



UNIVERSIDADE DA BEIRA INTERIOR
Covilhã | Portugal

An Object-Oriented Architecture for Transmission Simulation of Image and Video with Multiple Description Coding Over High-Speed Optical Fibre Channels

Rui Pedro Quaresma Braz

Submitted to the University of Beira Interior in candidature for the degree of
Master of Science in Computer Science and Engineering

Supervised by Dr. Mário Marques Freire

Co-supervised by Dra. Maria Manuela Areias Costa Pereira de Sousa

Department of Computer Science
University of Beira Interior
Covilhã, Portugal
<http://www.di.ubi.pt>

This page was intentionally left blank

Acknowledgements

Completing a thesis is a challenge that is only possible with the support of a lot of people.

First of all, i would like to dedicate this thesis to my parents for their continuous support and strength given.

To my Professors, Mário Freire and Manuela Pereira, for the scientific support, the teaching skills and human character.

The unconditional support that i had from my friends, Gonçalo and all the people from Network and Multimedia Computing Group (NMCG) and RELiable And SEcure Computation Group (Release) for the cheering and good moments they provided during this long year.

To my family, specially to João Pedro and Joseph, for the time spent and suggestions.

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia through the project TRAMANET: Traffic and Trust Management in Peer-to - Peer Networks, with contracts PTDC/EIA/73072/2006 and FCOMP-01-0124-FEDER-007253.

Abstract

This dissertation addresses the problem of providing a tool with an object-oriented architecture in order to allow the possibility of simulate the transmission of images and/or videos coded with Multiple Description over high-speed optical fibre channels.

In order to solve the problem proposed the choice was a simulator, once the advantage of this solution when compared to a real experiment as several advantages.

The solution presented on this dissertation presents several advantages when compared with the solutions already analysed that only allows to analyse some physical phenomena that occur on optical fibre, not filling the requirements pretended to simulate. The proposed solution was developed using an object-oriented architecture, providing the possibility of increment the number of scenarios possible to simulate.

On this dissertation all of the development process is described, since the structure of the proposed solution, to the classes developed as long as the Graphical User Interface implemented. The simulation results presented were obtained using a Multiple Description Coding algorithm.

This page was intentionally left blank

Keywords

Multiple Description Coding, Multiple Description of Image and Video Coding, Simulation of Multichannel Optical Communication Systems, Object-Oriented Architecture for Transmission Simulation, Optical Fibre Communications

This page was intentionally left blank

Contents

Acknowledgements	iii
Abstract	v
Keywords	vii
Contents	ix
List of Figures	xiii
List of Tables	xv
Acronyms	xvii
1 Introduction	1
1.1 Focus and Scope	1
1.2 Problem Definition and Objectives	3
1.3 Organization of the Dissertation	4
1.4 Main Contributions	5
2 Multiple Description of Image and Video Coding	7
2.1 Introduction	7
2.2 Multiple Description Coding Details	9
2.2.1 Coding Based on Quantization	10
2.2.2 Coding Based on Transform	10
2.3 Multiple Description Coding Used for Transmission Simulation	11

2.4	Summary	12
3	Object-Oriented Architecture for Simulation of Multichannel Optical Communication Systems	13
3.1	Introduction	13
3.2	User Interface of the Simulator	15
3.2.1	Laser Driver	16
3.2.2	Laser	19
3.2.3	Multiplexer	20
3.2.4	Fibre	21
3.2.5	Demultiplexer	22
3.2.6	Electrical Filter	23
3.2.7	Signal Reconstruction	24
3.2.8	Result	25
3.3	Object Oriented Architecture of the Simulator	26
3.4	Implementation Details	30
3.4.1	Laser Driver	30
3.4.2	Laser	33
3.4.3	Multiplexer	35
3.4.4	Fibre	36
3.4.5	Demultiplexer	38
3.4.6	Electrical Filter	39
3.4.7	Signal Reconstruction	40
3.4.8	Result	40
3.5	Validation of the Simulator	41
3.6	Summary	42
4	Simulations Results and Discussion	45
4.1	Simulation Scenario	45
4.2	Simulation Results and Discussion	48

5	Final Conclusions and Future Work	51
5.1	Main Conclusions	51
5.2	Directions for Future Work	52
	References	53
A	Mathematical Functions Implemented	59
A.1	Laser Driver	59
A.2	Multiple Quantum Well Laser	60
A.3	Single-Mode Fibre	63
A.4	Demultiplexer	63
A.4.1	Fabri-Perot Single Cavity	63
A.4.2	Fabri-Perot Three Mirror	64
A.4.3	First Order Resistor-Capacitor Filter	65

This page was intentionally left blank

List of Figures

1.1	High-level diagram of the implemented solution.	3
2.1	Typical structure of a common Multiple Description Coding. Adapted from [1].	9
3.1	Simplified Class Diagram of the simulator.	15
3.2	Laser Driver Tab of User interface.	16
3.3	Error provided when file exceeds <i>Max Bytes</i> value.	18
3.4	Laser Tab of User interface.	19
3.5	Laser Result Window (input:1100101001011011).	20
3.6	Multiplexer Tab of User interface.	21
3.7	Fibre Tab of User Interface.	21
3.8	Demultiplexer Tab	22
3.9	Electrical filter tab.	23
3.10	Reconstruction Tab of User Interface with all options enabled.	24
3.11	Example of an Eye Diagram.	24
3.12	Result tab with the reconstructed signal and errors.	25
3.13	Simulator classes developed	27
3.14	Simulator classes developed (continuation)	28
3.15	Flowchart of the implemented simulator.	29
3.16	Optical communication scheme	29
3.17	Laser driver data examples	31
3.18	Input size control Flowchart	32

3.19	Plots of Laser modelling (input:1100101001011011).	34
3.20	Multiplexer example	35
3.21	Fibre attenuation loss.	36
3.22	Fibre dispersion phenomenon.	37
3.23	Fabri Perot Filter transfer functions.	38
3.24	Plot of Signal Filter example.	39
3.25	Eye diagram of a 10 Gigabit per second transmission simulation.	41
3.26	Eye diagram of a 20 Gigabit per second transmission simulation.	42
3.27	Eye diagram of a 40 Gigabit per second transmission simulation.	42
4.1	JPEG 2000 Video coder Scheme	46
4.2	File merge and split flowchart	47
4.3	Simulation results for transmission over 20, 30 and 40 km with 0.0, 0.2 and 1.0 redundancy, respectively.	49

List of Tables

3.1	Laser parameters	34
4.1	Simulation parameters to 20, 30 and 40 km transmission.	48
4.2	Simulation results	50

This page was intentionally left blank

Acronyms

API Application Programming Interface

DWDM Dense Wavelength Division Multiplexing

FFT Fast Fourier Transform

FSR Free Spectral Range

FWHM Full Width at Half Maximum

Gbps Gigabit per second

GOP Group of Pictures

GUI Graphical User Interface

HDTV High-Definition TeleVision

IFFT Inverse Fast Furrier Transform

IP Internet Protocol

IPTV Internet Protocol Television

ISI Inter-Symbol Interference

ITU-T Telecommunication Standardization Sector

MD Multiple Description

MDC Multiple Description Coding

MDTC Multiple Description Transform Coding

MQW Multiple Quantum Well

NMCG Network and Multimedia Computing Group

OO object-oriented

OOP object-oriented programming

P2P Peer to Peer

QoE Quality of Experience

RC Resistor-Capacitor

Release RELiable And SEcure Computation Group

SCH Separate Confinement Heterostructure

SMF Single-Mode Fibre

SNR Signal-to-Noise Ratio

TCP Transmission Control Protocol

UHDV Ultra High Definition Video

UI User Interface

UML Unified Modelling Language

VoIP Voice over IP

WT wavelet-transform

Chapter 1

Introduction

1.1 Focus and Scope

The transmission of information through light has been done since the ancient times. During those times, the signal bonfires were frequently used to warn other positions, in a visible range, of an attack or invasion. This form of communication was a fast way to spread a message, but the worst disadvantage was that there had to exist a line of sight between the two points of the communication. This was the main problem of this transmission method [2].

During the year of 1870, John Tyndall, a physicist made a public demonstration that light could follow a curved path [3]. Since then, scientists tried to use this possibility to transmit information. Alexander G. Bell¹ was one of the scientists that used this method to transmit information. In 1880 he transmitted sound over a beam of light with the "photophone". The device used solar light that was reflected on a vibrating mirror. The light on the receiver generate a varying resistance in the selenium cells. This device is represented and registered at the U.S. Patents No. 235,496 (Photophone-Transmitter) and 241,909 (Photophonic Receiver).

Scientists tried to develop a material that could guide light through a long distance. The step that allowed this technology to evolve was reported in a paper published by Charles H. Kao and George A. Hockham [4] in 1960s, where they claimed that thin glass fibre could be used to transmit information. The major problem at that time was the attenuation, which had to be less than 20 dB/Km. That value was

¹Alexander Graham Bell was the inventor of the first telephone patented on March 7, 1876

only archived in 1970 by Corning Glass Works [5]. The improvement of optical fibre manufacturing keep on improving, and the attenuation issue become minor, reaching nowadays values close to 1.67 dB/km [6].

Nowadays, optical fibre is widely used for communications, once it provides a large capacity for data transmission. At the present, the requirements of broadband are growing, and that is caused by the growing use of Internet Protocol (IP) based services, like Internet Protocol Television (IPTV), WEBTV, [7], Peer to Peer (P2P) TV, like LiveStation and TVU Player [8]. These services have a large requirement of data and some have an assured quality of service, like IPTV, leading to the development of solutions to provide a better service, using the least bandwidth.

The use of optical fibre technology is costly, making suitable the usage of simulation to investigate the suitability of this technology to support that kind of services. The simulation of real problems is a widely used solution with their own advantages and disadvantages. Nevertheless according to several studies [9, 10, 11, 12], among others, there are more advantages in simulation than disadvantages.

Simulation tries to mimic the reality on a controlled environment, such as a computer. According to [12], simulation never resemble the reality, once it is a simplification of reality. Some variables are dismissed to represent reality mathematically, and also to simplify the understanding of the events. On simulation, it is possible to simulate complex physical phenomena changing the configuration easily, avoiding the costs and the time of the production of real models, making possible more experiments in a shorter time [10]. Therefore, this dissertation explores the use of simulation in the research work in order to keep costs as low as possible.

Several video formats are being developed and refined, with very high resolutions, like WHUXGA (7680x4800) for computer devices or 4K (4096x6144 35mm VistaVision) as a Digital Cinema Formats. An example of this kind of formats is *4K New Sony Projector* that provides a resolution of 4096x2160 pixel frames, coded at 24 to 30 frames per second. This technology is already used on digital cinema, with the problem of the distribution and storage of the data. The JPEG2000 is used to compress the signal, decreasing the requirements of bandwidth from about 6.3 Gbps to 500 Mbps [13]. Even with the bandwidth around 500 Mbps the packet loss caused by the switching of the video stream is inevitable [14]. Higher video resolutions is the next step in video, once several new formats are being developed and tested, as the example of a live transmission of a TV show with a 22.2 sound signal, using 28 Mbps (uncompressed),

Ultra High Definition Video (UHDV) using 600 Mbps as explained in [15] or the new video format that is being tested by NHK Science & Technical Research Laboratories, with a resolution of 7680x4320 at 60 frames per second [16], using an optical fibre between London and Amsterdam with an IP packet service [17].

Considering the developments in this area, we can accept that video transmission over optical fibre is already common and will continue to be a common scenario, growing the bandwidth needs and the quality assurance in order to give the end-user a multimedia experience with the best Quality of Experience (QoE) [18]. Another motive for that requirement is the High-Definition Television (HDTV) share growing and the display prices decreasing [19].

The developed simulator has the purpose to allow the simulation of a multichannel transmission through optical fibre of image and video descriptors, with several settable parameters.

1.2 Problem Definition and Objectives

The problem addressed in this research programme was focused on the design and implementation of a simulator based on object-oriented (OO) architecture to mimic a multichannel optical fibre transmission of multiple descriptors of image and videos. Multiple Description Multiple Description (MD) is a coding technique that fragments data into several sub-streams, naming each one as descriptors. There are several parameters that have to be taken into account for the simulation of the optical transmission system. The model taken into consideration is a model that presents results very close to reality, according to the research work reported in [20] and [21]. The proposed solution is described in the high-level diagram shown in Figure 1.1.

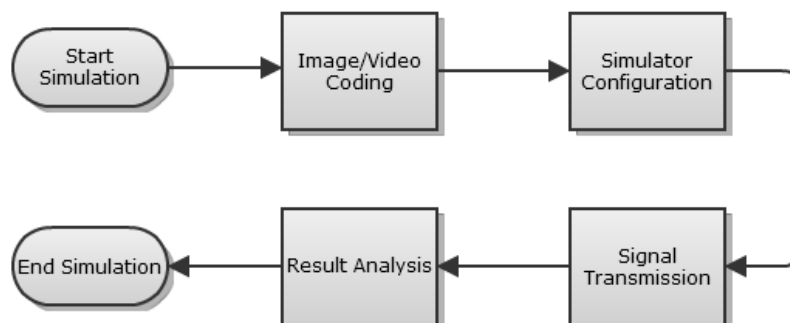


Figure 1.1: High-level diagram of the implemented solution.

The solution proposed in the dissertation intends to allow the user to simulate certain conditions very similar to the real world ones, giving the opportunity to test algorithms, like the one proposed in [22] with some improvements developed by members of the Network and Multimedia Computing Group (NMCG), namely the ones reported in [23] and [24], among others.

1.3 Organization of the Dissertation

This dissertation is arranged in five chapters. This first chapter describes the frame of the optical fibre communications as well as an historical background of the subject. It also provides a notion of the solution proposed to the simulation of MD images and videos using optical fibre.

On chapter 2 the basic concept of Multiple Description Coding (MDC) is explained in order to also understand the motivation of the use of this technology on the solution proposed. This chapter gives a theoretical introduction to this codification method with the purpose to provide the notions necessary to understand MDC. It also explain in more detail the codification method used on this dissertation.

On chapter 3 the simulator propose is described. As the solution was developed using an OO paradigm, in this chapter all of the classes and components developed are described in detail, as well as the architecture proposed, using class diagrams present this information. In a first part of the chapter the Graphical User Interface (GUI) is described and explained the restrictions defined in each field, if any. On a second part, the simulator architecture is described. On a third part the details of implementation are developed, describing the methods and formulas used in each class and component, including the explanation of some physical phenomena that are simulated.

On chapter 4 the results of simulations are displayed as well as the parameters used to obtain the results showed. The results presented are object of an analysis that is described also in this chapter. In this chapter the results are also presented with frames of the transmitted video, in order to provide a visual perception of the values presented.

On chapter 5 the final conclusions of the work done are presented, analysing the work done considering the results obtained, as well as proposal of future work as a continuation of the work developed. With the analysis of the work done the

conclusions taken are used to describe possibilities to use the simulator developed and the difficulties that had to be bypassed in order to complete the work developed.

1.4 Main Contributions

The contribution that this work provides as an advance in this area is to provide a tool that allows a wide variety of simulation scenarios providing the means to test an optical fibre communication and set the best parameters to the MD coding so the best quality can be obtained with the less redundancy.

This work also provides a simulation set up for optical fibre communications, once it provides a tool to validate algorithms designed for this kind of technology. The solution proposed allows a easily expansion and/or adaptation, once it was based on an OO structure, making possible to complete this solution to a specific problem or to add functionalities. The tool developed also allow to analyse and understand some physical phenomena that occurs on optical fibre signal transmission, such as attenuation and distortion. As several parameters might be configured the result of the system characteristics changes can be analysed, once the tool provides a step-by-step representation of the signal, in order to follow the resultant signal.

This page was intentionally left blank

Chapter 2

Multiple Description of Image and Video Coding

2.1 Introduction

The transmission of images and videos over networks is very frequent. An example of that are the websites, offering web-based video services such as *youtube* [7] or *tudou* [25], representing the largest part of it. This components, attending to the size, take more time to be downloaded by the user. The user experiment become more pleasant with progressive transmission. These approximation bases on the idea of transmitting the content, in this case images and/or videos, with different levels of detail. An initial image with a low detail level is transmitted, transmitting ordered information to detail the image on the consecutive packets as represented at [1]. This method assumes there is no packet loss and the sequence of the packets is important, needing all the due packets in order to complete the image at full detail level. An issue is raised because it is assumed that all packets are received and on the exact order, stalling the detail improvements until the packet needed is received. The system used by Transmission Control Protocol (TCP) to retransmit packets imply a delay, never shorter than the round trip time of the packet. This method presents advantages comparing to the non-progressive method, once an idea of the content can be shown before the complete transmission of the data, but a new problem arises, the packet lost that is impossible to be retransmitted or if retransmitted, with a time delay too high to be acceptable. This scenario become impracticable if we consider that packet loss is frequent during network transmission. This would result in problems during the

loading of multimedia components. This issue is worked out by the transport protocol used for packet transmission, the TCP, but even the lost packets are retransmitted, in some cases, the time of the retransmission, never smaller than the round trip time, is too long when considering specific applications such as real-time applications.

