlap add method

Fig. 3.8 shows the required OSNR versus the propagation distance for
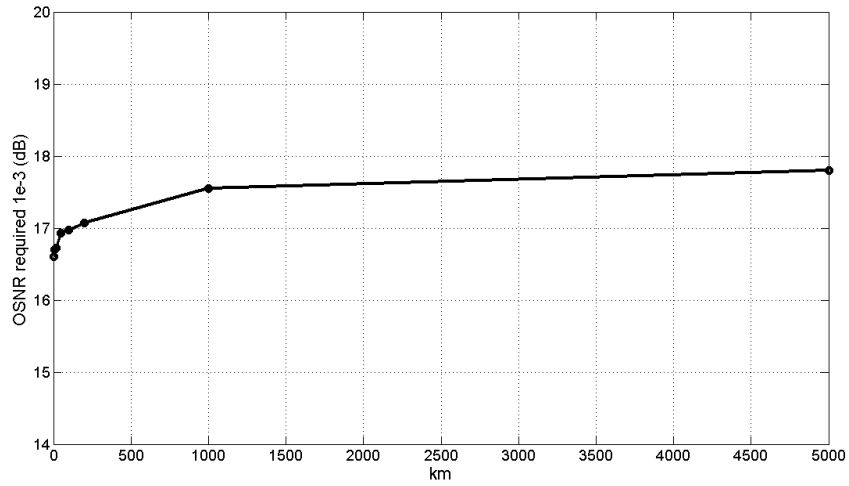
Fig. 3.8: OSNR required for BER of 10^{-3} versus the propagation distance using FFT method
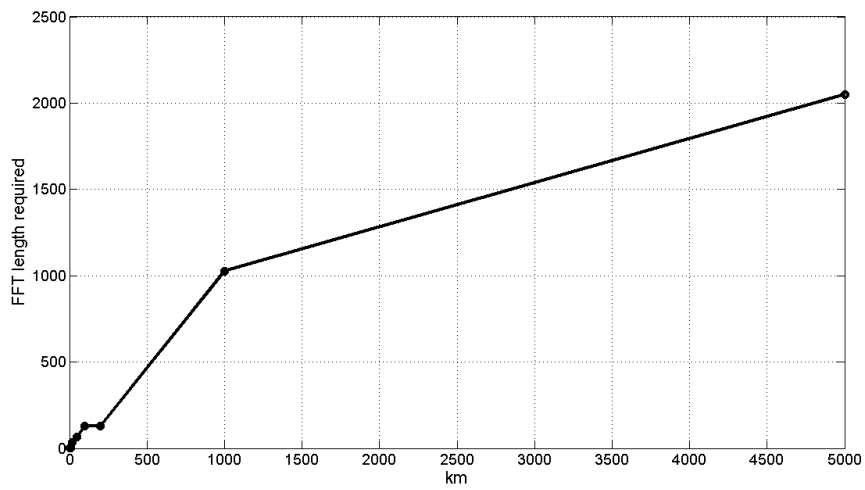
Fig. 3.9: FFT length required versus the propagation distance

chromatic dispersion compensation performed using only the frequency domain method. Even for 5 km the results found in back to back were not reached and increasing the distance a strong penalty appeared. For 5000 km we measured a 3 dB penalty compared with the 14.8 dB required for 5 km propagation with the adaptive filter compensation (Fig. 3.8).

Frequency domain method + CMA

As for the case Savory's filer + CMA, chromatic dispersion compensation was also performed with a combination of FFT overlap add method and then CMA filter and the same result were reached (Fig. 3.10).

