



**Bruno Jorge Ramos
Pereira**

**Simulador de sistemas de comunicações óticas
(OSIP)**



**Bruno Jorge Ramos
Pereira**

**Simulador de sistemas de comunicações óticas
(OSIP)**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Dr. Paulo Miguel Nepomuceno Pereira Monteiro, Professor Doutor do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Dr. Miguel Vidal Drummond, Investigador da Nokia Siemens Networks Portugal S.A.

Dedico este trabalho aos meus pais e ao meu irmão pelo incansável apoio.

O júri

Presidente

Prof. Doutor António Luís Jesus Teixeira
Professor associado do Departamento de Eletrónica,
Telecomunicações e Informática da Universidade de Aveiro

Prof. Doutor Henrique José Almeida da Silva
Professor associado da Universidade de Coimbra - Faculdade de
Ciências e Tecnologia

Prof. Doutor Paulo Miguel Nepomuceno Monteiro
Professor associado do Departamento de Eletrónica,
Telecomunicações e Informática da Universidade de Aveiro (orientador)

Doutor Miguel Vidal Drummond
Investigador da Nokia Siemens Networks Portugal S.A. (co-orientador)

Agradecimentos

Agradeço ao Professor Doutor Paulo Nepomuceno Monteiro, ao Professor Doutor Rui Sousa Ribeiro e ao Doutor Miguel Drummond pela disponibilidade e apoio na orientação desta dissertação. Agradeço também ao Instituto de Telecomunicações enquanto instituição de acolhimento e a todos os colegas que contribuíram para a concretização deste trabalho. Aos meus pais e ao meu irmão agradeço todo o empenho, apoio e investimento na minha formação.

Palavras-chave

OSIP, simulação, fibra ótica, pós-compensação, processamento digital, detecção coerente

Resumo

Nos últimos anos tem-se assistido a um grande aumento da largura de banda nas redes óticas, que é resultado do aumento do tráfego de dados. Para responder às necessidades e assim aumentar a capacidade das redes tem-se vindo a intensificar as atividades de investigação, procurando novos componentes que proporcionem uma melhor eficiência dos sistemas. O aumento da complexidade dos novos dispositivos e técnicas de transmissão levam a que a análise e projeto dos sistemas se torne cada vez mais difícil. Como resultado são necessárias ferramentas de análise e projeto assistidas por computador, o que levou ao desenvolvimento de um software de simulação implementado em ambiente Matlab[®]. Este software designado de OSIP, é de utilização livre e de uma alargada compatibilidade. O trabalho apresentado nesta tese visa o melhoramento do OSIP com vista a permitir a simulação de sistemas coerentes, o que implica a incorporação de novos componentes assim como a implementação de métodos de simulação eficientes. A preservação da informação de amplitude e fase inerente aos sistemas coerentes permite a implementação de algoritmos de compensação digital que visam a equalização da dispersão cromática e da dispersão dos modos de polarização e a estimação e correção da fase da portadora. Para validar os modelos e métodos de controlo no simulador são realizadas algumas simulações e os resultados são comparados com valores teóricos e com resultados publicados na literatura. É dada especial atenção ao desempenho dos algoritmos de compensação das dispersões cromática e dos modos de polarização e dos métodos de estimação e correção de fase.

keywords

OSIP, simulation, optical fiber, post-compensation, digital processing, coherent detection

abstract

Over the last years the bandwidth of optical networks has continuously increased, driven by the exponential traffic growth. In order to meet the needs and thereby increase the capacity of networks research activities have been intensified, searching for new components which provide better system efficiency. The increasing complexity of new devices and transmission techniques has brought increasing difficulties in the analysis and design of the new systems. As a result, analysis tools and computer-aided design software are necessary, which led to development of a simulation toolbox in Matlab[®] language. This software named OSIP is open source and has a wide compatibility. In this thesis the simulation platform is improved in order to adapt the simulator to coherent systems, which implies the incorporation of new components and the implementation of efficient simulation methods. Making use of the amplitude and phase information enabled by coherent detection, digital impairment compensation algorithms have been implemented, aiming for the equalization of chromatic dispersion, polarization mode dispersion as well as for the estimation and correction of the carrier phase. To validate the models and methods of control in the simulator some simulations are performed. The numerical results are compared with the theoretical values and with results published in literature. Special attention is given to the performance of chromatic dispersion equalization, polarization mode dispersion equalization and carrier phase estimation.