In [1] the authors present a representation of the same image using three different methods. The test was to send the image in ordered packets, delaying the third packet by a period of 20 seconds. In the non-progressive mode, the image is reconstructed as the packets arrive by order, but if one fails, the reconstruction stops until the packet is received. In the progressive mode the first packet has the entire picture, containing the consecutive details to improve the picture quality. If a packet fails or has a delay, the improvement stops, continuing when the missing packet is received. If a packet is lost and the system does not retransmit the packet, the previous modes do not have an alternative, leading to the Multiple Description Coding (MDC) mode that is explained in section 2.2. On MDC the image reconstruction only requires some packets to reconstruct the image with a lower quality, detailing the image when the packets arrive, regardless of the order.

The MDC coding was developed as a consequence of packet loss and delay, offering a great tolerance to packet loss and delay. The concept of separating the information was initially used by W. S. Boyle [1] to overcome such problems in phone lines, back in 1979. The system developed by Boyle had the purpose of improving the reliability of the calls, splitting the signal in two and sending each part in separate routes. In the case of failure of one of the links, the communication would continue with reduced quality. One of the first practical coder designs for Multiple Description (MD) was developed in 1981 by Jayant and Christensen [26] and [27]. They consider MD coding of differential pulse code modulation speech for combating speech coding degradation due to packet losses. If only even (odd) sample packets are lost, data contained in the odd (even) packet is used to estimate the missing samples using the nearest neighbour interpolation. Nowadays, the MDC process basis are the same, like information separation, but with a more complex process of information splitting.

The most common MDC method works with two channels (C_1, C_2) and three receivers (Side Decoder 1, Central Decoder and Side Decoder 2) as schematically represented in figure 2.1 [1]. To send the data, MDC codes the *Input* image into N descriptions, two in case of the figure 2.1, and sends them through the channels (C_N). When the descriptions are received, one of three scenarios may happen:

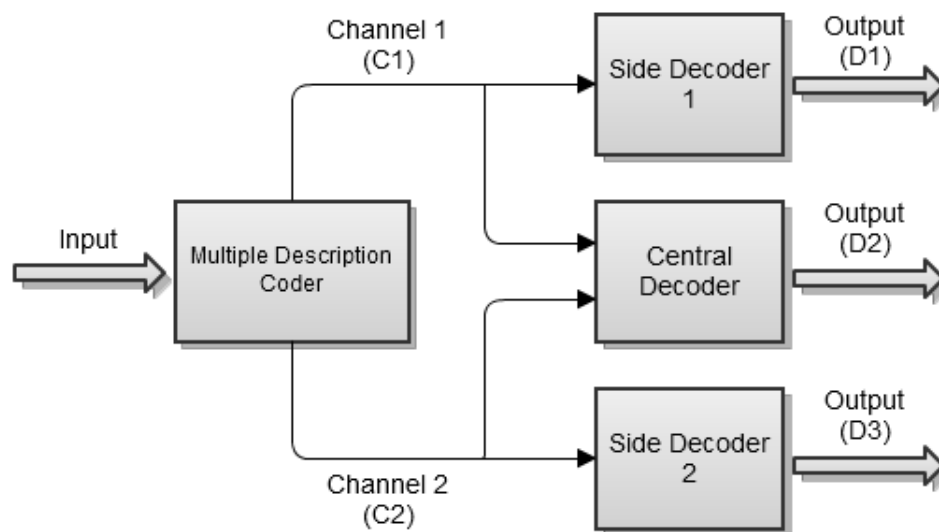


Figure 2.1: Typical structure of a common Multiple Description Coding. Adapted from [1].

- Both descriptions are received \Rightarrow Central decoder reconstructs the image with maximum quality;
- Only C_1 is received \Rightarrow Side Decoder 1 reconstructs the image, without the data of description 2, with lower quality;
- Only C_2 is received \Rightarrow Side Decoder 2 reconstructs the image, without the data of description 1, with lower quality.

The details of the model are explained in the following section, detailing the encoder structure.

2.2 Multiple Description Coding Details

MDC presents great advantages when compared to other methods of image and video transmission as analysed in [1]. Besides the advantages, it also brings some problems to the surface because of the encoder. The encoder of MDC is responsible of splitting the input in several channels. The method used to create the descriptions is intimately connected to the type of data coded and the channel characteristics. The influence of the transmission channels is done on the redundancy and data included in each description, *i.e.*, if the packet loss is high, each description needs more information of

the image, once the probability of receiving only part of the descriptions is high, on the contrary, if there is no packet loss, the need of redundancy is not useful. The type of data also influences the results of the MDC, despite that it can be used for other type of data transmission.

Considering the problems and barriers described, the coders usually have a few components like prediction, to analyse the transmission link so it can decide on the redundancy added and signal splitting according to the conditions and quantization, decorrelating transforms and entropy to input division. The prediction is used only when a dynamic codification is use, *i.e.*, when the transmission is analysed in real time and the descriptions coded according to that information. This issue is widely discussed in several articles, such as [24, 28, 29, 30], continuing the improvements on this area.

2.2.1 Coding Based on Quantization

The quantization is a complex part of the system. This component separates the signal into several N descriptions. The input data can be separated is several ways and the objective is, once the redundancy between descriptions is inevitable, to split data with the minimum of redundancy, but with the information necessary to reconstruct the input data. This also influences the values of the distortion and the the rate of each quantizer [22]. The objective is to separate the information in quantizer with some redundancy in order to be able to reconstruct the data with the minimal errors in case of one of the quantizer is not received. The main goal is to separate the information in order to minimize the redundancy and the distortion in case of a lost quantizer. The method commonly used is to divide the message in intervals, and represent each interval with a quantizer, being the quantizers correlated in order to make possible the predict , recurring to probability, the most value like to be in case of the lost of a descriptor. This method is explained in more detail at [23] and [1].

2.2.2 Coding Based on Transform

New MDC approaches exploit the natural correlation between symbols for reconstruction. The coding based in transform is similar to the square-transform based Multiple Description Transform Coding (MDTC) approach, except that the transform is

not actively designed. Some recently proposed methods, like in [22], have the possibility of chose the relative importance of the central and side decoders, and separating the input data according to the importance, making it possible to send more important data through a more reliable link and the less important through the other. This method gives a different distortion (quality of the approximation) for each channel, *i.e.*, the values of D_1 , D_2 and D_3 , as represented in figure 2.1, have different values. This imply that the values of the C_1 and C_2 bit rates are different. This method is usually applied when the channels have different characteristics. The steps used to code a MDTC are:

- Use a decorrelate transform T_1 (e.g. KLT, DCT, ...);
- Quantize the transformed coefficients;
- Transform the quantized vector with an invertible, discrete transform $T_2 : \mathbb{C}^n \rightarrow \mathbb{C}^m$ ($m = n$ i the square transform MDCT methodology and $m > n$ in the frame based MDTC methodology);
- Entropy code the resultant components;
- If the number of vector m is greater than the number of descriptors k , group them to be sent over the k channels.

2.3 Multiple Description Coding Used for Transmission Simulation

As referred before, MDC can be used in various types of data, as long as the possible loss of data might influence the system. As examples of usage of MDC, we can refer to speech, image and video. The great advantages of this codification is the packet loss tolerance, being useful for systems with a high packet loss. The other usage of MDC is the possibility of different quality (distortion) levels. This is specially useful for streaming video to several users that have different connection speeds, and different quality requirements, adjusting the descriptions sent to each user. There are several studies that analyse and propose different methodologies for the codification, according to the network type like [28, 31, 32]. Since we are only interested on the application of MDC for image/video transmission and due to it's complexity, MDC is not fully described in the dissertation. Therefore, consider reading the bibliography on this,

especially the references proposed, namely [22, 1, 28, 33]. Nevertheless the required details, needed for the understanding of the work reported in the dissertation, are provided below.

2.4 Summary

On this chapter an introduction to MDC is given. The MD coding have several approaches, being described on this chapter the one developed by [22, 23] and [30]. The basic about this concept is explained, as long as the codification based on the transform and on the quantization. The interest on this coding method is also target of analyses, once this method allow the transmission of video and/or images over noisy channels, with advantages when compared with the transmission of files coded without MD.

Chapter 3

Object-Oriented Architecture for Simulation of Multichannel Optical Communication Systems

3.1 Introduction

The usage of optical fibre has a short but busy story. The major problems of using optical communication until 1966 was the light source and optical fibres were in a stage of evolution that were impossible to rely on them to long distance communications, once other type of technology offered a more reliable and effective solution. In 1966 the technological barriers were abrogated by Kao and Hockham [4], that proposed the transmission of information using dielectric wave-guides or optical fibre. This was possible due to the the evolution on the attenuation of the fibre, that had values close to 1000 dB/Km at the beginning and was reduced to values close to 5 dB/Km. The other breakthrough was the development of laser as a light source that could be used to modulate the light and with more intensity light. The first lasers developed had a short lifetime, at most a few hours, but in a few years larger lifetimes were accomplished, around 1000 hours in 1973 and 7000 hours in 1977 [34].

After these technological advances, the possibility of communication based on this technology started to become a reality after 1980 [35]. Nowadays, optical fibre communications are widely deployed over the world. The number of connections are growing at fast speeds, according to the needs of users, that demand larger broadband

for several services, such as Internet Protocol Television (IPTV), WEBTV, Peer to Peer (P2P)TV, Voice over IP (VoIP) among other, that require higher bit rates. Other advantages of optical fibre are the basis of the fast grown of the usage of optical fibre communication. The huge bandwidth, the physics limit is around 12.5 Thz per fibre at 1550 nm window [36], the low transmission loss, the absence of electromagnetic interference, security, reliability and the costs make this technology the most suitable and advanced in the present, resulting in the fast proliferation of optical fibre.

Simulation is widely used to mimic experiments, leading to a reduction on costs and time of work. Knowing that, a solution to simulate single-mode optical fibres was developed. There are a few applications that allows the users to run some tests, but the need of one that was adjusted to our laboratory needs and to apply the results of the research work pursued at Network and Multimedia Computing Group (NMCG) lead to the development of this solution, once the simulation of file sending through optical fibre was not developed yet, only simulation of some of the physical phenomena of the fibre transmission as [11] and [37] or simulate the transmission of an array of bits [21].

Taking into account the scenario described above, we propose a simulator that allows the user to test several configurations, giving a feedback of the parameters used, showing messages of the values that would lead to a bad or impossible configuration. It also provides an optional step by step simulation providing the user a graphical result of the configuration proposed, allowing a more accurate and easy to understand analysis of the results.

There were several options and modules developed on the proposed solution. The main structure of the simulator Graphical User Interface (GUI) is divided in eight tabs, that are the Signal Parameters, Laser, Multiplexer and Demultiplexer, the Fibre, the Signal Filter, the Signal Reconstruction and the Result Tab. All this components are described in detail between section 3.2.1 and section 3.2.8. Some options were taken, according to the conditions and the tools available. Those options will be explained and justified during this chapter.

3.2 User Interface of the Simulator

The developed User Interface (UI) is divided in several tabs, each one representing a part of the system allowing the user to configure each component and getting the feedback of the configuration defined. Each tab represents a class that contains the data and pointers to other objects that it depends on. The connections of each class is represented on the class diagram shown in figure 3.1. This dependencies have a logical sequence, once the user needs to configure the components through a certain sequence according to the pretended objective and to the sequence of events that happen in real systems. A namespace with utility methods was also implemented, providing the user several functions, avoiding the implementation of similar methods in each class. The objects implemented and the utility methods will be described and explained in the next sections.

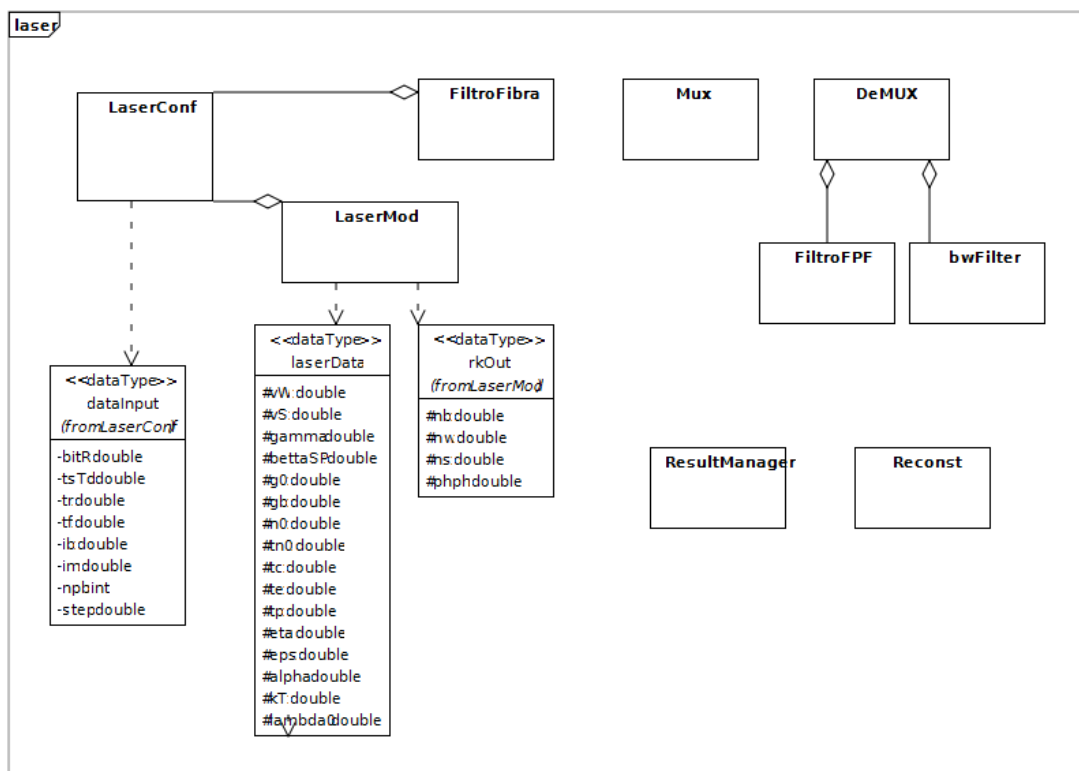


Figure 3.1: Simplified Class Diagram of the simulator.

The UI was developed according to the problem, creating a user interface for data configuration, and three more, one to show the results in a graphical mode, showing

the generated charts, another, similar to the Windows[©] Forms¹ MessageBox, once the needs for feedback messages were not fulfilled with the standard MessageBox and another Form that gives a feedback of the progress of the simulation when a file sending is simulated, with a progress bar and a message with the status of the progress. The steps (tabs) are developed in the next pages.

3.2.1 Laser Driver

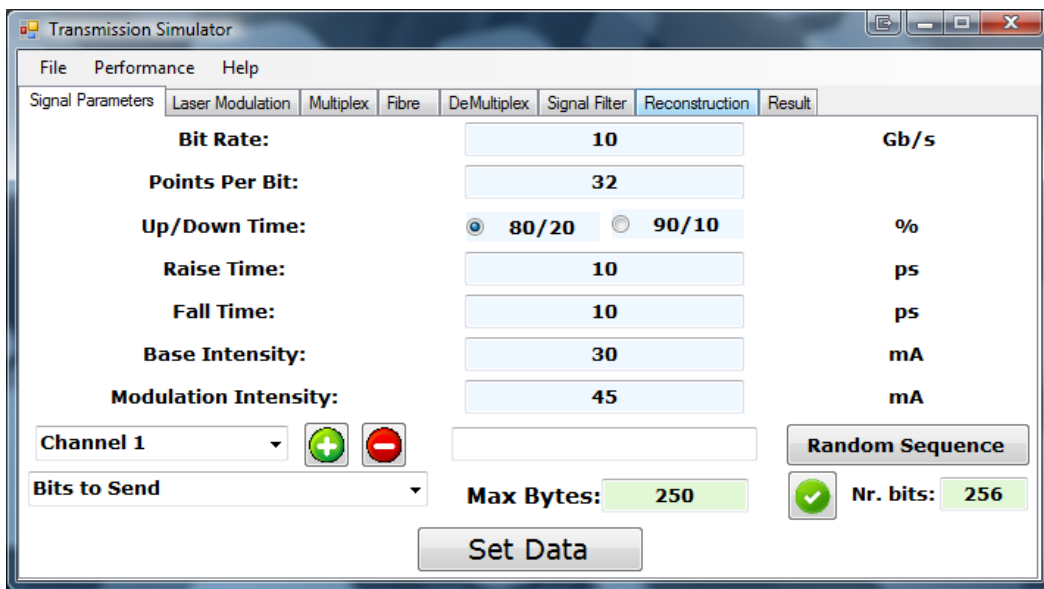


Figure 3.2: Laser Driver Tab of User interface.

The Laser Driver tab allows the user to configure some properties of the signal. On this configuration menu, the user have to provide the bit rate, the number of points per bit of the signal, the up/down time, the signal raise and fall time, the base intensity and the modulation intensity. On this menu, the user also defines the number of bits to transmit for each channel. The bit rate is the number of bits that are transmitted per second. This influences the time needed for data transmission, that can be computed with the formula:

$$T_{send} = \frac{1}{BitRate} * N_{bits} \tag{3.1}$$

¹Windows[©] Forms is a Graphical Application Programming Interface (API) to design UI

The number of points per bit, is the number of points used to numerically represent, within the bit time, the signal associated with a given bit. The higher this value is, the more detailed is the output signal. This value is restricted to values that can be represented as 2^n . Considering performance issues, this value is recommended to be less than or equal to 128 once values greater than this generate arrays too long, compromising the performance of the simulator. These two parameters, bit rate and number points per bit, are parameters that are not exclusive of the Laser Driver, once these values are parameters of the system and common to other elements of the simulator.