The CMA filter required only 7 taps that are needed for 500 km propagation while for shorter distance even 5 or 3 taps are necessary.
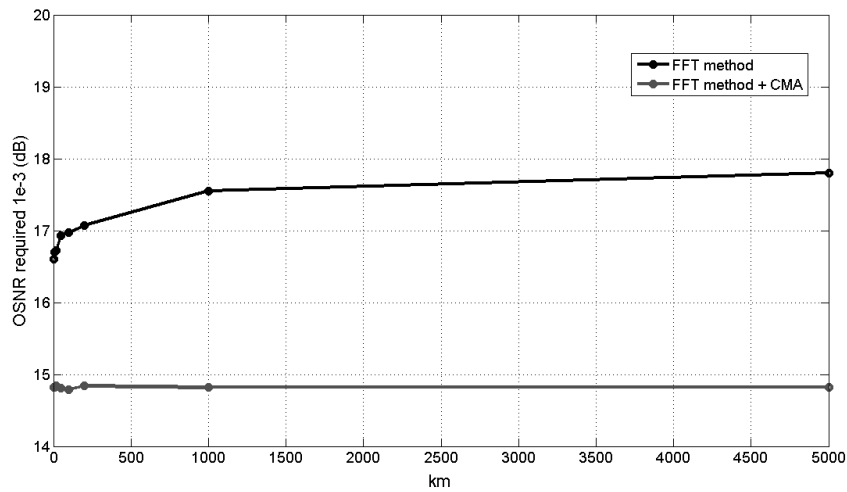
Fig. 3.10: OSNR required for BER of 10^{-3} versus the propagation distance using FFT method and CMA filter

3.2 Nonlinear effects compensation

3.2.1 System setup

For the nonlinear impairments compensation, data from a 100 Gb/s PolMux-QPSK coherent optical transmission system were provided by Politecnico of Torino within the framework of the EURO-FOS[1] project.

The transmission channel consists of a loop containing a SSMF long 63.6 km. Optical EDFA amplifiers were used for amplifying the signal after each fiber. For different experiments different number of loops can be used to emulate long-haul network. The fiber dispersion is 16.3 ps/nm/km. The receiver is composed of a polarization beam splitter (PBS) followed by 90° hybrids coupler and balanced photodetectors. Data is then stored in a digital oscilloscope for offline processing. The sampling of data is performed at the speed of 2 samples per symbols.

Two different type of data were provided: single channel system and 16 channels WDM system with 50 GHz spacing.

3.2.2 Single channel result

Single channel data were composed by different measures taken for 8, 16, 24 and 32 loops (respectively 508.8, 1017.6, 1526.4 and 2035.2 km) with two different OSNR levels 15 dB and 20 dB.

The DSP code used (A.1) is composed by:

- NLC and CD compensation using backpropagation (presented in section 2.5, see code in A.9):

    - frequency domain filter with FFT length set to 128;

---

[1]http://www.euro-fos.eu/

– NLC core.

- CMA butterfly structure with 25 taps for each filter (presented in section 2.3, see code in A.6);

- phase noise compensation using Viterby&Viterby method (presented in section 2.4, see code in A.8);

The α and β parameter (Eq. 2.34 and 2.35) must be optimize[14]. As figure 3.11 shows, α and β value were swept from 0 to 50. The best BER was found for $\alpha = \beta = 25$.
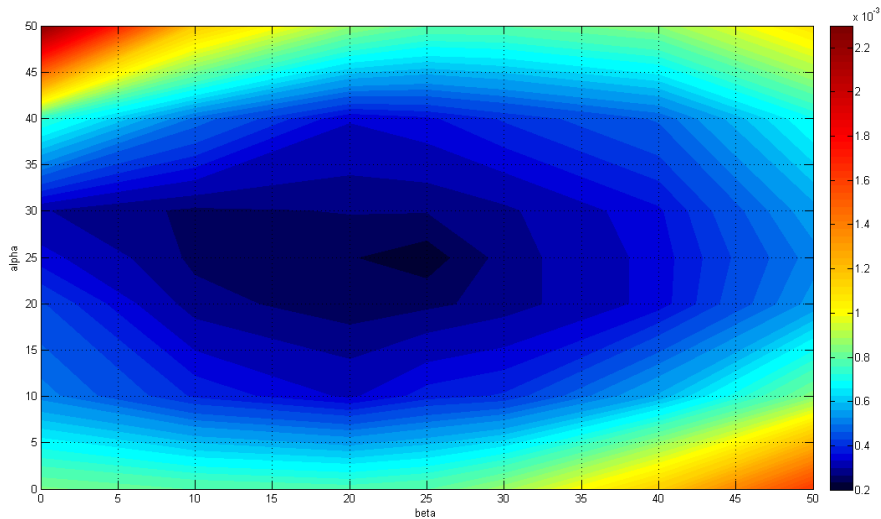
Fig. 3.11: BER dependency on intra-polarization nonlinear parameter (α) and interpolarization parameter (β)