Glossário	I
Lista de símbolos	III
Índice de figuras	V
Índice de tabelas.....	XI
1 Introdução.....	1
1.1 <i>Perspetiva histórica das Comunicações óticas</i>	1
1.2 <i>Motivação</i>	2
1.3 <i>Objetivos</i>	4
1.4 <i>Publicações do autor</i>	4
1.5 <i>Estrutura da dissertação</i>	4
2 Sistemas Óticos Coerentes.....	7
2.1 <i>Transmissor</i>	8
2.1.1 <i>Componente ótica</i>	8
2.1.2 <i>Componente digital</i>	12
2.1.3 <i>Modelo do transmissor ótico</i>	13
2.2 <i>Propagação na fibra</i>	14
2.2.1 <i>Dispersão Cromática</i>	15
2.2.2 <i>Dispersão dos Modos de Polarização</i>	16
2.2.3 <i>Efeitos não-lineares</i>	17
2.3 <i>Recetor coerente</i>	18
2.4 <i>Processamento Digital de Sinal</i>	22
2.4.1 <i>Compensação da dispersão Cromática</i>	22
2.4.2 <i>Compensação da dispersão dos Modos de Polarização</i>	25
2.4.3 <i>Estimação e correção de fase</i>	28
3 OSIP 2012.....	31
3.1 <i>Arquitetura do simulador</i>	31
3.2 <i>Novas funcionalidades</i>	40
4 Validação do Simulador	47
4.1 <i>Análise do desempenho do recetor com o ruído</i>	47
4.2 <i>Implementação de algoritmos DSP</i>	49
4.2.1 <i>Compensação da dispersão cromática</i>	50

4.2.1.1	Compensação com filtro no domínio do tempo	50
4.2.1.2	Compensação com filtro no domínio da frequência	53
4.2.1.3	Desempenho dos algoritmos TDE e FDE	54
4.2.2	Compensação da dispersão dos modos de polarização	56
4.2.2.1	Desempenho do algoritmo CMA	61
4.2.3	Estimação e correção de fase	62
4.2.3.1	Desempenho dos algoritmos de correção de fase	66
4.3	<i>WDM</i>	71
4.3.1	Dois canais SP-QPSK a 50Gbit/s	71
4.3.2	Dois canais PDM-QPSK a 100Gbit/s	74
4.4	<i>PDM-QPSK a 100Gbit/s</i>	78
5	Conclusões e Trabalho Futuro	83
5.1	<i>Tópicos de Trabalho Futuro</i>	84
	Referências	87
	Apêndice A – Cálculo teórico da SER e BER	91
	Apêndice B – Descrição dos componentes adicionados	99
	Apêndice C – Criar uma interface gráfica utilizando o GUIDE	125

Glossário

Acrónimo	Designação em Português	Designação em Inglês
ADC	Conversor analógico-digital	Analog to digital converter
BER	Taxa de erros de bit	Bit error rate
BPD	Fotodíodos balanceados	Balanced photodiode
CD	Dispersão cromática	Chromatic Dispersion
CMA	Algoritmo de módulo constante	Constant modulus algorithm
DAC	Conversor digital-analógico	Digital to analog converter
DCF	Fibra de compensação de dispersão	Dispersion compensating fiber
DD-LMS	Algoritmo de menor erro quadrático com decisão direta	Decision directed Least mean square
DGD	Atraso de grupo diferencial	Differential group delay
DP-QAM	Modulação digital em amplitude e fase nas duas componentes de polarização	Dual polarization Quadrature amplitude modulation
DSP	Processamento digital de sinal	Digital signal processing
EDFA	Amplificador de fibra dopada com Érbio	Erbium Doped Fiber Amplifier
EVM	Amplitude de erro vetorial	Error vector magnitude
FDE	Equalização no domínio da frequência	Frequency domain equalizer
FEC	Correção de erros à posteriori	Forward error correction
FFT	Transformada rápida de Fourier	Fast Fourier Transform
FIR	Resposta ao impulso finita	Finite impulse response
FSK	Modulação em frequência	Frequency-shift keying
FWM	Mistura de quatro ondas	Four-wave mixing
GUIDE	Ambiente de desenho da interface gráfica	Graphical user interface design environment
GVD	Dispersão da velocidade de grupo	Group velocity dispersion
IFFT	Transformada inversa rápida de Fourier	Inverse Fast Fourier Transform
IFWM	Mistura de quatro ondas intracanal	Intrachannel Four-wave mixing
IM/DD	Modulação em intensidade com deteção direta	Intensity modulation with direct detection
IQM	Modulador em fase e quadratura	In phase and quadrature modulator
IXPM	Modulação de fase intracanal	Intrachannel Cross-phase modulation
ISI	Interferência entre símbolos	Inter-Symbol Interference
LMS	Algoritmo de menor erro quadrático	Least mean square
LPF	Filtro passa-baixo	Low pass filter
LUT	Tabela de correspondência	Look up table
M-PSK	Modulação digital de fase de ordem M	M-Phase-shift keying