The fall and raise times of the electrical signal are the time (in percentage) of the signal transition from 1 to 0 or from 0 to 1, respectively. The user has two options, being possible the choice of the raise of the signal between 20 and 80% or between 10 and 90% of the value of the signal corresponding to a bit 1 in steady state (assuming positive logic). This option will shape the signal according to the formulas on the appendix A.3. The predefined values were set as 10 ps. This value is acceptable once it was tested on article [20], where the values were tested for a 60 Gigabit per second (Gbps) bit rate. This field has no restriction.

The Base Intensity and Modular Intensity are the values of the current injected. The base intensity is the lower intensity, and the modular is the intensity of the step, leading to a maximum current intensity injected on the laser of $I_{max} = I_{base} + I_{mod}$. The input values have several options and might have several channels. When adding channels, a verification of the maximum number of channels is made according to the formula 3.3. For each channel a different kind of input can be selected. The user can use the pseudo-random bit generator, that generates a sequence of bits with the size selected on the *number of bits* field. The user can also select a file with the bit sequence to send, or select a file to send. The field *Max Bytes* limits the size of the input. In case a file has more elements than the *Max Bytes* field, the message represented in figure 3.3 is presented, allowing the user to cancel, to send only a part of the file with the size of *Max Bytes*, or to send the all parts of the file. This last option gives the user the possibility of define all parameters of all the steps, and the simulator separates the file or files to send in several pieces, according to the *Max Bytes* value, and simulate the dispatch of each one, merging the result in a single file.

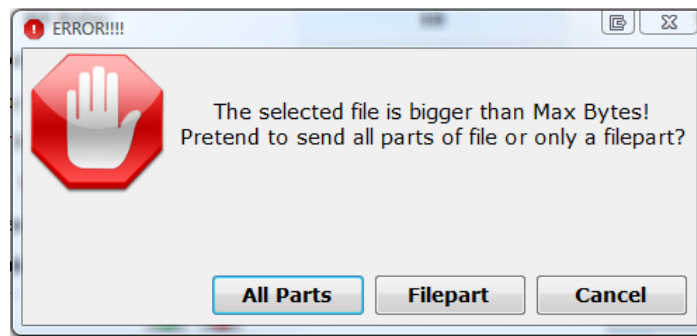


Figure 3.3: Error provided when file exceeds *Max Bytes* value.

The Signal Parameters Tab is the most restrained part, concerning to the data limits and restrictions. The restrictions on this option are at:

- Bit Rate
- Base Intensity and Modulation Intensity
- Max Bytes
- Number of bits
- Number of Channels

The bit rate parameter has a maximum value of 100 Gbps. This restriction was set based on the studies developed by [21], that was used as a reference during the simulator. The tested parameters were validated on that thesis, deciding not to allow higher values than 100 Gbps before deep tests. The Base and Modulation Intensity value has a restriction to avoid possible errors: $I_b > 0$ and $I_m > 0$.

The value of *Max Bytes* is restricted according to $MaxBytes \leq 1500$ Bytes. This restrictions make sense on performance matters, limiting the resources needed to allocate all data and the time to execute the simulations, and on logical matters, once this value was defined taking in concern the Transmission Control Protocol (TCP) packet maximum size, once the algorithm of Multiple Description Coding (MDC) use a packet base transmission method.

The Number of Bits is restricted to a value of 12000 bits (1500 Bytes) with the same reasons of the Max Bytes restriction. The value is 1500 bytes according to TCP maximum payload.

The Number of channels is validated according to the equation 3.3. The dividend returns the maximum frequency of the signal. To get the maximum number of channels that can be fitted on the signal, the maximum frequency is divided by the frequency deviation of each channel. According to the Telecommunication Standardization Sector (ITU-T) standard [38], the value of Dense Wavelength Division Multiplexing (DWDM) shift between each channel is 100 Ghz, simplifying the formula in 3.2, resulting in the formula presented in 3.3.

$$N_{channels} = \frac{BitRate * N_{ppb}}{\Delta\lambda} \quad (3.2)$$

$$N_{channels} = \frac{BitRate * N_{ppb}}{100 * 10^{(9)}} \quad (3.3)$$

3.2.2 Laser

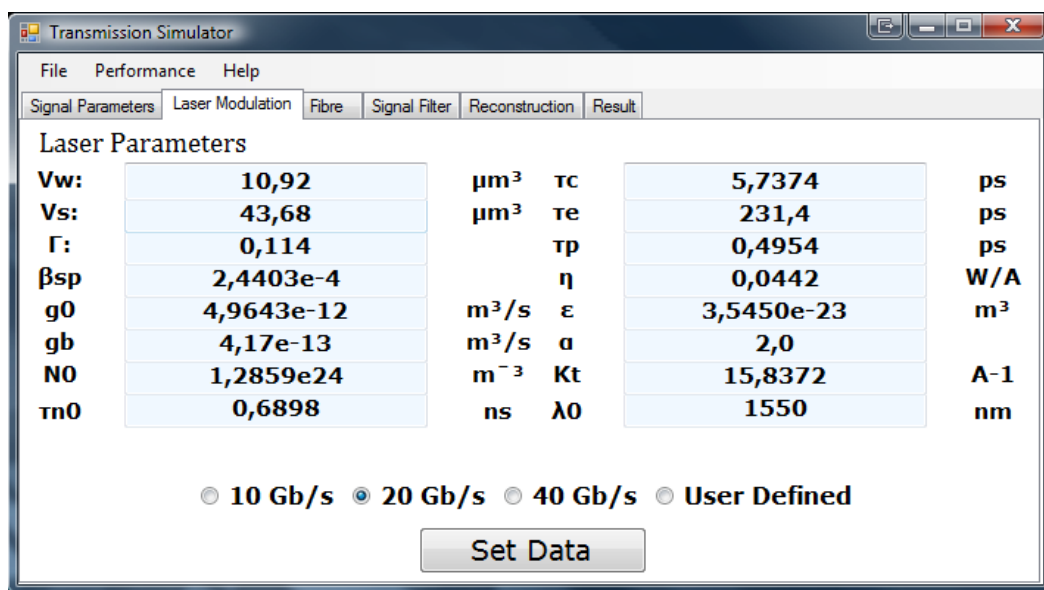


Figure 3.4: Laser Tab of User interface.

The Laser tab allow the user to input the values of the laser configuration, allowing the user to insert the model of the laser needed for the experiment. The configurable parameters are represented at appendix A in section A.2 and showed in figure 3.4. The user can choose a laser configuration based on parameters estimated on [20] and on chapter 3 of [21]. There are three previously defined different configurations according

to the maximum speed the laser can support. The laser parameters can be input by the user or a model parameters can be selected. The configured models that can be selected are the models estimated in [21] to a maximum bit rate of 10 Gbps, 20 Gbps or 40 Gbps, with a comparative graph in [21] that shows the similarity of the parameters with real fibre. The user can also define a model based on the parameters of the values already defined clicking on the 'User Defined' Radio Button, allowing the modification of the model values. The 'Set Data' button induce the execution of the simulation code of the laser, explained with more detail at section 3.4.2. As a result, if the option is selected, the result of the operation is shown in a plot, using the *ImageDisplay* class, giving the user three plots, one with the result of the equation presented in the appendix A in section A.17 (modulus and phase in separated plots), the result of the Fast Fourier Transform (FFT) (the real part and the imaginary in separate), and the result of the Inverse Fast Furrier Transform (IFFT), having this last one a purpose of user analyses, once it have no use on the simulation.

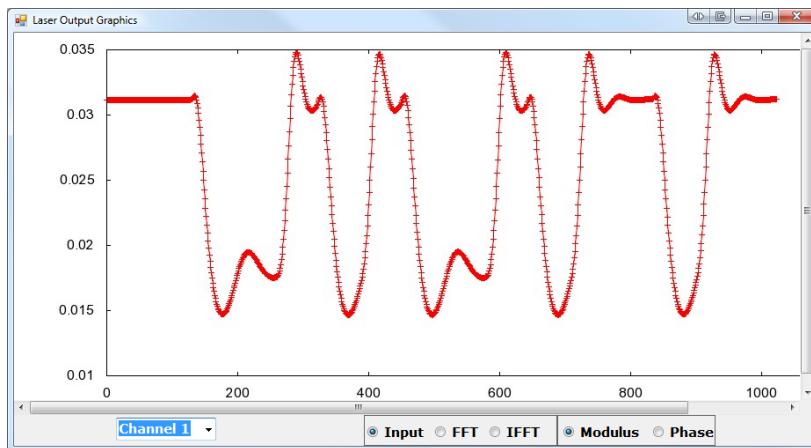


Figure 3.5: Laser Result Window (input:1100101001011011).

An example of the result presentation of this module is represented in figure 3.5.

3.2.3 Multiplexer

The multiplexer tab, as figure 3.6 shows, has only one field. On this version of the simulator the modification of this value is not possible as a result of the value specified is a ITU-T standard [38]. The execution of the multiplexer is made without the perception of the user of the data manipulation, returning as a feedback to the

user, if the option selected, plots of the result of the operation, similar to figure 3.20. The implementation details of this feature are developed on section 3.4.5.

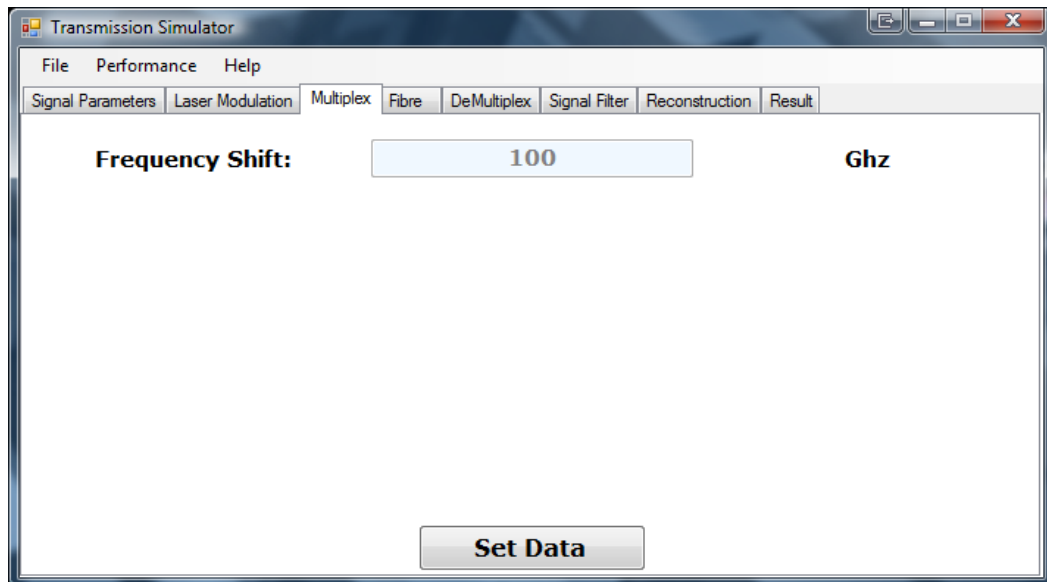


Figure 3.6: Multiplexer Tab of User interface.

3.2.4 Fibre

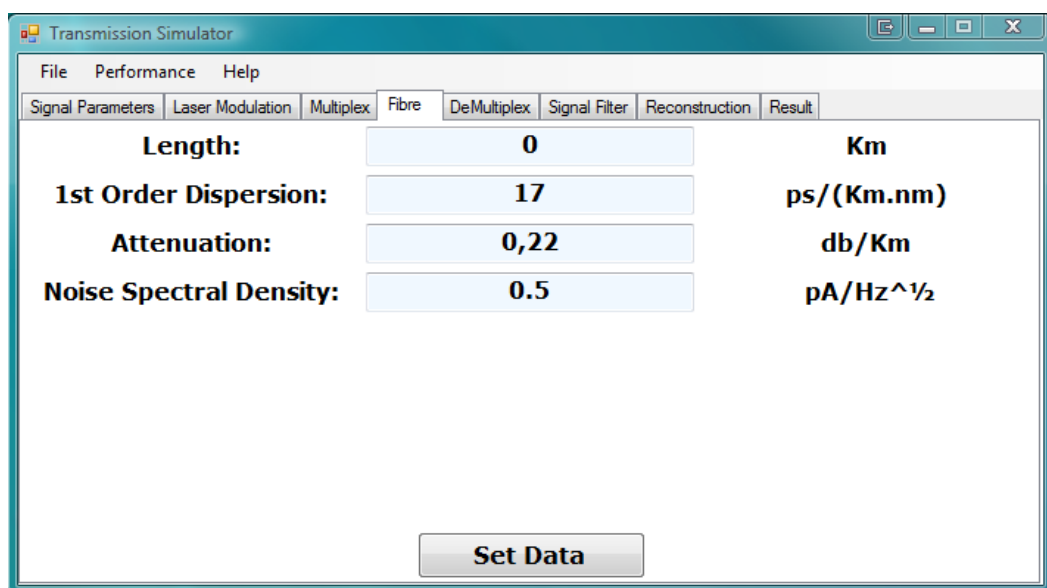


Figure 3.7: Fibre Tab of User Interface.

The Fibre tab is where the user input the parameters to the fibre modelling, according to the formulas in the appendix A in section A.3. The UI of this module is represented in figure 3.7. On this module, the user has to set the fibre length (in Km), the first (1st) Order Dispersion. The dispersion is explained in more detail at section 3.4.4. The attenuation is another physic phenomenon that represents the reduction of the transmitted energy over the light carrying medium. This phenomenon is developed in section 3.4.4.

3.2.5 Demultiplexer

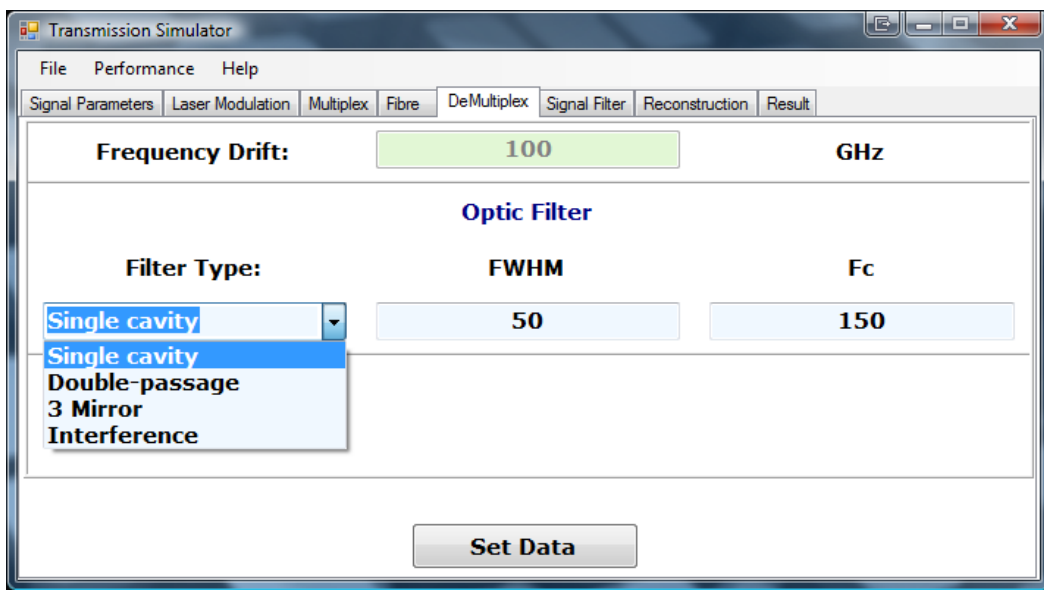


Figure 3.8: Demultiplexer Tab of User Interface.

The Demultiplexer tab, as represented in figure 3.8, requires more parameters than the multiplexer, as a result of the elimination of part of the interference caused by the transmission of the signal and the filter requirements resulting in a more reliable and usable signal. The structure of this component is described in section 3.4.5. In result of the requirements to demultiplex the signal the user has to input the parameters of an optic filter. The optic filter can be chosen between a single cavity, double-passage, three (3) mirror and interference. For each one the value of Full Width at Half Maximum (FWHM), that is a parameter frequently used to characterize the width on a curve, that gives the distance between the points of the curve at which the curve reaches half of its maximum value. The F_c parameter is the filter fineness that is

used to compute the Free Spectral Range (FSR) described in the formula A.21 given in appendix A. The formulas and the implementation details are described in section 3.4.5. The restriction of this tab are the range of values of each box. The *Frequency Drift* field is just an information field, once it was previously defined at the multiplexer tab, assuming the same value. The other fields have restrictions on the value, allowing only *values* > 0 and with 4 decimals.