The result with and without the NLC are shown in figure 3.12. The input power level in the fiber is -3dBm. From the graph it can be seen that, as the power level and nonlinearities are not so high, the improvement is only 1 dB for 200 km propagation. The blue line in figure 3.12 was compared to

result obtained without NLC by Politecnico of Torino[2].
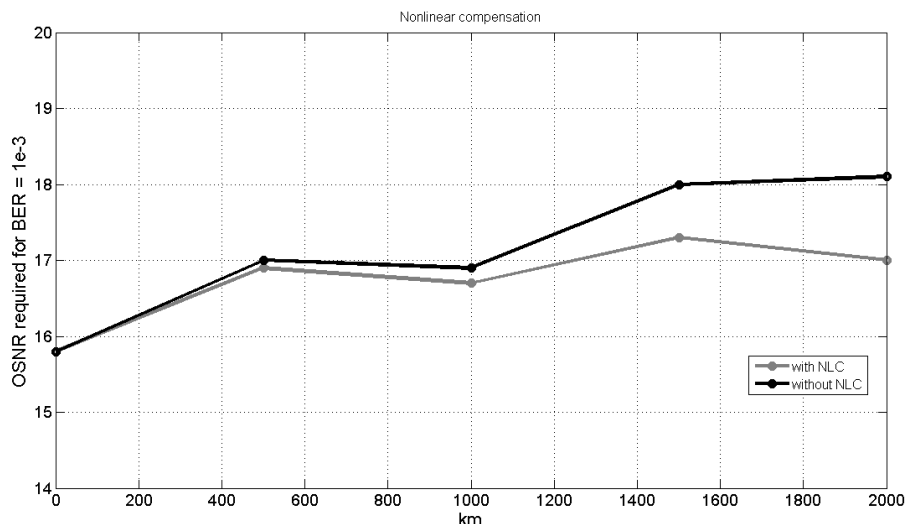
Fig. 3.12: OSNR required for a bit error rate (BER) of 10^{-3} versus the propagation distance with and without NLC

3.2.3 Multichannel result

In this section data for 16 channels WDM system with the same characteristic, as in 3.2.1, is processed. The channels are 50 GHz spacing and data is again provided by Politecnico of Torino[3]. Several measurements have been performed, for the central channel, a 1017.6 km (16 loops) propagation, a fixed attenuation per span and varying the power launched in the fiber form -1.2 dBm to 1.6 dBm. The DSP algorithms are the same used in section 3.2.2 with an α and β set to 10.

As the figure 3.13 shows there is a little improvement using the NLC (as we said in the previous section the power level is not so high). If the nonli-

---

[2]http://www.optcom.polito.it/research/Experimental_files/exp_files.htm

[3]http://www.optcom.polito.it/research/Experimental_files/eurofos.htm

nearities were all compensated for, the blue line should continuing decreasing with the same slope even for a power of 1.6 dBm. Using the nonlinear structure, as there is no information about the neighbors channel, only SPM can be compensated while the penalty for 1.6 dBm comes from inter-channel nonlinearities as discussed in section 1.10.
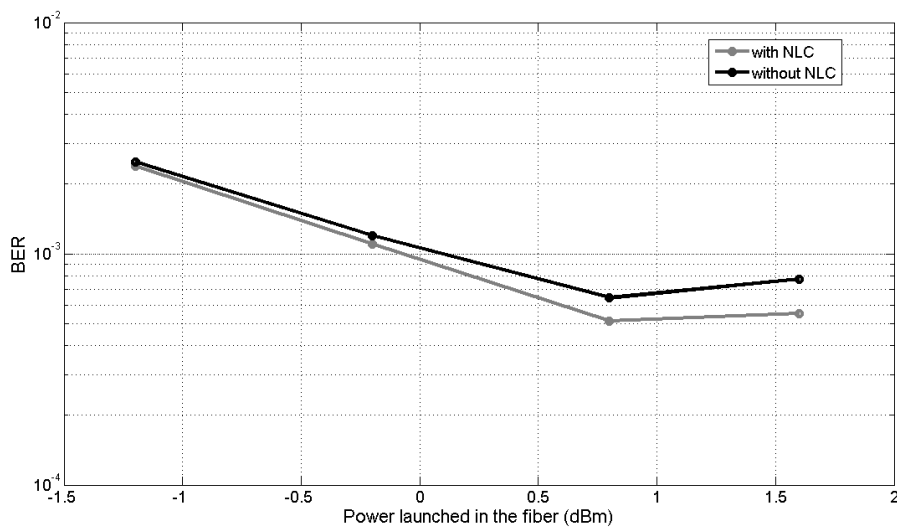
Fig. 3.13: BER versus power launched in the fiber with and without NLC