M-QAM	Modulação digital em amplitude e fase de ordem M	M-Quadrature amplitude modulation
MIMO	Múltipla entrada e múltipla saída	Multiple input multiple output
MZM	Modulador Mach-Zehnder	Mach-Zehnder modulator
NLSE	Equação não-linear de Schrödinger	Nonlinear Schrödinger Equation
OOK	Modulação digital binária em amplitude	On-off keying
OSIP	Plataforma de simulação ótica	Optical Simulator Platform
PBC	Combinador de polarizações	Polarization beam combiner
PBS	Divisor de polarizações	Polarization beam splitter
PDF	Formato de documento portátil	Portable Document Format
PDM-QPSK	Multiplexagem por divisão da polarização – Modulação digital de fase quaternária	Polarization Division Multiplexed – Quadrature Phase-Shift Keying
PN	Ruído de fase	Phase noise
PM	Modulador de fase	Phase modulator
PMD	Dispersão dos modos de polarização	Polarization mode dispersion
PRBS	Sequência binária pseudoaleatória	Pseudo-random binary sequence
PSP	Estados de polarização principais	Principal state of polarization
PSK	Modulação de fase	Phase-shift keying
SER	Taxa de erros de símbolo	Symbol error rate
SMF	Fibra ótica monomodo	Single-mode fiber
SNR	Relação sinal ruído	Signal-to-noise ratio
SP-QAM	Modulação digital em amplitude e fase numa componente de polarização	Single polarization Quadrature amplitude modulation
SPM	Modulação de fase intracanal	Self-phase modulation
SSMF	Fibra ótica monomodo padrão	Standard single-mode fiber
TDE	Equalização no domínio do tempo	Time domain equalizer
TIA	Amplificador de trans-impedância	Trans-impedance amplifier
V&V	Viterbi e Viterbi	Viterbi and Viterbi
WDM	Multiplexagem por divisão no comprimento de onda	Wavelength division multiplexing
XML	Linguagem de marcação extensível	Extensible Markup Language
XPM	Modulação cruzada de fase	Cross-phase modulation
XPS	Documento descrito em XML	XML paper specification

Lista de símbolos

Símbolo	Significado
$u(t)$	Tensão elétrica aplicada ao modulador de fase
$u_1(t), u_2(t)$	Tensão elétrica aplicada aos braços do modulador Mach-Zehnder
φ_{PM}	Rotação de fase no modulador de fase
φ_1, φ_2	Rotação de fase nos ramos do modulador Mach-Zehnder
V_π	Tensão que aplicada ao modulador causa um de 180° na fase ótica
α	Atenuação do sinal ao longo da propagação na fibra
β_2	Coefficiente de dispersão da velocidade de grupo
D	Parâmetro de dispersão
β_3	Termo de terceira ordem da dispersão cromática
γ	Coefficiente não-linear
c	Velocidade da luz no vácuo
z	Distância de propagação na fibra
t	Variável temporal
ω	Frequência angular
f	Variável de frequência
φ	Ângulo de variação das polarizações na vertical
θ	Ângulo de variação das polarizações na horizontal
τ_{DGD}	Atraso de grupo entre as duas polarizações
R	Responsividade
$\psi_{i,j}$	Coefficientes de fase do componente 2x4 90° híbrido
$E_s(t)$	Sinal de dados ótico no recetor
$E_{lo}(t)$	Sinal ótico do laser local
Δw	Desvio de frequência entre o laser do recetor e do transmissor
T	Período de amostragem das ADCs
N_0	Número de amostras da FFT sem sobreposição
N_c	Número total de amostras da FFT
N_e	Número de amostras sobrepostas na FFT
F_s	Frequência de amostragem das ADCs
μ	Parâmetro de convergência do filtro adaptativo
$h_{i,j}$	Coefficientes do filtro MIMO
$\mathcal{E}_{CMA,j}$	Sinal de erro no algoritmo CMA
$\mathcal{E}_{LMS,j}$	Sinal de erro no algoritmo LMS