3.2.6 Electrical Filter

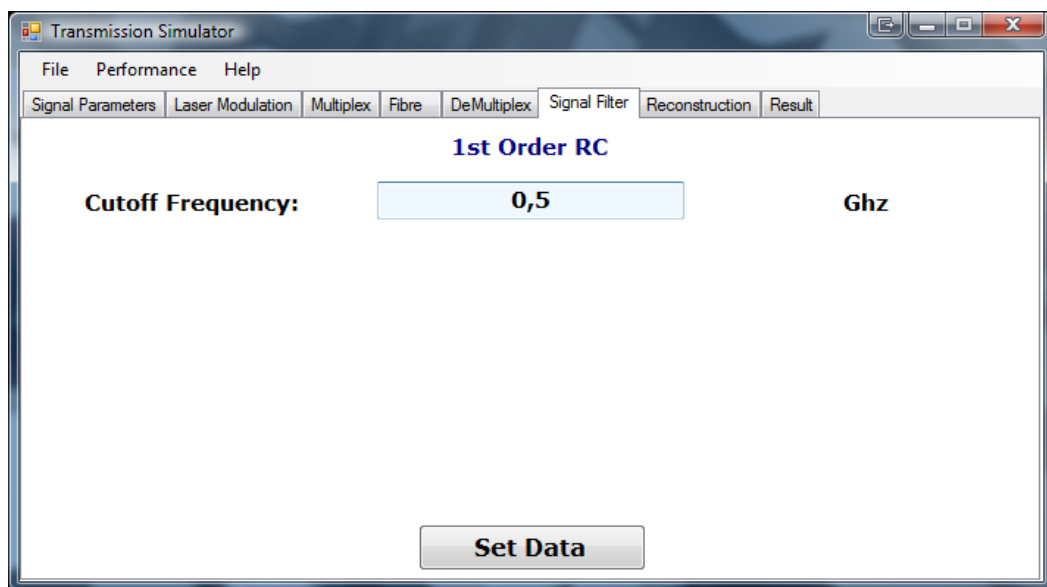


Figure 3.9: Electrical filter tab.

The electrical filter tab filters the signal resulting from the Demultiplexer tab or from the Fibre tab, if only one channel is defined. The operation done on this step is the application of an electronic filter, and Resistor-Capacitor (RC) filter. The purpose of the filter is to remove the interference caused by the multiplexing of the signal, and aliasing the signal removing the very high frequencies noise, smoothing the signal. The parameter necessary to the execution of the filter is the cut-off frequency of the filter. The default value of the filled is the value inserted in the Bit Rate inserted on the Signal Parameters tab, being possible changed by the user to allow a more narrow or wide filtering, according to the user requirements. The implementation details of this step, as well as an explanation of the RC filter is developed in section

3.4.6.

3.2.7 Signal Reconstruction

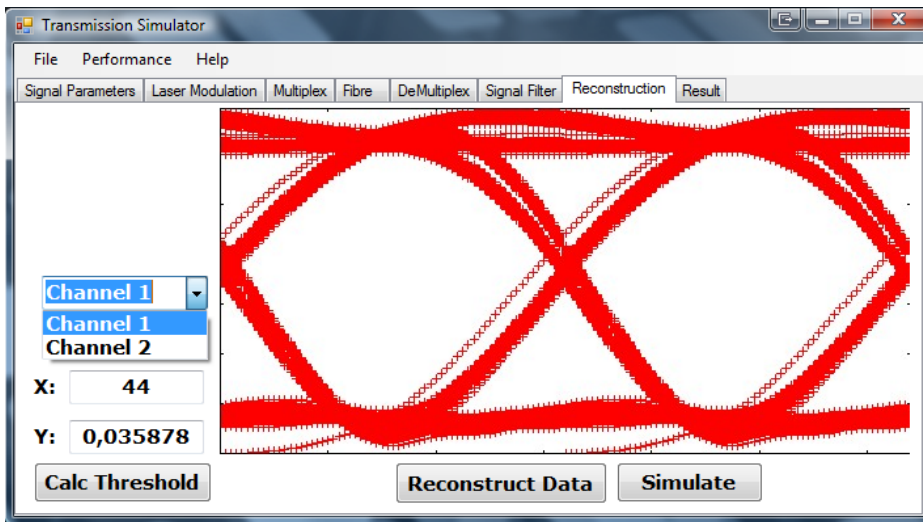


Figure 3.10: Reconstruction Tab of User Interface with all options enabled.

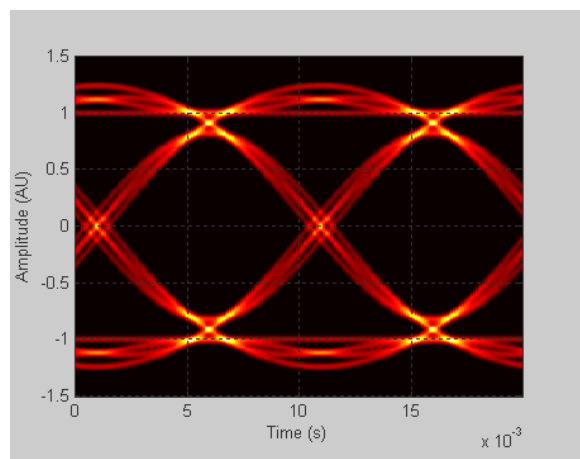


Figure 3.11: Example of an Eye Diagram.

The reconstruction tab, as showed in figure 3.10, allows the user to reconstruct the signal, setting the values of the threshold. During this step the signal from the Demultiplexer (or the Fibre, depending on the number of channels)is analysed, and using the trheshol computed based on the eye diagram reconstructs the signal. The reconstruction of the signal is done considering the value defined by the user or computed and generates the signal, as explained in section 3.4.7. This tab can be shown to the user in two different ways, according to the type of simulation in execution. In case of the scenario is an array of bits transmission, or a file that have no need to

be splitted, the GUI presented is similar to the one presented in figure 3.10, without the button *Simulate*. If the input is a split file, only the *Simulate* button is showed, disabling all the other options, the channel selection and the x and y field. If the file to send is not split or a bit array is sent, the user have to define the threshold on this step clicking on the eye diagram or with the button *Calc Threshold*, that computes the value using the method described in section 3.4.7, to all channels defined or an error is shown. If the simulation is continuous, *i.e.*, the transmission of several file parts, this step is computed with the same method, avoiding the user to intervene in the simulation.

The Eye Diagram is a useful way to analyse the quality of a signal, providing informations of the system performance, giving also information of the signal-to-noise, clock timing, Inter-Symbol Interference (ISI) among other information. The diagram is a oscilloscope display of the digital signal, with the repeated representation of the signal, as represented in figure 3.11. The eye diagram, in a purely digital system, gives the necessary information to test the integrity of the transmission system. The information that is possible to extract from this tool is very useful, since the ones described above, as long as information in order to compute the optimal sampling value to reconstruct the signal. More details of this subject can be found in [39] and [40].

3.2.8 Result

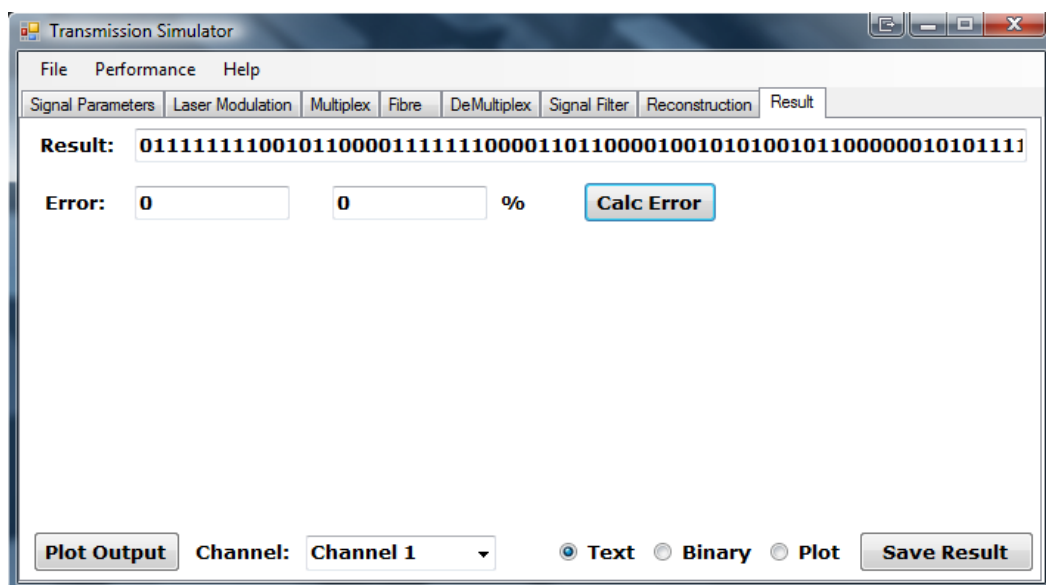


Figure 3.12: Result tab with the reconstructed signal and errors.

The Result tab presents the result of the simulation to the user. This step is only available in the case of a simulation of a bit array or a file part since the results of a continuous simulation is saved in a file as well as a file with the error information of each part. This part of the GUI, as represented in figure 3.12, gives the user to analyse the result of the simulation, as long as the possibility of the number of errors of the transmission, in absolute value and in percentage (%). The file containing the information of the simulation have the information:

- Number of bits transmitted

- Number of bit errors

- Number of burst errors

- Errors in percentage (%) of the resultant signal (comparing with the input)

The simulator also allows the user to save the result in a file, text or binary, or simply plot the data in a window or save the plot.

3.3 Object Oriented Architecture of the Simulator

The object-oriented programming (OOP) paradigm is widely used nowadays. This paradigm had a great disclosure, once it provides an approximation, when comparing to the procedural programming, to the real world, once in the real world the system is composed by objects. This paradigm also provides the developers an easier and faster way of code re-use, since each class models a part of the problem and is encapsulated, becoming easy the reuse of the code, and the expansion of the same, once it is not in the middle of a program. Besides those advantages, the code resulted is more structured once the functions of each object are well defined. This paradigm also has it downsides, the code generated using OOP is not so fast than the result of a procedural based program.

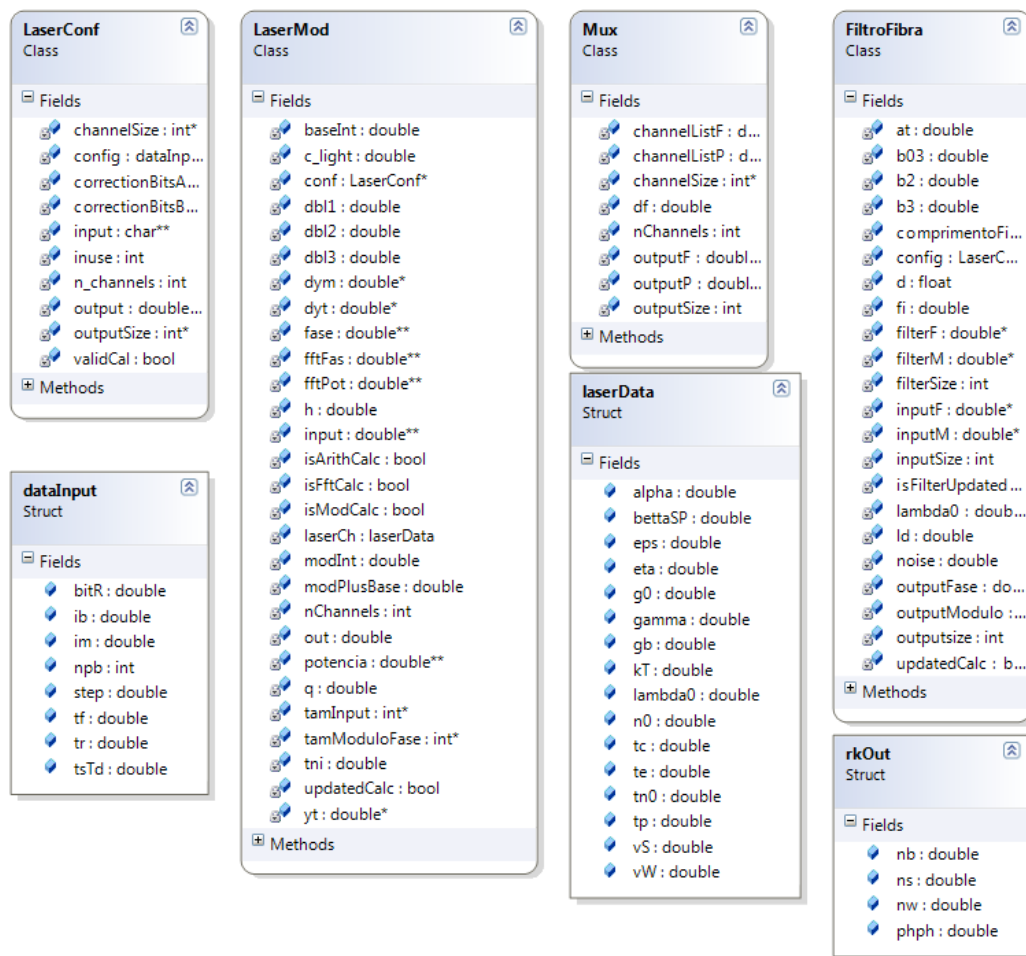


Figure 3.13: Scheme of the developed Simulator classes.

When comparing the downsides and upsides, OOP is a good solution for development once the hardware used today are more than enough for the general applications developed, and the development time required is less, justified by the code reusing and the available library's with developed classes. In a OOP paradigm, the objects are created by classes trading messages between them to get, set or trade some information or results.

For OOP projects modelling is very important, once, usually the projects developed with OOP are complex projects, with several classes, each one with even more attributes and methods, creating a complex development task and leading to results that cannot be the expected and/or the coordination and the messages traded between the objects malformed. To simplify this task, the Unified Modelling Language (UML) surges. The UML, as a language to model the application to develop, presents a standard

to characterize an application. Modelling the pretended solution, it avoids coding errors, and simplifies the task of the development of the solution by a large team, once the specifications are defined. Concerning to the UML, there are several types of diagrams like the Class, Package, Use Case, Sequence among others, being those the most commonly used. The class diagrams describes the static structure of a system. With this diagrams the structure of the application is defined, representing the several classes, the connection/dependencies between them, and the data structures defined to solve the problem.

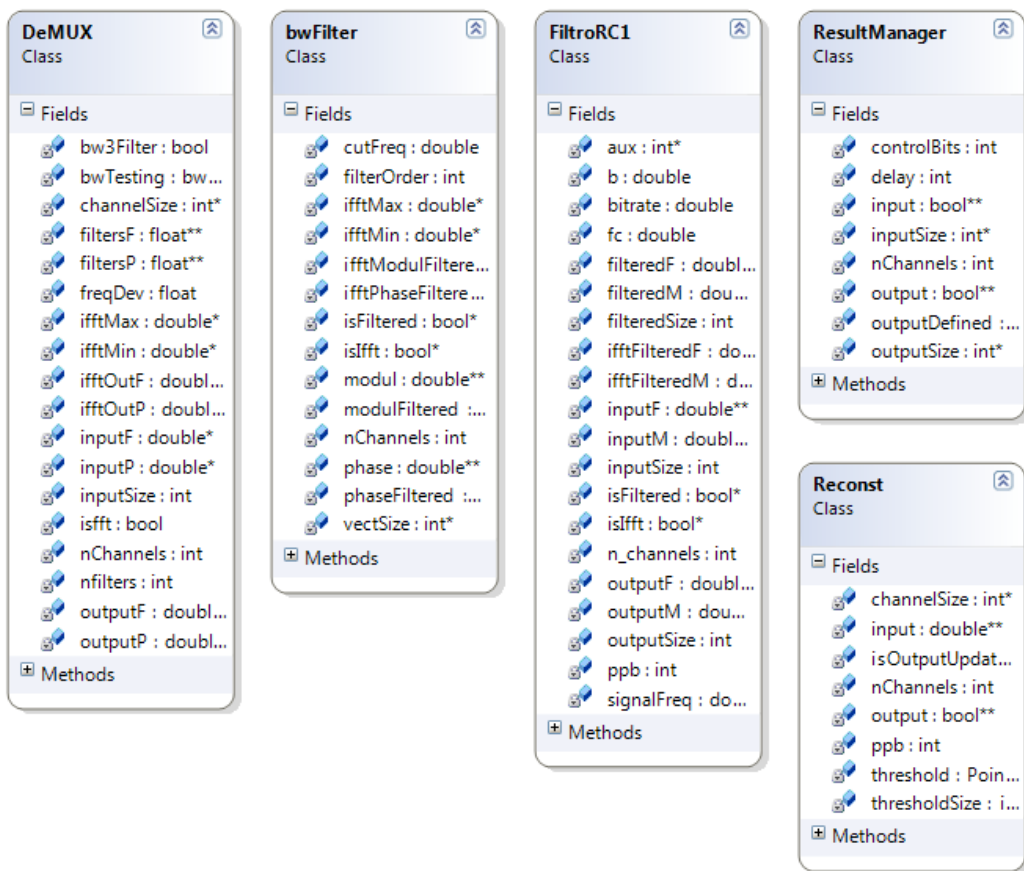


Figure 3.14: Scheme of the developed Simulator classes (continuation).

In figure 3.1 a simplified representation of the class diagram is showed. On this representation the data structures are defined, and the relation of each structure with the classes. The *Utils* namespace and the methods implemented on it are also represented in figures 3.13 - 3.14. This subject is developed with more detail in [41].