Chapter 4

Conclusion and future work

Coherent detection at 100 Gb/s will become feasible in the future using DSP for linear impairments compensation. DSP allow polarization de-multiplexing, compensation of linear transmission impairments, as CD and PMD, and also gives higher OSNR tolerance. Increasing OSNR tolerance leads to increasing the maximal propagation distance that means less optical amplifiers, less noise and less costs of the system. On the other hand a more complex receiver is required as polarization tracking has to be performed but this is well paid by the improvement on system performance. The use of NLC (section 2.5) in DSP algorithm brings also important improving in performance but when inter-channel nonlinearities become predominant this advantage decreases.

Considering the future work, there are a lot of issue that can be addressed. First is to implement more complex and advanced adaptive algorithms for the PMD and residual CD compensation (especially step size updating algorithms) compared to the one used in section 2.3. Investigation over nonlinear effects compensation would also be interesting. As result in section 3.2.3 shows, multi-span backpropagation technique doesn't compensate for inter-

channel nonlinearities. Advanced techniques based on Volterra compensator are interesting because can avoid this problem.

Bibliography

[1] Marco Forzati. *Phase Modulation Techniques for On-Off Keying in Optical Fiber.* PhD thesis, Chalmers University of Technology, 2007.

[2] Govind P.Agrawal. *Nonlinear Fiber Optics.* Academic Press Second edition, 1995.

[3] Daniel J. F. Barros Joseph M. Kahn Ezra Ip., A. Pak Tao Lau. Coherent detection in optical fiber systems. *Optics Express Vol. 16, No. 2*, January 2008.

[4] M.Santagiustina. *Progettazione e simulazione di circuiti ottici*, 2009.

[5] Michael Finkenzeller. Delivering 100G per wavelength with today's DWDM infrastructure. Motivation, Experiments and Standards RIPE 55, Amsterdam, October 2008. http://www.ripe.net/ripe/meetings/ripe-55/presentations/finkenzeller-delivering-100g.pdf.

[6] Seb J. Savory. Digital filters for coherent optical receivers. *Optic Express 804, Vol. 16, No. 2*, January 2008.

[7] Robert I. Killey P. Bayvel Seb J. Savory, G. Gavioli. Electronic compensation of chromatic dispersion using a digital coherent receiver. *Optic Express 2120, Vol. 15, No. 5*, March 2007.

[8] T. van den Borne D. Schulien C. Schmidt E.D. Wuth T. Geyer J. De Man E. Khoe Giok-Djan de Waardt Fludger, C.R.S. Duthel. Coherent equalization and POLMUX-RZ-DQPSK for robust 100-GE transmission. *Journal of lightwave technology, Vol. 26, No. 1*, January 2008.

[9] S.Haykin. Signal processing: where physics and mathematics meet. *IEEE Signal process, MAG18*, 2001.

[10] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing, Cap. 14,18*.

[11] M.Montazeri P.Duhamel and K.Hilal. Classical adaptive algorithms (LMS, RLS, CMA, Decision Directed) seen as recursive structure. *IEEE*, 1993.

[12] Weizhen Yan Oda S. Hoshida T. Rasmussen J.C. Ling Liu, Zhenning Tao. Initial tap setup of constant modulus algorithm for polarization de-multiplexing in optical coherent receivers. *OSA/OFC/NFOEC*, 2009.

[13] Ezra Ip and Joseph M. Kahn. Compensation of dispersion and nonlinear impairments using digital backpropagation. *Journal of lightwave technology, Vol. 26, No. 20*, October 2008.