Índice de figuras

Figura 1.1: Evolução das publicações relacionadas com comunicações óticas nos últimos anos.....	3
Figura 2.1: Exemplo de um modelo de um sistema de transmissão ótico, adaptado de [17]: a) Transmissor, b) Canal de transmissão, c) Recetor.	7
Figura 2.2: Modelo do modulador de fase, [7].	8
Figura 2.3: Modelo do modulador Mach-Zehnder, [7].	9
Figura 2.4: Ilustração das saídas do modulador MZM: (a) – potência ótica; (b) - campo ótico; (c) – ponto de quadratura; (d) – mínimo de transmissão; (e) – máximo de transmissão.	10
Figura 2.5: Esquema representativo do modulador IQ [22].	11
Figura 2.6: Exemplo da arquitetura do DSP para o mapeamento dos sinais: S/P - conversor série para paralelo, DAC – conversor digital para analógico.	12
Figura 2.7: Ilustração do modelo de um transmissor ótico, adaptado de [27]: DSP – processamento digital de sinal, LPF – filtro passa-baixo.	13
Figura 2.8: Ilustração do modelo de um transmissor ótico com diversidade de polarização, adaptado de [17, 27]: DSP – processamento digital de sinal, LPF – filtro passa-baixo, PBC – Combinador de polarizações.	14
Figura 2.9: Exemplo do modelo de um recetor ótico coerente, adaptado de [39, 40]: BPD – fotodíodos balanceados, TIA – amplificador de trans-impedância, LPF – Filtro passa-baixo, ADC – conversor analógico-digital, DSP – processamento digital de sinal.	18
Figura 2.10: Modelo de um fotodíodo balanceado.	20
Figura 2.11: Exemplo do modelo de um recetor ótico coerente com diversidade de polarização, adaptado de [39, 40]: PBS – divisor de polarização, BPD – fotodíodos balanceados, TIA – amplificador de trans-impedância, LPF – Filtro passa-baixo, ADC – conversor analógico-digital, DSP – processamento digital de sinal.	21
Figura 2.12: Ilustração do processamento digital de sinal realizado num recetor coerente, adaptado de [18, 22].	21
Figura 2.13: Exemplo ilustrativo da técnica de compensação FDE, retirado de [42].	24
Figura 2.14: Exemplo ilustrativo do modelo de um filtro adaptativo, adaptado de [44]. ...	25
Figura 2.15: Modelo ilustrativo de um filtro MIMO em estrutura <i>butterfly</i> , [44, 45].	26
Figura 2.16: Modelo representativo do algoritmo V&V.	28
Figura 2.17: Exemplo representativo do algoritmo V&V [7].	29
Figura 3.1: Organização das diretorias do simulador OSIP.	32
Figura 3.2: Conteúdo da diretoria <i>modules</i>	32
Figura 3.3: Ilustração do ambiente de simulação do OSIP: a) – categorias de componentes; b) – pré-visualização do componente; c) – componentes da categoria selecionada; d) – barra de menus; e) – área de desenho.	34
Figura 3.4: Estrutura da função de execução da simulação.	34
Figura 3.5: Exemplo de uma janela de configuração da fibra linear.	35
Figura 3.6: Modelo de um componente no OSIP.	36
Figura 3.7: Exemplo ilustrativo do modulador IQ.	36
Figura 3.8: Organização das portas do componente no ambiente de simulação.	36