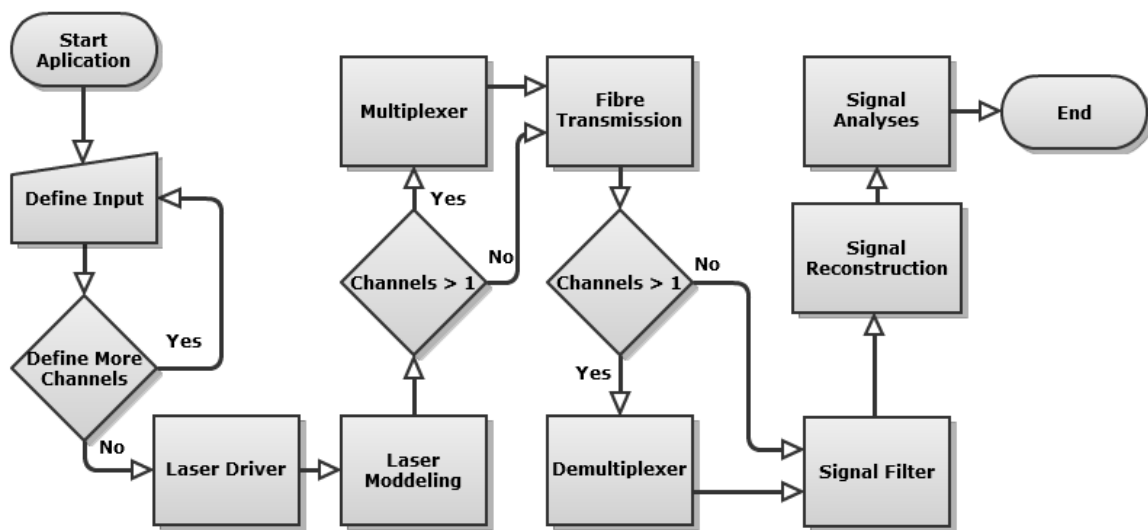


Figure 3.15: Flowchart of the implemented simulator.

The structure was designed as represented in figure 3.13 - 3.14 and defined after analysing the problem proposed to be solved. Once in real-world optic transmission has a structure/components as the represented in figure 3.16, the architecture of the simulator was defined as represented in figures 3.13 - 3.14. The system implemented consists in nine classes, three data structures, and a namespace of utility functions. The GUI was implemented using Visual Studio[©] 2008 and Matlab. Visual Studio[©] uses a *wusiwug WhatUSeelsWhatUGet* method, generating the code of the UI. The classes and objects are instantiated in the main file, controlling the messages between the object and validating results. The Matlab software was used in the signal filtering, as explained at section 3.4.6.

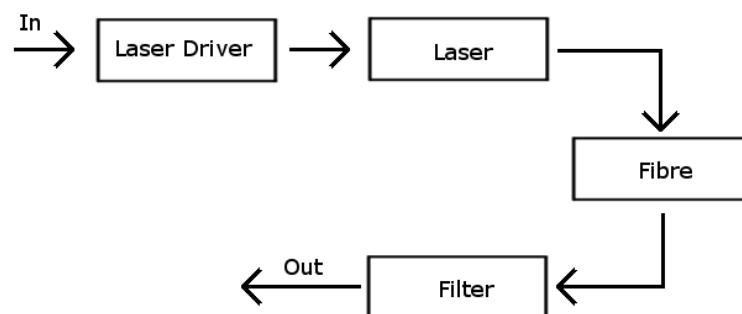


Figure 3.16: Simplified scheme of the Optical communication system.

3.4 Implementation Details

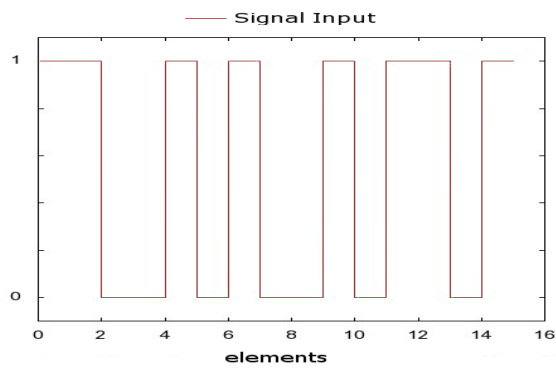
The simulator was developed using the object-oriented (OO) paradigm. Several modules were developed, each one representing a component of the optical communication system. A simplified class diagram of the modules developed is shown in figure 3.13 and in figure 3.14. A simplified diagram is shown in figure 3.1.

Each module is described in this chapter, justifying the choices made for each one. The Multiple Quantum Well (MQW) laser may be modelled by a set of differential equations, equations A.4-A.7, which are provided in [21]. This model was chosen once it represents a close representation of reality, as represented in chapter 3 of [21].

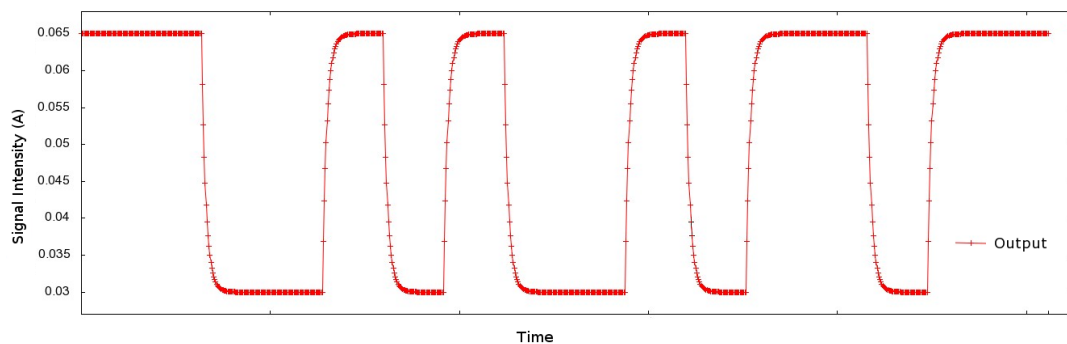
The implementation of the interface was done using the Windows[®] Forms, and the objects were implemented using C++. The usage of C++ was a requirement of the project, once this language brings advantage on the high runtime efficiency as verified in [42]. The choice of Windows[®] Forms was made in consideration of the used language for codification (C++) and the integration of the language with the UI, providing an easy integration of both. The UI was developed using Microsoft[®] Visual Studio[®] 2008, once this is a powerful tool do develop applications on the used language. During the development other choices where made with the purpose of improve the user experience and the application reliability. The implementation of the formulas and filters was a complex task due to the application and system requirements. This implementations are explained during this section.

3.4.1 Laser Driver

The laser driver is a component where the current intensity is modulated to send to the laser. The ideal scenario would be the one represented in figure 3.17(a), but in real world, the signal (in time domain) has a certain time to change state, as represented in figure 3.17(b).



(a) Laser driver input (binary)



(b) Laser driver output

Figure 3.17: Input and Output data examples of the laser driver (input:1100101001011011).

The signal parameters tab of the UI led to the implementation of utility methods, like the pseudo-random bit generator. This method was implemented applying the *rand()* method of the *stdlib* library. This method is used to generate bit sequences to be sent by one channel when the option 'Bits to Send' is selected. A feature of this component is the control of the size of the input signals, whether sequences or file parts. The size is controlled according to the first channel defined and/or by the *Max Bytes* value. In case the first channel is larger than the maximum value defined, a truncation or a continuous simulation is mandatory. When defining others channels, the size is controlled by the size of the first channel. If the channel to define is larger than the first one, the user can truncate the input of the channel or to extend the other channels already defined with a pseudo-random sequence that is ignored in the reconstruction feature. This feature is described in figure 3.18.

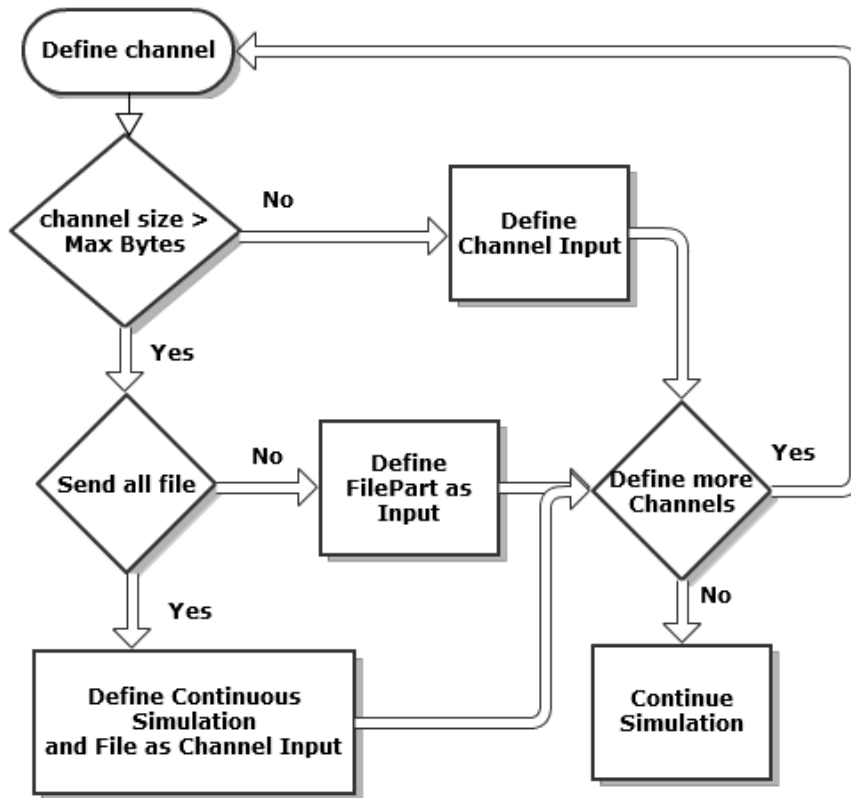


Figure 3.18: Input size control Flowchart.

The user also have the option of select a certain part of the file, considering the max number of bits and the part the user wants to select. This is done using a auxiliary method that opens the file, shift the number of bytes necessary to start reading the data on the pretended position. When a file is transmitted, the reading is done byte-by-byte, decomposing each one in bits, shifting the byte, getting the bit pretended. The solution used to set each channel was to set and validate each one step by step while it is defined, once this prevents the agglomeration of errors, becoming more user friendly the task of creating an error free configuration. The input is validated according to the size using: $N_{elements} = 2^n, n \in \mathbb{N}$, once FFT has a better performance with arrays that meet this requirement. In the continuous simulation, the *max Size* is a unrestricted value, leading to a array with a size that does not fulfil the 2^n requirement most of the times. The solution to this issue was to fill in the array until the size fulfil the requirements. The fill is done adding half of the bits at the beginning and the rest at the end. The number of elements added is kept in order to cut them out in the signal reconstruction.

When the button 'Set Data' is pressed, the procedure that simulates the laser command circuit starts with fields validation, according to the restrictions explained at section 3.2.1, and set the objects with the parameters defined, generating, if the option is selected, the plots using files with the data to plot. The tool used to create the plots is Gnuplot [43]. The simulation of the command circuit is done based on the formulas A.1 - A.3 developed on appendix A.

3.4.2 Laser

The laser modelling models the behaviour of the MQW. The equations of the model described between section A.4 and A.7, were develop by Nagarajan *et al.* on [44] and by Ribeiro *et al.*. The formulas are used to get the derivative, applying the fourth (4th) order Runge-Kutta method, obtaining the results as a complex number in polar form, as a result of function described on section A.17. In detail, when the user click the 'Set Data' the set and validation of parameters is performed, *i.e.* the conversion of the *Strings* inserted into numbers, returning an error message in case of error. The *LaserMod* object instanced is set with the inserted values, or with a model selected with the values of table 3.1. When the parameters are defined and the object fields set, the laser modelling starts, applying the formula, applying the fourth (4th) order Runge-Kutta. This method was implemented using a base of the Numerical Recipes [45] method. Once the modelling is complete and the array of the $\sqrt{P(t)}$ (modulus), and $\varphi(t)$ (phase), the FFT is performed, performing a transition from time domain to the frequency domain. To note that the FFT implemented performs the transform using arithmetic coordinates, forcing to convert the coordinates, using the formulas 3.4-3.5. The parameters of the models provided to 10, 20 and 40 Gbps are presented in table 3.1. The models used were proposed by [21].

$$P(t) = |Z| = \sqrt{x^2 + y^2}, \quad (3.4)$$

$$\varphi = \arg(Z) = \pm \arctan \frac{y}{x}, \quad (3.5)$$

where Z is the Cartesian representation ($Z = x + iy$).

The result of the laser modelling is two arrays, one with the result of the equations described in section A.17, labelled as input once is the input of the FFT, and a second array with the result of the FFT as represented in figure 3.19(a) and 3.19(b).

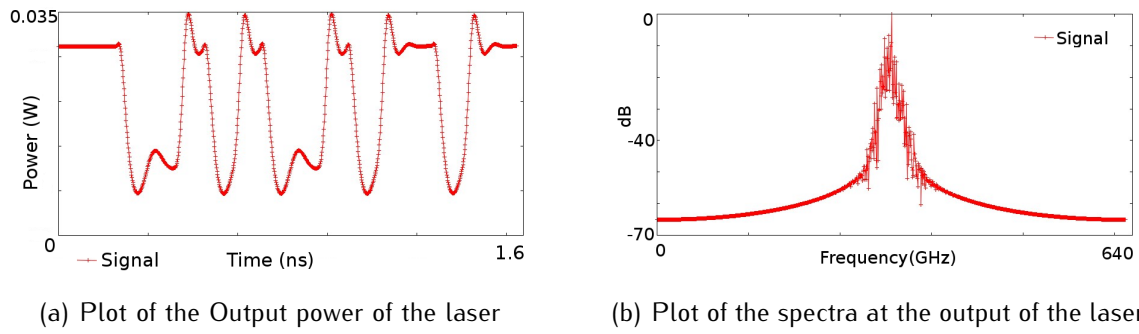


Figure 3.19: Plots of Laser modelling (input:1100101001011011).

	10 Gb/s	20 Gb/s	40 Gb/s	
V_W (Quantum Well Volume)	18	10.92	23.425	μm^3
V_S (SCH Volume)	72	43.68	93.7	μm^3
Γ (Optical Confinement Factor)	0.093	0.114	0.18715	
β_{sp} (Spontaneous Emission Factor)	10^{-4}	$2.4403 * 10^{-4}$	$9.8067 * 10^{-4}$	
g_0 (Linear Coefficient of Optical Gain)	$4 * 10^{-12}$	$4.9643 * 10^{-12}$	$9.22 * 10^{-12}$	m^3/s
g_b (SCH Coefficient)	$4.17 * 10^{-13}$	$4.17 * 10^{-13}$	$4.17 * 10^{-13}$	m^3/s
N_0 (Transparency Density)	$1 * 10^{24}$	$1.2859 * 10^{24}$	$1.12348 * 10^{24}$	m^{-3}
τ_{n0} (Bimolecular Recombination Lifetime)	0.718	0.6898	0.569935	ns
τ_c (Transport Time through SCH)	56.8	5.7374	7.4334	ps
τ_e (Thermionic Emission Time)	225	231.4	48.238779	ps
τ_p (Photon LifeTime)	3.95	0.4954	0.925391	ps
η (Face Quantic Differential Efficiency)	0.0442	0.0442	0.11	W/A
\mathcal{E} (Gain Compression Factor)	$2.33 * 10^{-23}$	$3.5450 * 10^{-23}$	$1.808 * 10^{-23}$	m^3
α (Spectral Dilatation Factor)	3.22	2	2	
K_t (Thermal Constant)	15.9	15.8372	10.468154	A^{-1}
λ_0 (Emission Wavelength)	1532	1550	1575	nm

Table 3.1: Laser parameters for simulation at 10, 20 and 40 Gbps

3.4.3 Multiplexer

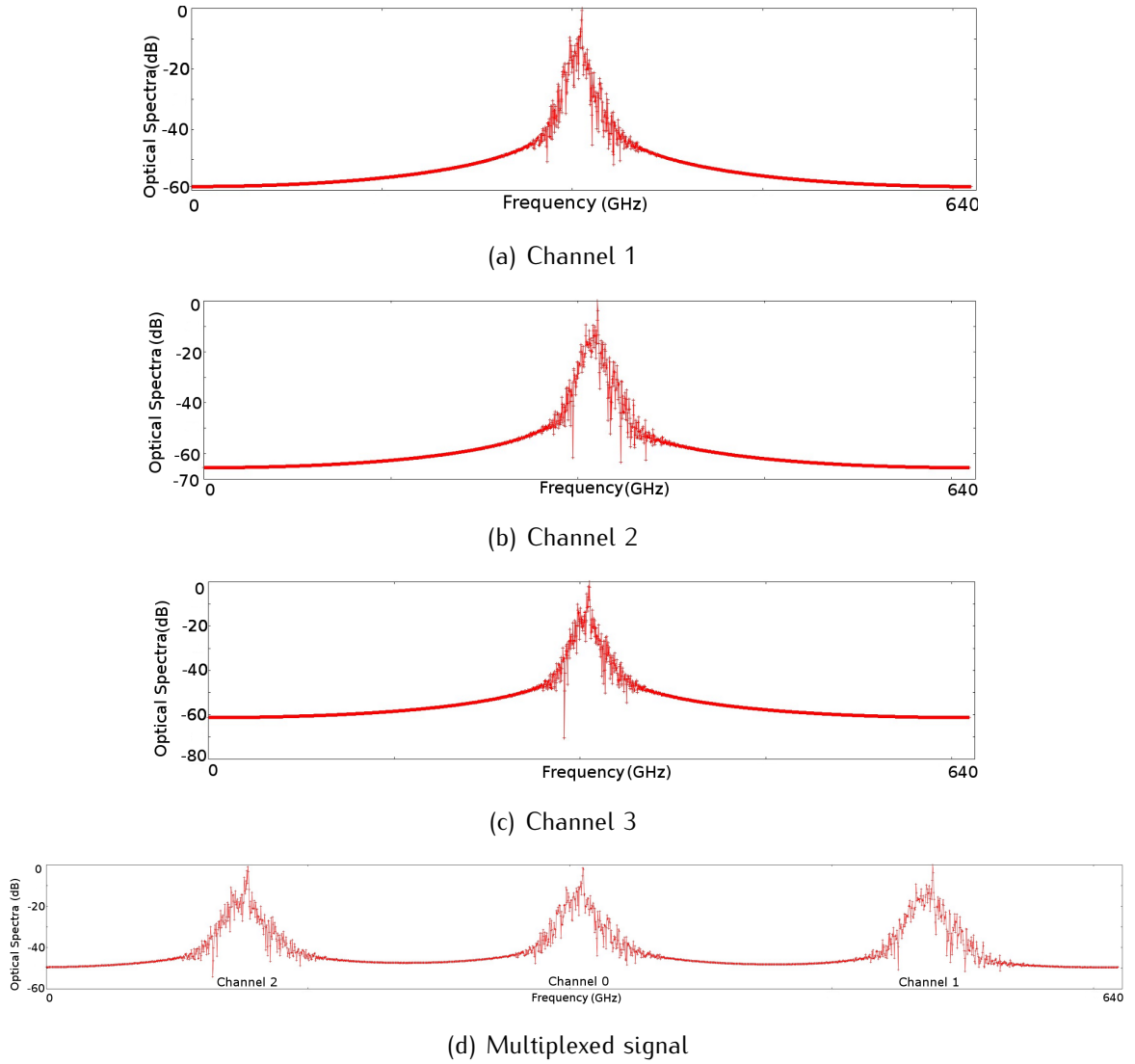


Figure 3.20: Spectra at the output of the Multiplexer with three (3) channels.