[14] Oda S. Tanaka T. Ohsima C. Zhenning Tao Rasmussen J.C. Tanimura T., Hoshida T. Systematic analysis on multi-segment dual-polarisation nonlinear compensation in 112 Gb/s DP-QPSK coherent receiver. *ECOC*, September, 2009.

[15] Noriaki Kaneda and Andreas Leven. Coherent polarization-division-multiplexed QPSK receiver with fractionally spaced CMA for PMD com-

pensation. *IEEE Photonics technology letters, Vol. 21, No. 4*, February 2009.

[16] J. Renaudier et al. Transmission of 100Gb/s coherent PDM-QPSK over 16x100km of standard fiber with allerbium amplifiers. *IEEE Photonics technology letters, Vol. 21, No. 4*, February 2009.

[17] Gabriel Charlet. Coherent detection associated with digital signal processing for fiber optics communication. *C. R. Physique 9*, 2008.

[18] Young-Kai Chen Andreas Leven, Noriaki Kaneda. A real-time CMA-based 10 Gb/s polarization demultiplexing coherent receiver implemented in an FPGA. *OFC/NFOEC*, 2008.

[19] I. Fijalkow J.R. Treichler and JR C.R. Johnson. Fractionally spaced equalizers. *IEEE Signal Processing Magazine*, 1996.

[20] I. Fijalkow J.R. Treichler and JR C.R. Johnson. Chromatic dispersion compensation using digital IIR filtering with coherent detection. *IEEE Photonics technology letters, Vol. 19, No. 13*, July 2007.

[21] Kiyoharu Aizawa Sanghoon Song and Mitsutoshi Hatori. A blind adaptive array based on CMA and LMS. *Electronics and Communications in Japan, Part 3, Vol. 81, No. 7*, 1998.

[22] Sander Lars Jansen Torsten Wuth Maxim Kuschnerov Guido Grosso Antonio Napoli Mohammad S. Alfiad, Dirk van den Borne and Huug de Waardt. A comparison of electrical and optical dispersion compensation for 111-Gb/s POLMUX-RZ-DQPSK. *Journal of lightwave technology, Vol. 27, No. 16*, August 2009.

Appendix A

DSP test code

A.1   main.m

```matlab
%Main program used

%The algorithm is composed by a series of module that can be switch on or
      off as needed.
5
clear all functions; clc
close all

%% exp. parameters
10      SymbolRate =   ; % Baud
        SamplesPerSymbol =   ;
        LoopDispersion =   ; % [ps/nm]
        RefWavelength =   ; % [m] ref. wavelength for dispersion compensation
        distance =   ; %km
15      osnr =   ;
        fft_length =   ; %must be a power of 2

        %% module activation
        %1 --> active
20
        frequency_domain_CD = 0
        time_domain_CD = 0
```

```matlab
        NLC = 0
        phase_noise_compensation = 0
25      PMD_compensation = 0

    % number of bits to read and start sample

        NSymbols= ;
30      StartSample= ;
        EndSample = StartSample + NSymbols*SamplesPerSymbol;

    %% read data

35      file_name = ['./' num2str(distance) 'km/' num2str(LoopDispersion) 'ps'
                num2str(distance) 'km_' num2str(osnr) 'dB.mat'];

        load(file_name)

        read_Ix = Ix.band.E(StartSample:EndSample)';
40      read_Qx = Qx.band.E(StartSample:EndSample)';

        read_Iy = Iy.band.E(StartSample:EndSample)';
        read_Qy = Qy.band.E(StartSample:EndSample)';

45      read_IQx=complex(read_Ix,read_Qx);
        read_IQy=complex(read_Iy,read_Qy);

    %% resample
    % it use the matlab function resample
50
        NewSamplesPerSymbol= ;

        data_IQx=resample(read_IQx,NewSamplesPerSymbol, SamplesPerSymbol);
        data_IQy=resample(read_IQy,NewSamplesPerSymbol, SamplesPerSymbol);
55
    %% Cromatic dispersion compensation in frequency domain

        if frequency_domain_CD == 1

60          sample_period=1e12/(SymbolRate*NewSamplesPerSymbol); %recovering
                    sampling period in 'ps'
            Dispersion=LoopDispersion*distance;
```