The multiplexer is a component that merges several inputs/channels into a single fibre, using different wavelengths. This operation brings a bandwidth expansion, for example, if 8 channels at 10 Gbps are merged in the wavelength dimension, a final net bandwidth of 80 Gbps is obtained. The optical multiplexer simulation is done according to the scheme showed in figure 3.20. The formula 3.6 describes the implementation of this module:

$$\sigma = \begin{cases} (n/2) * N_p, & n = 2k \quad k \in \mathbb{N} \\ -((n+1)/2) * N_p, & n = 2k+1 \quad k \in \mathbb{N} \end{cases} \quad (3.6)$$

where σ represents the number of elements to shift the signal, N_p represents the number of elements to shift on the array. The number of elements is calculated using $N_p = \frac{f_{max}}{N_e}$, where N_e is the number of elements of the array and f_{max} is the maximum frequency of the signal, that is obtained using $f_{max} = \frac{F_S}{2}$, where F_S represents the sample frequency. This formulas are explained at [46].

3.4.4 Fibre

The fibre is the component that, in real world, is the mean used to transmit the optical signal with low losses. The fibre affects the signal with physical phenomenon, as an example of the attenuation and dispersion. There are other some phenomenon that can be ignore, once the effects of them are not significant, and ignoring them make possible the modelling of the Single-Mode Fibre (SMF). The validation of a model ignoring this phenomenon is develop by [21] on chapter 3.4. Considering this, the model implemented ignores the non-linear effects and the birefringence, as explained in [21]. This phenomenons are develop at [11], [47], [48], among others. The type of fibre considered on this model was a SMF, in consequence of the complexity of the multi-mode fibres phenomenons and consequent model complexity.

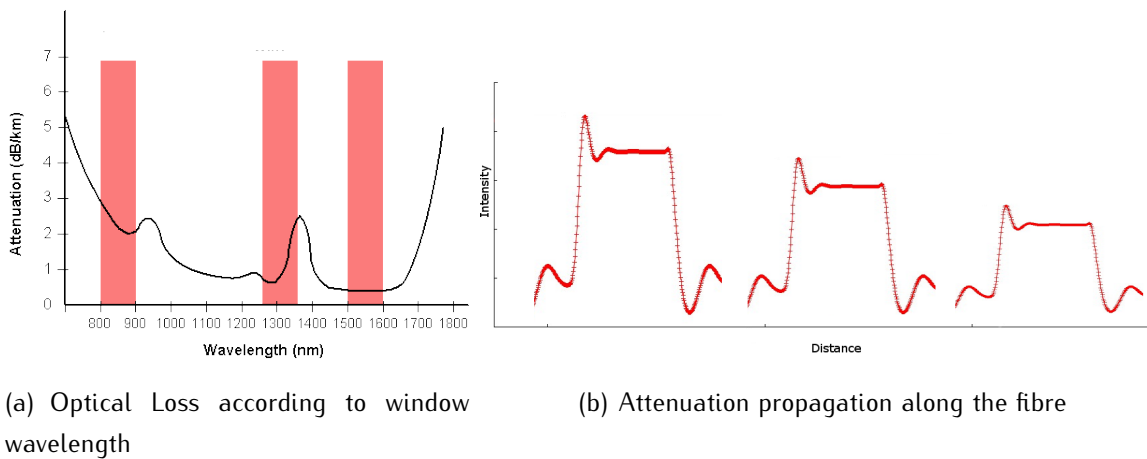


Figure 3.21: Fibre attenuation loss.

The attenuation is the reduction in the transmitted power [48]. This was the major problem on the optical communications during several years, once the first fibres produced presented a very high attenuation, around 1000 dB/Km, making the

transmission impossible. Nowadays, there are considerable three main wavelengths that are used with low optical levels that are represented in figure 3.21(a). The attenuation happens because of the absorption of the light signal through the fibre (and connections), and is caused by the impurity of the fibres and/or by the possible imperfections of the fibre. This phenomenon is represented in figure 3.21(b).

The attenuation is modelled in application according to the formula described in section A.19, getting as parameters the values configured at the UI.

On optics, the dispersion is the spreading of the pulse along the fibre. The bases of this phenomenon are explained at [47], but the importance for this dissertation is the consequences of it on optical transmission. The dispersion is caused because of the different speeds that different wavelengths propagates, causing the pulse to spread in consequence of that. The influence of this is schemed in figure 3.22, where is visible the initial pulses losing the intensity and spreading in time.

The dispersion can be inter-modal (or modal), or intra-modal. The first one occur only in multi-mode fibres, considering only the second one on our case, once this type (intra-modal) affects mono-mode and multi-mode fibres. The intra-modal dispersion can have two causes:

- Light propagates differently in the core and in the cladding of the fibre;
- Different wavelengths travel in wave-guide structures at different speed.

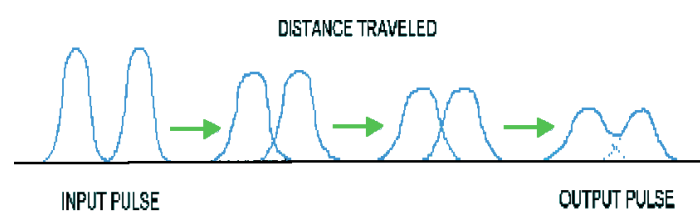


Figure 3.22: Fibre dispersion phenomenon.

The dispersion is a parameter defined by the user, and computed by the formula described in section A.18 along with the fibre transfer function. This parameter unit is in $ps/(Km.nm)$.

The fibre module also allows the addition of noise to the signal. the noise spectral density is modelled as a noise parameter defined by the user, restricted in a range

between 0 and 1. This value is added to the phase component of the signal, causing the white noise on the channel.

3.4.5 Demultiplexer

The multiplexer allows the separation of the channels multiplexed with 3.4.3. The function of this component is to invert the multiplexer function with filters to select the channel and to reduce the interferences between the neighbour channels. This action is done using an optic filter. The optic filters implemented are Fabry-Perot filters, the single cavity, double-passage, three mirror and interference [21]. An example of each optical filter can be analysed in figure 3.23, where a plot of the transfer functions is represented.

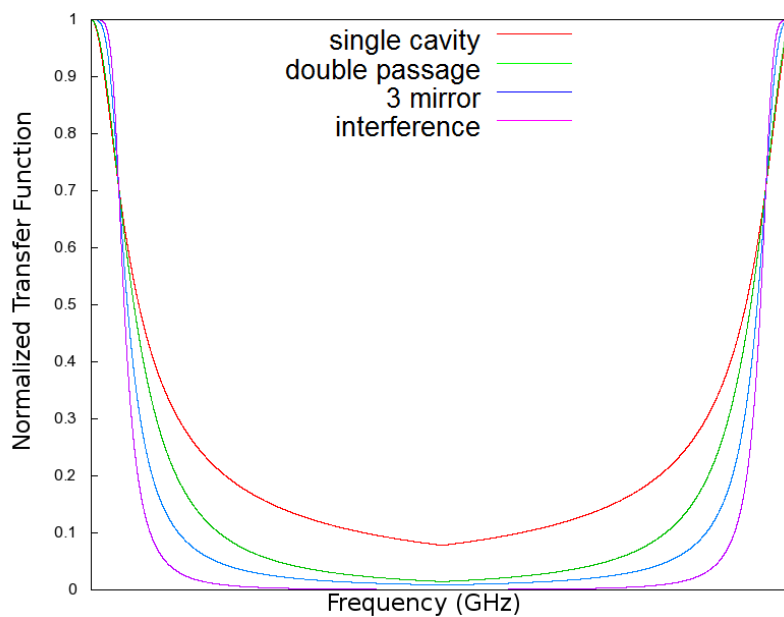


Figure 3.23: Fabri Perot Filter transfer functions.

The Fabri-Perot filter of single cavity was modelled by the formulas described in appendix A, with the formulas described in sections A.20-A.22. The value of the filter fineness considered as default was 150, once this is a value considered suitable in [21].

The double-passage Fabri-Perot filter, as suggested, uses a double passage of a single cavity filter, considering the same filter, *i.e.*, considering the double-passage filter with two equal cavities. Considering this, the value of *FWHM* has the following

value [21]:

$$FWHM_{DCF} = \sqrt{\sqrt{2} - 1} \frac{FSR}{F_f}. \quad (3.7)$$

The three mirror is modelled by the transfer function represented at appendix A in sections A.25–A.26. The interference filter can be modelled by a 3rd order Butterworth filter. This filter induces low losses on the signal, about 1dB at most. The filter used was a Matlab[©] 3rd order Butterworth filter. This method was used with Matlab[©] because of the difficulties on performing phase filtering with an accurate method to long signals (around 16K elements). To solve this problem, a Matlab[©] function was developed to compute and apply the filter to a signal, developing after a C++ Class to convert a double array to a Matlab[©] datatype and vice-versa in order to use the exported Matlab[©] filter in the simulator.

3.4.6 Electrical Filter

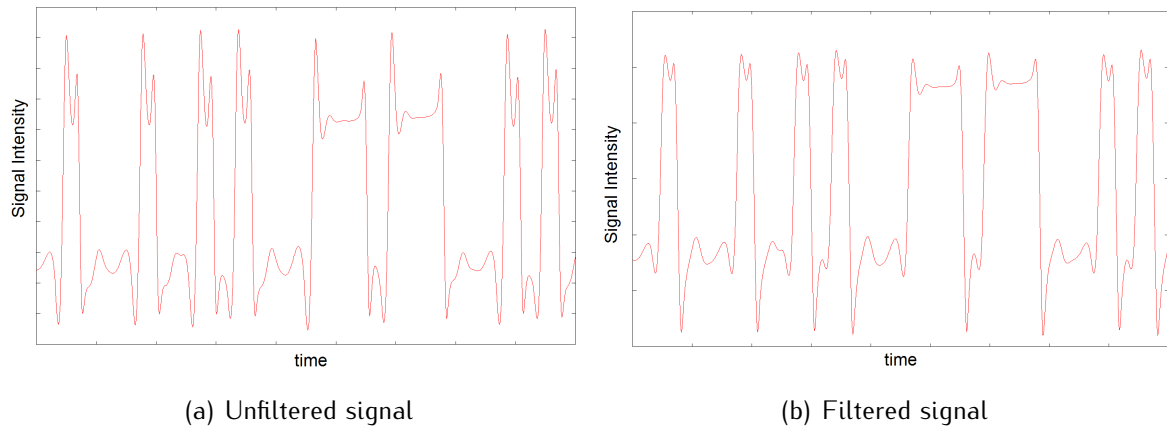


Figure 3.24: Plot of Signal Filter example.

The signal filter is a filter that is applied to remove the very high frequencies interference of the signal. The signal is modified during the transmission by physical and electrical phenomena, adding some very high frequencies among other, that difficult the signal analysis. The use of the electrical filter is an attempt to remove the very high frequency interferences and to narrow the bandwidth of the noise. The signal used as result of the demultiplexer (or the fibre if only one channel is defined) is filtered with the aim of remove information not relevant, improving the signal.

The filter applied is an electronic filter to remove the interference generated by the other channels and the transmission. The filter applied is a Low pass first order RC filter, *i.e.* a filter that only filters the low frequency of the channel. The filter used has a formula described in section refapp:rc1. The filter requires the information of the cut-off information, being this information provided by the user, once the filtering might be more or less restricted.

3.4.7 Signal Reconstruction

The signal reconstruction is the step that analyse the signal, generates the eye-diagram and reconstruct the signal based on the threshold defined or computed. The signal reconstruction is done using the formula 3.8.

$$\begin{cases} 1, & x_i \geq x_{threshold} \\ 0, & x_i < x_{threshold} \end{cases} \quad (3.8)$$

The *Threshold* value can be defined by the user or computed by the simulator. The UI shows the user the eye diagram, where the user have the information of the (x, y) point of the position of the mouse. When the user click the mouse the value is defined as the *threshold* point of the channel, generating an axis on the point defined. In case of a continuous simulation and the *threshold* value is computed by the simulator, the value is obtained computing the eye aperture, considering the coordinates of the maximum value as the *threshold*. The y value of the point is computed using the mean value of the array and the x value is defined with the max value between the highest value below y and the lowest above y .

3.4.8 Result

On the result tab or step, the array is resized, considering the values added in order to fulfil the 2^n requirement. The 2^n requirement is mandatory once the FFT have a great performance when the input fulfil this requirements, generating an error when the input does not fulfil this requirement. To perform this task a method was created in order to discard the irrelevant information when the information is reconstructed and to plot the data. The method to remove the information is the inverse of the method to add information, *i.e.*, is removed half of the information of the beginning of the array

and the rest from the end. Besides the array resize, the Class created allows the user to get information about the signal, like the error percentage, and the burst error information.

To perform the error analyses the method compares the initial array, from the *Signal Parameters* object, and compares element by element with the result. The user can also save the data to files, as a binary file, as a file with the signal, or the plot of the signal. To save the data as a binary file the information is reconstructed using bitwise operation in order to convert the data in bytes and write in a file, once C++ accept bytes as the smallest data type to manage. The plot option is done similar to the one used to plot the data in the step-by-step simulation, *i.e.*, using the Gnuplot software.

3.5 Validation of the Simulator

In order to ensure the results are correct, *i.e.*, to validate the simulator, eye diagrams were generated for the transmission a signal with 10, 20 and 40 Gbps as presented above.

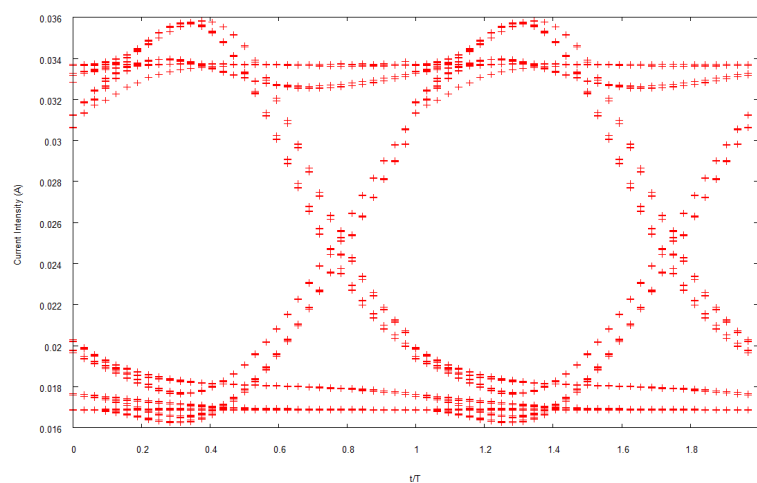


Figure 3.25: Eye diagram of a 10 Gigabit per second transmission simulation.

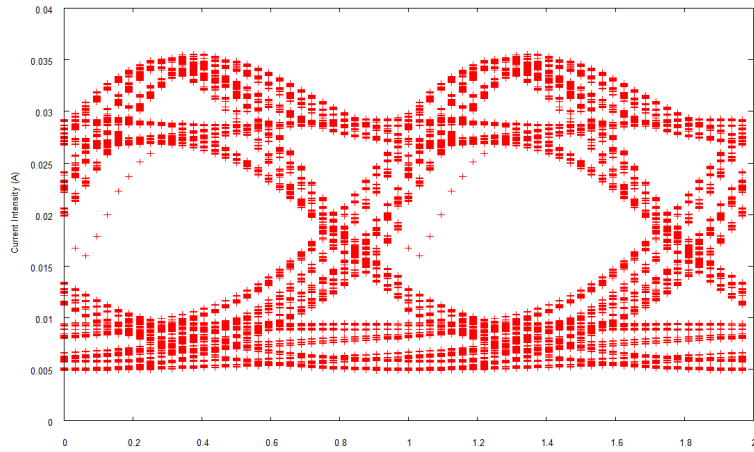


Figure 3.26: Eye diagram of a 20 Gigabit per second transmission simulation.

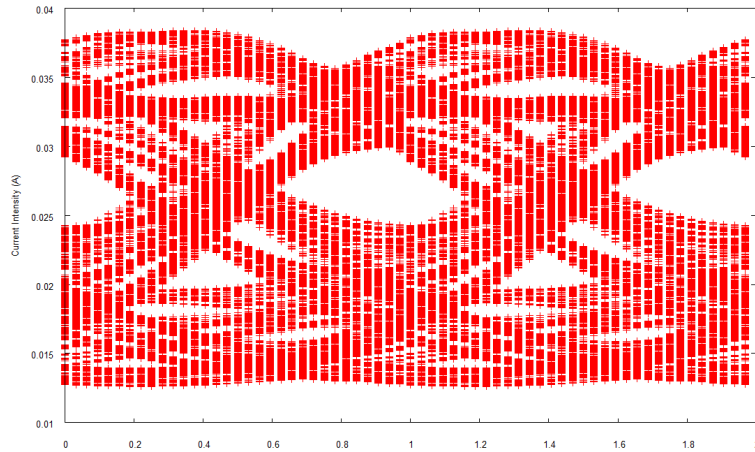


Figure 3.27: Eye diagram of a 40 Gigabit per second transmission simulation.

The simulator was validated using a visual inspection of the eye diagrams generated by the solution developed, comparing with the ones generated in [21] with similar parameters of simulation. The eye diagram in figure ?? is visually similar to the one presented in chapter 4 of [21]. With this parallel characteristics we can accept the proofing of the data generated by the solution developed.

3.6 Summary

On this chapter the solution proposed is described in detail, with an explanation of all the elements and components, along with the GUI developed. All of the components

are explained in detail, including the parameters acceptable as well as the options the user have available in each step of the process of the simulation. The justification of the choices taken during the development of the simulator proposed are also analysed and explained.

This page was intentionally left blank

Chapter 4

Simulations Results and Discussion

In this chapter the results of simulation runs with the solution developed are presented and discussed. The conditions of simulations are also described. The simulations were run at a Hp laptop, with an IntelTMCoreTM2 Duo CPU @ 2.53GHz with 3 GB of RAM. In order to validate the simulator,the eye-diagram is used, once this component, as explained in [39] and [40], provides important information.

4.1 Simulation Scenario

The simulation was done using Multiple Description Coding (MDC) coded groups of images, with the coder developed by [22] with improvements done on [23]. The coder is described with more detail in figure 4.1 and, as represented, based on [22], [23] and [30].

The specialized side decoder that uses JPEG 2000, developed at [30], receives each description and fulling decoding the input, producing two distinct outputs: the decoded image and the description information. The decoded image is the output of the JPEG 2000 decoder, and the description information is a bit-stream containing key-information of the description decoding process, including:

- MQ-Encoder registers
- MQ-Decoder registers
- J2K code-stream

The description information is analysed by the Central Decoder, component 5 in figure 4.1, using the information to make key decisions about the best information to use to optimally reconstruct the original data. The model described is based on a two-channel MDC in order to simplify, being scalable to any given configuration. The JPEG 2000 encoder represented in [30] produces code-streams provided with some special registers as described in [23], the side decoder can successfully detect which J"K segments are corrupted by comparing the encoder-saved special registry with the corresponding decoder registry, as explained in [23]. When the central decoder runs, each available description information (fig. 4.1, D1 and D2) contains all the information needed to decide which description segment to use so that the reconstructed image Signal-to-Noise Ratio (SNR) is maximized.

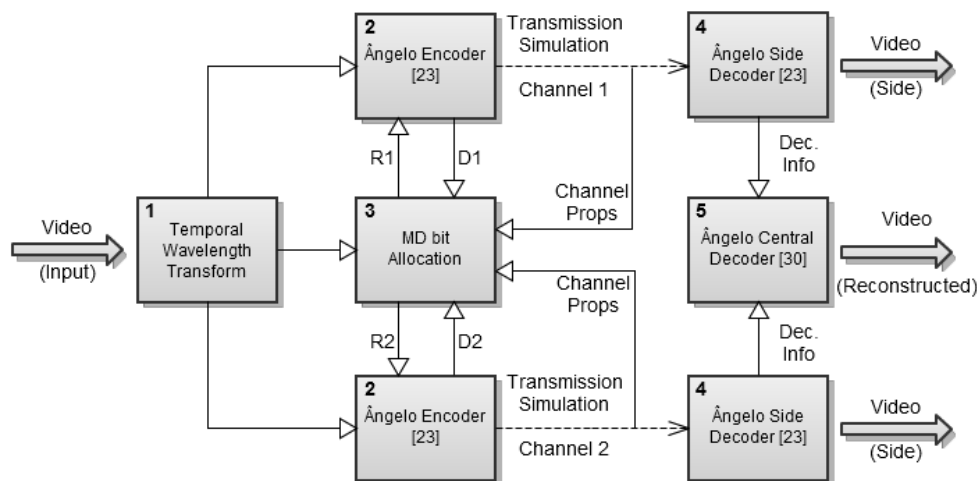


Figure 4.1: JPEG 2000 Video coder Scheme for two channels.

The Multiple Description (MD) video coder described in figure 4.1 follows the scheme proposed by [22] and based on [23] and [49] improved by JPEG 2000 as presented in [30]. The video coding scheme starts by performing the temporal motion-compensated wavelet-transform (WT) presented in [50]. The MD bit-allocation, component 3 of 4.1, efficiently distributes resources to descriptions based on the redundancy parameter [49] and temporal sub-band rate-allocation is achieved by a rate-distortion algorithm [50] that includes an MD-bitallocation that efficiently distributes resources to descriptions based on the channel characteristics [49]. The redundancy parameter aims to adapt redundancy to the current channel characteristics, within the flow. Afterwards, they are encoded with [30] using the target rate pretended and the bit-stream is

transmitted by the simulator developed. In the decoding process, an inverse temporal motion-compensated WT takes place and temporal sub-bands are decoded using the method explained in [30] section II.

The transmission simulation was done sending each channel in a different channel by the simulation. This was done using a Python script to merge the several resulting files of the coding, creating a single file with the files of each channel. The file merge does not affect the files information, once the file names and size is kept in a different file in order to reconstruct the data keeping the file names to decode the Group of Pictures (GOP). After the simulation the result file is splitted to create the original files to compute de MDC decoding. This sequence is represented in figure 4.2

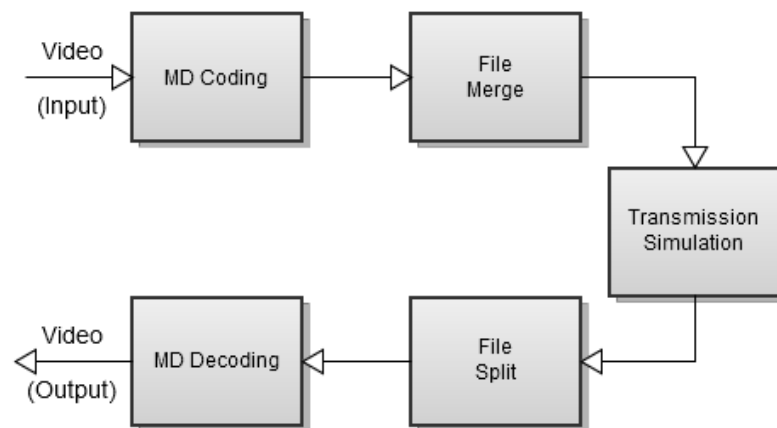


Figure 4.2: File merge and split flowchart.

The MDC coder was used with three (2.0) temporal decomposition levels with 1/4 pixel motion vectors. The sequence used was "foreman.cif" was encoded with a total target rate $R_t = R_1 + R_2 = 2000$ kbps, where only the MQ-encoder-saved register overhead is not accounted for. The spacial JPEG 2000 coder was also tuned for error-resilience, using a (irreversible) 9-7 DWT, while EBCOT code-blocks were size 8x8. The noise introduced in the channel is done during the simulations by the fibre component, being this value dependent to the fibre length.

The simulations were done using the parameters validated by [21] on chapter 3, using a 10 Gbps transmission with the 20 Gbps parameters. The parameters were:

- Bit Rate: 10 Gigabit per second (Gbps)
- Points per Bit: 32

- Kind of Raise/Fall Time: 80/20
- Raise and Fall Times: 10 ps
- Base Intensity: 30 mA Set of Laser Parameters: The ones for 20 Gbps
- Frequency Shift (Multiplexer and Demultiplexer): 100 Ghz
- First Order Dispersion: 17 ps/(Km.Nm)
- Fibre attenuation: 0.22 dB/Km
- Cut-off Frequency of the Third Order Butterworth Filter: 10 GHz

The others parameters are dependent of the fibre length, such as the 1st order RC filter and the modulation intensity, being these values defined based on the results presented by [21] and detailed in table 4.1.

	Modulation Intensity (mA)	RC Cutoff Frequency (0-1)GHz*10)
20	35	0.575
30	40	0.35
40	43	0.24

Table 4.1: Simulation parameters to 20, 30 and 40 km transmission.

In order to obtain the results, the Demultiplexer filter used was the 3rd order Butterworth filter with a cutoff frequency of 10 GHz.

4.2 Simulation Results and Discussion

The results presented were obtained using the proposed simulator with the parameters defined previously. In table 4.2 the results of the transmission simulation are presented, providing the comparison between the MDC video transmitted in different length (between 20 and 40 km) with and without redundancy.

The results were computed using the first eight (8) frames of foreman video coded with the MD coder described at [22, 23] and [30]. The video was coded from a YUV

format video to a MD coding (defined with lossy compression). The pSNR is a value that represent the peak signal-to-noise ratio, in decibels, *i.e.*, the relation between the maximum power value of a signal and the noise power. To compute the pSNR the Mean Square Error is needed. MSE is a measure of how close a fitted line is to data points. For every data point, the distance from the point to the corresponding y value on the curve fit is computed. After that the mean of the values obtained is computed. The smaller the Mean Squared Error, the closer the fit is to the data.



Figure 4.3: Simulation results for transmission over 20, 30 and 40 km with 0.0, 0.2 and 1.0 redundancy, respectively.

The results obtained with the simulator can allow the user to adjust the simulation in order to simulate a specific configuration, allowing to have a notion of the coding needed in order to deliver a good image quality. As presented on the resulted images, the MD coding presents advantages when compared to the coding without redundancy (0.0). This difference is more clearer with a more noisy channel as the example of the

		MDC (0.0)		MDC (0.2)		MDC (1.0)	
		pSNR (dB)	MSE	pSNR (dB)	MSE	pSNR (dB)	MSE
20 Km	D1	32.853943	$3.37 * 10^7$	32.853943	$3.37 * 10^7$	31.670766	$4.43 * 10^7$
	D2	22.210338	$3.91 * 10^8$	29.317750	$7.61 * 10^7$	31.670766	$4.43 * 10^7$
	Cd	33.273801	$3.06 * 10^7$	33.159646	$3.14 * 10^7$	31.670766	$4.43 * 10^7$
30 Km	D1	32.848661	$3.37 * 10^7$	32.407951	$3.25 * 10^7$	31.656542	$4.44 * 10^7$
	D2	22.210004	$3.91 * 10^8$	26.259481	$1.54 * 10^8$	31.670766	$4.43 * 10^7$
	Cd	33.273725	$3,06 * 10^7$	33.056269	$3.22 * 10^7$	31.670766	$4.43 * 10^7$
40 Km	D1	9.437153	$7.40 * 10^9$	14.549325	$2.28 * 10^9$	20.956248	$5.22 * 10^8$
	D2	8.683672	$8.80 * 10^9$	10.145322	$6.29 * 10^9$	11.371142	$4.74 * 10^9$
	Cd	8.703115	$8.77 * 10^9$	24.301029	$2.42 * 10^8$	23.480938	$2.92 * 10^8$

Table 4.2: Simulation results to 20, 30 and 40 km transmission with 0.0, 0.2 and 1.0 redundancy, respectively.

40km transmission. Considering the simulator usage, the experience was very good, the structure defined fits the purpose that it was designed for, giving the user a good user experience when using the simulator developed.

Chapter 5

Final Conclusions and Future Work

5.1 Main Conclusions

Completing the development and the test steps, we can conclude that this tool is very useful to use in order to simulate the characteristics of an optical communication system, and take advantage of that knowledge to make decisions, for example, the ideal redundancy to had to a Multiple Description (MD) coded image/video in order to provide the user the best experience considering the possibilities.

The architecture of the proposed simulator proved to be suitable to overcome the problem that was proposed to solve, once the simulator development was separated into parts, being each one developed and tested in a separate way, making it more easy to group all parts together and improve each part in separate. Concerning the results, the simulator produced results useful for the Multiple Description Coding (MDC) analysis, being also possible to simulate the transmission of other types of data. With the results that can be produced with the simulator developed, an analyses can be produced in order to the set the coding of the video/image with the best parameters, adapting the codification with the transmission conditions. As reported on the results, the MDC proposed by [22, 30] provides good results and advantages when comparing the transmission of the coded images/videos with and without redundancy.

5.2 Directions for Future Work

The simulator developed presents good results, enhancing once again the advantages of MD coded images and videos when transmitted on noisy channels. Besides the solution developed fulfil the requirements defined some features would be interesting to implement, in order to improve the solution presented. One of the feature that would be interesting would be to take advantage of the multi-core architecture widely used nowadays in order to speed up the execution time. The transition to a multi-platform would increase the potential users of the simulator.

Considering improvements in the simulator itself, it would be interesting in develop new possibilities of simulation, as the multiplexing and demultiplexing, once this feature is being improved to allow the transmission of more channels with a lower frequency drift as long as other multiplexing methods. An important component developed, the eye diagram threshold computation is done in a way that might not provide the best results and a better analysis of the eye diagram with an algorithm that could take advantage of the signal characteristics and/or prediction of the transmitted signal could provide better results.

References

- [1] V.K. Goyal. Multiple description coding: compression meets the network. *Signal Processing Magazine, IEEE*, 18(5):74–93, Sep 2001.
- [2] John Crisp and Barry Elliott. *Introduction to Fiber Optics*. Newnes, 2005.
- [3] Ajoy Ghatak and K. Thyagarajan. *Introduction to Fiber Optics*. Cambridge University Press, March 1997.
- [4] Gerd Keiser. *Optical Communications Essentials*. McGraw-Hill Professional, 1 edition, July 2003.
- [5] Stefan Nilsson-Gistvik. *Optical Fiber Theory for Communication Networks*. Ericsson Network Technologies AB, 2002.
- [6] Ivan P. Kaminow, Tingye Li, and Alan E. Willner, editors. *Optical Fiber Telecommunications V A - Components and Subsystems*, volume V. Academic Press, 2008.
- [7] <http://www.youtube.com/>. Youtube. Last access: - June 28, 2010.
- [8] David Alexandre Milheiro Carvalho. Towards the detection of encrypted peer-to-peer file sharing traffic and peer-to-peer tv traffic using deep packet inspection methods. Master's thesis, University of Beira Interior, October 2009.
- [9] Donald C. Craig. Advantages and disadvantages of model-driven computer simulation. Master's thesis, Memorial University of Newfoundland, 1996.
- [10] D.M. Sullivan, Jun Liu, and M. Kuzyk. Three dimensional optical fiber simulation. In *IEEE Antennas and Propagation Society International Symposium*, volume 2, pages 1082–1085 vol.2, Aug 1999.

- [11] Sabah Hawar Saeid Al-Bazzaz. Simulation of single mode fiber optics and optical communication components using vc++. *Proceedings of International Journal of Computer Science and Network Security*, 2008, 8(2):300–308.
- [12] Rik Min. *Simulation Technology and Parallelism in Learning Environments*. Academic Book Center, 2005. <http://projects.edte.utwente.nl/pi/Book/Contents.html> - last access: 28 of June of 2010.
- [13] H. Takahashi, T. Murooka, K. Toyoshima, H. Uematsu, and T. Fujii. A high speed video/audio stream splitter for 4k digital-cinema. In *Proceedings of International Symposium on Intelligent Signal Processing and Communications, ISPACS '06*, pages 808–811, Dec. 2006.
- [14] D. Shirai, T. Kawano, and T. Fujii. 6 gbit/s uncompressed 4k video stream switching on a 10 gbit/s network. In *On proceedings of International Symposium on Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007.*, pages 542–545, December 2007.
- [15] T. Sugimoto R. Okumura A. Imai M. Iwaki K. Hamasaki A. Ando Y. Nishida K. Mitani M. Kanazawa Y. Nakayama, T. Nishiguchi and S. Kitajima. Live production and transmission of large-scale musical tv program using 22.2 multichannel sound with ultra high definition video. In *IBC2007 Conference Publication*. IBC, 2007.
- [16] Sugawara Masayuki, Kanazawa Masaru, et al. Ultrahigh-definition video system with 4000 scanning lines. In *Gazo Denshi Gakkai Kenkyukai Koen Yoko*, 2002.
- [17] J. Zubrzycki et al. Super hi-vision - the london-amsterdam live contribution link. Technical report, NHK ,BBC at al., January 2009.
- [18] Touradj Ebrahimi. Quality of Multimedia Experience: Past, Present and Future. In *Proceedings of the seventeen ACM international conference on Multimedia MM'09*, pages 3–4, New York, NY, USA, 2009. ACM.
- [19] Nils Walravens and Caroline Pauwels. From high hopes to high deficit and back: a historic overview of europe's hdtv policy and reflections towards the future of hdtv. In *Proceedings of the seventh european conference on European interactive television conference, EuroITV '09*, pages 33–40, New York, NY, USA, 2009. ACM.