54

```matlab
                %CD_fil_fde generates the filter and perform the filtering in the
                %frequency domain
65              [data_IQx data_IQy]=CD_fil_fde(data_IQx,data_IQy, fft_length ,
                        Dispersion , RefWavelength*1e9 , sample_period );


            %%%%% Nonlinear compensator


            if NLC == 1

70
                alpha =  ;
                beta =  ;
                %they need to be optimized


75              data_IQx=data_IQx.*exp(-i*(alpha*(abs(data_IQx)).^2+beta*(abs(
                        data_IQy )).^2));
                data_IQy=data_IQy.*exp(-i*(alpha*(abs(data_IQy)).^2+beta*(abs(
                        data_IQx )).^2));


            end


80      end


    %% Sample data
    % It chooses the best sample looking at the highest average intensity of
    % the samples

85
    FirstSample_x=sample_data_adap(data_IQx , NewSamplesPerSymbol ,
            CalcStartSample );
    FirstSample_y=sample_data_adap(data_IQy , NewSamplesPerSymbol ,
            CalcStartSample );



90  %% PMD compensation with butterfly CMA structure


    if    PMD_compensation == 1


        %% avoid singularity problem

95
        %initialization of the filer weights
        tap_num_1= ; % even number (the real number of taps will be tap_num_1+1)
```

```matlab
        w_1=zeros(tap_num_1+1,1);
        w_1(tap_num_1/2+1)=1;
100     w_2=zeros(tap_num_1+1,1);
        w_mat=[w_1 w_2];

        %preparing data for downsampling inside the butterfly structure
        if FirstSample_x == 2
105         data_IQx = data_IQx(2:end);
        else
            data_IQx = data_IQx(1:end-1);
        end
        if FirstSample_y == 2
110         data_IQy = data_IQy(2:end);
        else
            data_IQy = data_IQy(1:end-1);
        end

115     amp_x=mean(abs(data_IQx));
        amp_y=mean(abs(data_IQy));

        data_IQx=data_IQx/amp_x;
        data_IQy=data_IQy/amp_y;
120
        myu= ;
        [fake_x err_x w]=pmd_cma_samp_sing_prob(data_IQx, data_IQy, tap_num_1,
             myu, w_mat);

    %% linear effect compensation with butterfly structure CMA
125
        w11=w(:,1);
        w12=w(:,2);
        w21=-conj(flipud(w12));
        w22=conj(flipud(w11));
130
        w_mat=[w11 w21 w12 w22];

        myu= ;
        [data_IQx data_IQy err_x err_y w_mat]=pmd_cma_samp(data_IQx, data_IQy,
             tap_num_1, myu, w_mat);
135
    end
```

56

```matlab
%% Phase noise compensation

140        if phase_noise_compensation == 1

             window= ;
             data_IQx=phase_comp(data_IQx,window);
             data_IQy=phase_comp(data_IQy,window);

145
         end


    %% data recovery

150        rec_Ix=real(data_IQx)>0;
         rec_Qx=imag(data_IQx)>0;

         rec_Iy=real(data_IQy)>0;
         rec_Qy=imag(data_IQy)>0;

155
         all_data_x=rec_Ix*2-1+j*(rec_Qx*2-1);
         all_data_y=rec_Iy*2-1+j*(rec_Qy*2-1);


    %% BER calculation

160
         BERstart=1000; %calculate BER cutting the first samples

         all_data_x=all_data_x(BERstart+1:end).';
         all_data_y=all_data_y(BERstart+1:end).';

165
         [BER_diff_x BER_diff_y errs_x errs_y length_x length_y]=BER(all_data_x,
               all_data_y, BitSequence_x, BitSequence_y);


    end % end of blocking routine
```