- [20] M.M. Freire and H.J.A. Da Silva. Estimation of multiple-quantum well laser parameters for simulation of dispersion supported transmission systems at 20 gbit/s. *IEEE Proceedings - Optoelectronics*, 146(2):93–98, Aug 1999.
- [21] Mário M. Freire. *Estudos de Desempenho de Redes de Comunicação Incorporando Aplicações Não Lineares de Amplificadores Ópticos Semicondutores*. PhD thesis, Universidade da Beira Interior, 1999.
- [22] Manuela Pereira. *Multiple description Image and Video coding for Noisy Channels*. PhD thesis, Université de Nice Sophia Antipolis - UFR Sciences, 2004.
- [23] Angelo M. Arrifano, Manuela Pereira, Marc Antonini, and Mário M. Freire. Jpeg 2000 mq-coder registry based error detection for lossy transmission channels. volume 7257, page 72571T. SPIE, 2009.
- [24] M. Pereira, M. Antonini, and M. Barlaud. Multiple description coding for internet video streaming. In *Proceedings of ICIP 2003 - International Conference of Image Processing*, volume 3, pages III–281–4 vol.2, Sept. 2003.
- [25] <http://www.tudou.com/>. Tudou. Last access: - June 28, 2010.
- [26] N. Jayant. Subsampling of a dpcm speech channel to provide two "self-contained" half-rate channels. *Bell Systems Technical Journal*, 60(4):501–509, April 1981.
- [27] N. Jayant and S. Christensen. Effects of packet loss in waveform coded speech and improvements due to an odd-even sample-interpolation procedure. *IEEE Transactions on Communications*, COM-29(2):101–109, February 1981.
- [28] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro. Multiple description video coding for scalable and robust transmission over ip. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(3):321–334, March 2005.
- [29] Nan Zhang, Yan Lu, Feng Wu, Xiaolin Wu, and Baocai Yin. Efficient multiple-description image coding using directional lifting-based transform. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(5):646–656, May 2008.
- [30] Angelo M. Arrifano, Manuela Pereira, Marc Antonini, and Mário M. Freire. Multiple description video coding based on jpeg 2000 mq-coder registers. 2010 IEEE International Symposium on Circuits and Systems (ISCAS 2010).

- [31] M. Pereira and M. Antonini. Multi-channel adaptive bit allocation and error control for video transmission over wireless networks. Proceedings of EURASIP 13th European Signal Processing Conference (EUSIPCO 2005).
- [32] M. Pereira and M. Antonini. Multiple description coding with automatic control for video transmission over wireless networks. In *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, volume 2, pages 1112–1117 vol.2, June 2005.
- [33] V.K. Goyal and J. Kovacevic. Generalized multiple description coding with correlating transforms. *IEEE Transactions on Information Theory*, 47(6):2199–2224, Sep 2001.
- [34] Prasenjit Pal. Optical fiber communication system - theory & analysis, 2005. Electronics & Communication Engineering Department, Asansol Engineering College, Kanyapur.
- [35] Casimer DeCusatis, editor. *Fibre optic data communication: Technological trends and advances*. Academic Press, 2002.
- [36] Jean-Baptiste Thomine & Loic Le Fur. Optical fibre transmission - executive summary. <http://www.axiom.fr/upload/fiber-techno.pdf>. Last access: - June 28, 2010.
- [37] S.M. Rossi and E. Moschim. Simulation of high speed optical fiber systems using pc-simfo. In *Electronic Components and Technology Conference, 1997. Proceedings., 47th*, pages 1289 –1294, 18–21 1997.
- [38] ITU-T. *Spectral grids for WDM applications: DWDM frequency grid*, 2 2002.
- [39] John G. Proakis. *Digital Communications*. McGraw-Hill, 3 edition, 1995.
- [40] A. Bruce Carlson. *Communication Systems - An Introduction to Signals and Noise in Electrical Communication*. McGraw-Hill, 3 edition, 1986.
- [41] Russ Miles and Kim Hamilton. *Learning UML 2.0*. O'Reilly Media, Inc., 2006.
- [42] Mohammad Reza Nami. A comparison of object-oriented languages in software engineering. *SIGSOFT Softw. Eng. Notes*, 33(4):1–5, 2008. last access: 09 of August of 2009.

- [43] Gnuplot homepage. Website. <http://www.gnuplot.info>, Last access: - June 28, 2010.
- [44] M. Ishikawa, R. Nagarajan, T. Fukushima, J.G. Wasserbauer, and J.E. Bowers. Long wavelength high-speed semiconductor lasers with carrier transport effects. *IEEE Journal of Quantum Electronics*, 28(10):2230–2241, Oct 1992.
- [45] William H. Press. *Numerical Recipes : the art of scientific computing*. Cambridge University Press, 3 edition, September 2007.
- [46] Voltech, http://www.feebel.be/files/Feebel-Web/files/2008electronics/presentations/3A_FFT_Analysis.pdf. *FFT Analysis - Understanding the limitations of synchronisation and frequency resolution*, July 2009. Last access: - June 28, 2010.
- [47] M. J. Stephen. First-order dispersion forces. *The Journal of Chemical Physics*, 40(3):669–673, 1964.
- [48] Ashley Hagans and Kyle Paulson. Attenuation in fiber optics. Website, july 2001. <http://education.uncc.edu/cmste/papers%20-%20OLD/Attenuation%20in%20Fiber%20Optics.doc> - last visit - 28 of June of 2010.
- [49] M. Antonini M. Pereira and M. Barlaud. Multiple description image and video coding for wireless channels. In *EURASIP Signal Processing: Image Communication*, number 10, pages 925–945, Nov. 2003.
- [50] Marco Cagnazzo, Thomas André, Marc Antonini, and Michel Barlaud. A model-based motion compensated video coder with jpeg2000 compatibility. In *Proceedings of ICIP 2004- International Conference of Image Processing*, pages 2255–2258, 2004.

This page was intentionally left blank

Appendix A

Mathematical Functions Implemented

This appendix describes the mathematical formulas modulated by M. Freire in [21] that were implemented on the simulator. The formulas are grouped according to the component/object they model.

A.1 Laser Driver

The laser driver gives the current applied to the laser, $I(t)$ and is given by:

$$\sum_{K=1}^{L_{PRBS}+1} I_K [t - (K - 1)T], \quad (\text{A.1})$$

where T represents the bit period, $L_{PRBS}+1$ represents the total length of a sequence $A \equiv \{a_k\}$. The values of $I_K(t)$ is defined on the time interval of $0 \leq t \leq T$ is modelled by:

$$I_1(t) = \begin{cases} I_b & , a_1 = 0 \\ I_b + I_m & , a_1 = 1 \end{cases} \quad (\text{A.2})$$

$$I_k(t) = \begin{cases} I_b + I_m & , a_k = 1 \quad \& \quad a_{k-1} = 1 \\ I_b + I_m e^{-\frac{C_t t}{\tau_f}} & , a_k = 0 \quad \& \quad a_{k-1} = 1 \\ I_b & , a_k = 0 \quad \& \quad a_{k-1} = 0, k \in [2; L_{PRBS} + 1] \\ I_b + I_m \left(1 - e^{-\frac{C_t t}{\tau_f}}\right) & , a_k = 1 \quad \& \quad a_{k-1} = 0 \end{cases} \quad (A.3)$$

where I_b is the laser polarization intensity, $I_m (I_m \geq 0)$ represents the modulation amplitude. τ_f and τ_r represents the time or rise and of fall, by this order, and constant $C_t = \ln 10 + \ln 0.9$ if up/down time is 10% and 90% and $C_t = \ln 5 + \ln 0.8$ if up/down time is 20% and 80%.

A.2 Multiple Quantum Well Laser

The laser model has several parameters that can be configured:

V_W - Quantum Well Volume (μm^3)

V_S - Separate Confinement Heterostructure (SCH) Volume (μm^3)

Γ - Optical Confinement Factor

β_{sp} - Spontaneous Emission Factor

g_0 - Linear Coefficient of Optical Gain (m^3/s)

g_b - SCH Coefficient (m^3/s)

N_0 - Transparency Density (m^{-3})

τ_{n0} - Bimolecular Recombination Lifetime (ns)

τ_c - Transport Time through SCH (ps)

τ_e - Thermionic Emission Time (ps)

τ_p - Photon LifeTime (ps)

η - Face Quantic Differential Efficiency (W/A)

ε - Gain Compression Factor (m^3)

α - Spectral Dilatation Factor

K_T - Thermal Constant (A^{-1})

λ_0 - Emission Wavelength (nm)

The modelling of the Multiple Quantum Well (MQW) laser is done using differential equations, developed by [44] *et al.* that can be written in volumetric density form, as follows:

$$\frac{dN_b}{dt} = \frac{I}{qV_w} - \frac{N_b}{\tau_c} - \frac{N_b}{\tau_n} + \frac{N_w}{\tau_e}, \quad (\text{A.4})$$

$$\frac{dN_w}{dt} = \frac{N_b}{\tau_c} - \frac{N_w}{\tau_e} - \frac{N_w}{\tau_n} - g_0 \frac{N_w - N_0}{1 + \epsilon S} S, \quad (\text{A.5})$$

$$\frac{dNS}{dt} = \Gamma g_0 \frac{N_w - N_0}{1 + \epsilon S} S - \frac{S}{\tau_p} + \Gamma \beta_{SP} \frac{N_w}{\tau_n}, \quad (\text{A.6})$$

$$\frac{d\phi}{dt} = \frac{\alpha}{2} \Gamma g_0 (N_w - N_{wr}) + (1 - \Gamma) g_b \frac{V_w}{V_s} (N_b - N_{br}), \quad (\text{A.7})$$

with

$$N_b = N_s \frac{V_s}{V_w}, \quad (\text{A.8})$$

where N_s is the SCH Volume, N_b is a fictitious density, correlated with N_s by A.8. N_w is the density of the quantum well carriers density, S is the density of photons in the cavity, I is the injected current, q is the electron charge, V_w is the quantum well volume, V_s is the SCH and barriers volume, N_0 is the carriers density on transparency, G_0 is the linear coefficient of optical gain, ϵ the gain compression factor, τ_n the bimolecular recombination lifetime, τ_c the carriers transport time through SCH, τ_e the Thermionic Emission Time, τ_p the photon lifetime, Γ the optical confinement factor, β_{sp} the spontaneous emission factor, ϕ the phase of the electric field transmitted by the laser, g_b the SCH coefficient, α the spectral dilatation factor, N_{wr} the carriers density on the quantum well for the reference polarization level and N_{br} the fictitious density of the carriers for the reference polarization level.

As explained on chapter 3 of [21], the laser threshold intensity have a known dependency of temperature. Considering this model, the dependence of τ_{n0} (Bimolecular Recombination Lifetime) with temperature is expressed by A.9.

$$\tau_n = \tau_{n0} e^{-K_T l_0}, \quad (\text{A.9})$$

where τ_{n0} represents the bimolecular recombination lifetime for a reference temperature, l_0 the mean intensity applied to the laser, K_T the constant that depends of the laser thermal characteristics. The optic power emission is given by A.10.

$$P(t) = \frac{V_w \eta h c}{\Gamma \tau_p \lambda_0} S(t), \quad (\text{A.10})$$

As explained in 3.4.1 the equations A.4-A.7 are numerically solved for an arbitrary injected intensity (I), using the fourth order *Runge-Kutta* method. These equations, for a stationary intensity, can be solved with the equations A.4-A.7, once the temporal derives are annulled. For an I_b intensity the equations A.11-A.17 can be obtained for a stationary regime.

$$S_0 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (\text{A.11})$$

with

$$a = -\frac{1}{\tau_p} \left[\left(1 + \frac{\tau_c}{\tau_e} + \frac{t a u_c}{\tau_n} \right) \varepsilon + \left(1 + \frac{\tau_c}{\tau_n} \right) \tau_n g_0 \right], \quad (\text{A.12})$$

$$b = \frac{I_b \Gamma}{q V_w} (\tau_n g_0 + \beta_{sp} \varepsilon) - \left(1 + \frac{\tau_c}{\tau_e} + \frac{\tau_c}{\tau_n} \right) (N_0 \Gamma g_0 + \frac{1}{\tau_p}) + \left(1 + \frac{\tau_c}{\tau_n} \right) \beta_{sp} N_0 \Gamma g_0 \quad (\text{A.13})$$

$$c = \beta_{sp} \Gamma \frac{I_b}{q V_w}, \quad (\text{A.14})$$

$$N_{w0} = \frac{\frac{I_b}{q V_w} + \left(1 + \frac{\tau_c}{\tau_n} \right) \frac{N_0 g_0 S_0}{1 + \varepsilon S_0}}{\left(1 + \frac{\tau_c}{\tau_e} + \frac{\tau_c}{\tau_n} \right) \frac{1}{\tau_n} + \left(1 + \frac{\tau_c}{\tau_n} \right) \frac{g_0 S_0}{1 + \varepsilon S_0}}, \quad (\text{A.15})$$

$$N_{b0} = \frac{\frac{I_b}{q V_w} + \frac{N_{w0}}{\tau_e}}{\frac{1}{\tau_c} + \frac{1}{\tau_n}}, \quad (\text{A.16})$$

The complex component of the signal is represented by A.17,

$$E_l(t) = \sqrt{P(t)}e^{j\phi(t)}, \quad (\text{A.17})$$

where $\sqrt{P(t)}$ represents the modulus of the signal and $\phi(t)$ the phase. The complex number is in the polar representation, using Euler's generic formula $z = re^{i\varphi}$.

A.3 Single-Mode Fibre

The modelling of the Single-Mode Fibre (SMF) implemented was develop by [21], ignoring some factor that can affect the signal but can be easily overcome using filters, and ignoring certain parameters, that besides are recognized have an influence that, when compared with other factors, are insignificant. Considering this, the transfer function is described at A.18.

$$H_{SMF}(\omega) = A_{SMF}e^{j\frac{\lambda_0^2}{4\pi c}DL\omega^2}, \quad (\text{A.18})$$

where ω represents the angular frequency on the base band, λ_0 the wavelength, c represents the light speed in vacuum, D the first order dispersion parameter and L the SMF length. The losses factor that is represented by A_{SMF} is determined by A.19.

$$A_{SMF} = 10^{-100\alpha_{SMF}L}, \quad (\text{A.19})$$

where α_{SMF} represents the attenuation coefficient, represented in dB/Km.

A.4 Demultiplexer

The demultiplexer was implemented based on [21], separating the channels using an optic filter and decreasing the noise interferences with electronic filters.

A.4.1 Fabri-Perot Single Cavity

This filter was implemented according to the formula:

$$H_{SCF}(\omega) = \frac{T_f}{1 - R_f e^{j\frac{(\omega-\omega_c)}{FSR}}}, \quad (\text{A.20})$$

where T_f represents the transitivity ($T_f = 1 - R_f$), R_f represents the reflectivity and $F_c = \frac{\omega_c}{2\pi}$ represents the central frequency of the filter. The FSR parameter represents the free spectral range. The Full Width at Half Maximum (FWHM) is dependent from FSR and the fineness by:

$$FWHM_{SCF} = \frac{FSR}{F_f}, \quad (\text{A.21})$$

The fineness can be calculated using:

$$F_f = \frac{\pi\sqrt{R_f}}{1 - R_f}. \quad (\text{A.22})$$

A.4.2 Fabri-Perot Three Mirror

The three mirror Fabri-Perot filter is modeled by:

$$H_{TMF}(\omega) = \frac{\sqrt{T_0} T_1}{1 - 2\sqrt{R_0 R_1} e^{-\frac{j\omega}{FSR}} + R_1 e^{\frac{2j\omega}{FSR}}}, \quad (\text{A.23})$$

where R_0 and R_1 represent the interior and exterior mirror reflectivity, T_0 and T_1 the correspondent transmittance, and FSR the free spectral region in each half cavity. In the conditions used, considering the frequency response is a second order Butterworth and ignoring the losses, R_0 can be obtained by:

$$R_0 = \frac{4R_1}{(1 + R_1)^2}, \quad (\text{A.24})$$

Considering A.24, and ignoring the losses, is possible to prove that the FWHM of 3dB of the frequency response of the Three Mirror Filter is:

$$FWHM_{TMF} = \frac{FSR}{\pi} \arccos \left(1 - \sqrt{1 + \frac{\beta}{4R_1}} \right), \quad (\text{A.25})$$

with

$$\beta = 2T_0 T_1^2 + R_1(2 - R_1 - 4R_0) - 1. \quad (\text{A.26})$$

In the simulator the value considered was $R_1 = 0.8$ [21].

A.4.3 First Order Resistor-Capacitor Filter

The RC filter has a frequency response modelled by:

$$H_{RC}(\omega) = \frac{1}{1 + j\left(\frac{\omega}{2\pi f_p}\right)}, \quad (\text{A.27})$$

where f_p represents the cut frequency at 3dB.