A.2 CD_fil_fde.m

%% OVERLAP–ADD METHOD

```
%CD_fil_fde CD compensation in frequency domain.
% —— Algorithmic details ——
%This function construct the frequency filter and call the Overlap−add
%method main function of filtering using FFT.
%datax, datay are the input data stream.
%fft_length is the length of the FFT, L is the length of the data window
%that we consider and G is the total zero pad length that is applied before
        and
%after the data window.

function [data_IQx data_IQy]=CD_fil_fde(datax, datay, fft_length, dispersion
        , wavelength, sample_period)


m = size(datax, 1);
    if m == 1
        datax = datax(:);
        datay = datay(:);
    end


    m = size(datax, 1);
    if m == 1
        datay = datay(:);
    end


    ndata = size(datax,1);


    if fft_length > 2^20,
        err(generatemsgid('filterTooLong'), ...
            'Filters of length greater than 2^20 are not supported. Use dfilt.
                fftfir instead.');
    end


    if fft_length > ndata
        err(generatemsgid('filterTooShort'), ...
            'Filter length must be shorter than data length');
    else
```

58

```matlab
%% L and G calculation

40  G = fft_length/4;
    L =fft_length/2;


    %% constuction of the filter
    %here the filter is constructed using the frequency domain formula directly
45  %from the theory


    C=3e5; %speed of light in km/s
    q=[-fft_length/2:fft_length/2-1]';
    omega=2*pi*(q)/(sample_period*fft_length);
50  filt=exp(-j*(dispersion*(wavelength)^2/(4*pi*C))*omega.^2);


    %fil is a structure where are saved all the parameters needed for perform
            the overlap method
    fil = struct('filter', filt,'nfft', fft_length, 'window', L,'ndata', ndata,
            'overlap' , G);


55  %main program
    data_IQx = overlap_add(fil,datax);
    data_IQy = overlap_add(fil,datay);


    end
```

A.3 overlap_add.m

```matlab
%% Overlap-add method of FIR filtering using FFT.

%   --- Algorithmic details ---
% b is a structure where are stored all the parameter needed to perform
%     the overlap
% add method.
% b = struct('filter', ..., 'nfft', ..., 'window', ..., 'ndata', ..., '
%     overlap', ...)

% filter is the filter defined in frequency domain, nfft is the FFT
% length, window is the window data length, ndata is the length of all
% the received data, overlap is the zero padding length defined as nfft/4

% The overlap/add algorithm convolves b.filter with blocks of data, and
%     adds
% the overlapping output blocks. It uses the FFT to compute the
% convolution.

function y = overlap_add(b,data)

%% loading parameters

nfft=b.nfft;
ndata=b.ndata;
L=b.window;
G=b.overlap;

B = fftshift(b.filter);

if length(B)==1,
    B = B(:);  % make sure that B is a column
end
if size(B,2)==1
    B = B(:,ones(1,size(data,2)));  % replicate the column B
end
if size(data,2)==1
    data = data(:,ones(1,size(b,2)));  % replicate the column data
end
```

60

```matlab
    y = zeros(size(data));

    istart = 1;
40      while istart <= ndata
            iend = min(istart+L-1,ndata);
            if (iend - istart) <= 0
                break
            else
45              X = fft([zeros(1,G)'; data(istart:iend,:)],nfft);
            end
            Y = ifft(X.*B);
            yend = min(ndata,istart+nfft-1);
            y(istart:yend,:) = y(istart:yend,:) + Y(1:(yend-istart+1),:);
50          istart = istart + L;
        end

    end
```

A.4   lms.m

```
function [y err w]=lms(x, N, FirstSample, myu, w)
% input data stream, filter length, best sample chosen looking
% the higher mean intensity, step size, tap weights

5

for q=N+FirstSample:2:length(x)
    X=x(q:-1:(q-N));
    y(q)=w'*X;

10      y_re=real(y(q))>0;
        y_re_1=y_re*2-1;
        y_im=imag(y(q))>0;
        y_im_1=y_im*2-1;

15      dec=(y_re_1+j*y_im_1);

        err(q)=dec-y(q);
        w=w+myu*X*conj(err(q));
    end
20  err=err(N+FirstSample+200:2:length(x)).';
    y=y(N+FirstSample+200:2:length(x)).';

    end
```

A.5 cma.m

```matlab
function [y err w]=cma(x , N, FirstSample , myu, w)
% input data stream, filter length, amplitude, best sample chosen looking
% the higher mean intensity, step size, tap weights

y=zeros(length(x),1);
for q=N+FirstSample:2:length(x)
    X=x(q:-1:(q-N));
    y(q)=X.'*w;


    err(q)=1-(abs(y(q)))^2;
    w=w+myu*conj(X)*err(q)*y(q);


end
err=err(N+FirstSample+200:2:length(x));
y=y(N+FirstSample+200:2:length(x));


end
```

A.6 pmd_cma_samp.m

```matlab
function [x_new y_new err_x err_y w]=pmd_cma_samp(x, y, N, myu, w, key) %
        input data streams, filter length, amplitude, tap weights

  w_xx=w(:,1);
5 w_yx=w(:,2);
  w_xy=w(:,3);
  w_yy=w(:,4);

  for q=N+1:2:length(x)
10     X=x(q:-1:(q-N));
       Y=y(q:-1:(q-N));

       x_new(q)=w_xx.'*X+w_xy.'*Y;
       y_new(q)=w_yx.'*X+w_yy.'*Y;
15
       err_x(q)=1-(abs(x_new(q)))^2;
       err_y(q)=1-(abs(y_new(q)))^2;

       w_xx=w_xx+myu*err_x(q)*x_new(q)*conj(X);
20     w_xy=w_xy+myu*err_x(q)*x_new(q)*conj(Y);
       w_yx=w_yx+myu*err_y(q)*y_new(q)*conj(X);
       w_yy=w_yy+myu*err_y(q)*y_new(q)*conj(Y);

  end
25
  x_new=x_new(N+1:2:end).';
  y_new=y_new(N+1:2:end).';

  err_x=err_x(N+1:2:end);
30 err_y=err_y(N+1:2:end);

  w=[w_xx w_xy w_yx w_yy];

  end
```

A.7 CD_fil.m

```matlab
% dispersion compensation in time domain
% Chromatic dispersion is described as all−pass filter
% ref [Kuschnerov, "Adaptive chromatic dispersion equalization for
% non−dispersion managed coherent systems"] (OMT1.pdf)
5
    function [data w]=CD_fil(data, num_taps, dispersion, wavelength,
            sample_period, calc_tap, w)

    if calc_tap==1

10      C=3e5; % Speed of light (km/s)
        for q=1:num_taps
            arg=(j*C*sample_period^2)/(dispersion*wavelength^2);
            b(q)=sqrt(arg)*exp(-arg*pi*(q-round(num_taps/2))^2);
        end
15
        w=b.*kaiser(num_taps)';

    end

20  data=filter(w,1,data'); %chromatic dispersion equalization
    data=data.';

    end
```

A.8 phase_compl.m

% phase noise compensation using viterbi and viterbi

```matlab
function [ph_rec_data rot_angle]=phase_comp(data, window);


5  const_rot=-pi/4;


   for q=1:length(data)
        n1=q-window;
          if q-window<1
10            n1=1;
          end
        n2=q+window;
          if (q+window)>length(data)
            n2=length(data);
15        end
        rot_angle(q)=angle(sum(data(n1:n2).^4));


   end


20 rot_angle=[(last_angles*4) rot_angle];
   rot_angle=(unwrap(rot_angle))/4;


   for qq=1:length(data)
        ph_rec_data(qq)=data(qq)*exp(-j*rot_angle(qq))*exp(j*const_rot);
25 end
```

A.9 CD and NLC

```
Dispersion=LoopDispersion ;
    sample_period=1e12/(SymbolRate*NewSamplesPerSymbol ) ;
    fft_length = value ;

5   for span = 1: str2double (loop)

    %% Chromatic dispersion compensation using FFT filter %%

        if frequency_domain_CD

10
            [ data_IQx data_IQy]=CD_fil_fde ( data_IQx , data_IQy , fft_length ,
                Dispersion , RefWavelength*1e9 , sample_period ) ;

            %%Nonlinear compensator%%

15          if NLC

                alpha = value ;
                beta = value ;

20              data_IQx=data_IQx.* exp(−i *(alpha *(abs ( data_IQx )).^2+beta *(
                    abs ( data_IQy )).^2));
                data_IQy=data_IQy.* exp(−i *(alpha *(abs ( data_IQy )).^2+beta *(
                    abs ( data_IQx )).^2));

            end
        end
25  end
```