Figura 3.9: Janela de configuração das portas de entrada/saída de um componente.	37
Figura 3.10: Exemplo da estrutura das funções descritivas dos componentes.	37
Figura 3.11: Ilustração do primeiro passo do método de adicionar graficamente novos componentes.	38
Figura 3.12: Ilustração do primeiro passo do método de agrupar graficamente os componentes.	38
Figura 3.13: Função do <i>kernel</i> do simulador que define as constantes físicas da simulação.	39
Figura 3.14: Janela de configuração dos parâmetros de configuração da simulação.	39
Figura 3.15: Janela de impressão dos esquemas presentes na área de desenho: a) – zona de pré-visualização da área de desenho.	40
Figura 3.16: Comparação dos parâmetros de configuração das duas versões do OSIP: a) - versão 3.1; b) - versão 2012.	41
Figura 3.17: Diagrama do algoritmo de controlo do método de simulação.	42
Figura 3.18: Exemplo ilustrativo da passagem de parâmetros entre componentes: id – identificador do componente.	45
Figura 3.19: Ilustração do algoritmo de controlo para cada parâmetro de configuração gráfico.	46
Figura 4.1: Modelo do transmissor ótico implementado no OSIP para as simulações com ruído.	47
Figura 4.2: Modelo do recetor ótico implementado no OSIP para as simulações com ruído.	47
Figura 4.3: Relação entre o EVM e a SNR obtida variando a SNR à entrada do circuito de decisão.	48
Figura 4.4: Relação entre BER e SNR obtida variando o valor da SNR à entrada do circuito de decisão.	49
Figura 4.5: Modelo do transmissor ótico implementado no OSIP para estudar o desempenho dos algoritmos de compensação da dispersão cromática.	50
Figura 4.6: Modelo do recetor ótico implementado no OSIP para estudar o desempenho dos algoritmos de compensação da dispersão cromática.	51
Figura 4.7: Resultados da compensação da dispersão cromática no domínio do tempo para 500km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.	51
Figura 4.8: Resultados da compensação da dispersão cromática no domínio do tempo para 1000km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.	52
Figura 4.9: Resultados da compensação da dispersão cromática no domínio da frequência para 500km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.	53
Figura 4.10: Resultados da compensação da dispersão cromática no domínio da frequência para 1000km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.	54
Figura 4.11: Desempenho dos algoritmos de compensação da dispersão cromática, TDE e FDE, com o ruído.	55
Figura 4.12: Modelo de implementação do algoritmo CMA.	56
Figura 4.13: Modelo do transmissor ótico com multiplexagem na polarização implementado no OSIP.	57
Figura 4.14: Modelo em OSIP do canal ótico para simular os efeitos da PMD.	57

Figura 4.15: Modelo do recetor ótico com diversidade de polarização implementado no OSIP.	57
Figura 4.16: Constelações obtidas para a polarização x com uma DGD de 1ps e PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.	58
Figura 4.17: Constelações obtidas para a polarização y com uma DGD de 1ps e PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.	59
Figura 4.18: Constelações obtidas para a polarização x com uma DGD de 1ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.	59
Figura 4.19: Constelações obtidas para a polarização y com uma DGD de 1ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.	60
Figura 4.20: Constelações obtidas para a polarização x com uma DGD de 3.2ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.	60
Figura 4.21: Constelações obtidas para a polarização y com uma DGD de 3.2ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.	61
Figura 4.22: Desempenho do algoritmo CMA com o ruído para diferentes números de coeficientes dos filtros.	62
Figura 4.23: Modelos de implementação dos algoritmos de correção de fase: a) - V&V; b) - V&V com realimentação.	62
Figura 4.24: Modelo no OSIP do transmissor para teste dos algoritmos de correção de fase.	63
Figura 4.25: Modelo no OSIP do recetor para teste dos algoritmos de correção de fase.	63
Figura 4.26: Resultados obtidos para os dois circuitos com rotações de fase: a) – 30°; b) - -30°; c) – 45°; d) – 60°.	65
Figura 4.27: Resultados obtidos com o algoritmo com realimentação para uma variação temporal da rotação de fase: a) – sobreposição temporal do sinal recebido na DSP; b) – sinal compensado.	66
Figura 4.28: Modelo dos algoritmos de fase para testar o seu desempenho ao ruído do canal e de fase dos lasers: a) – V&V; b) – V&V com realimentação.	67
Figura 4.29: Relação entre a BER e a SNR teórica e para os algoritmos V&V e V&V com realimentação e processamento paralelo de 16 amostras.	67
Figura 4.30: Resultados obtidos para os dois algoritmos com uma SNR de 10dB, processamento paralelo de 16 amostras e variando a largura de linha do laser: a) - erro de fase; b) - BER.	68
Figura 4.31: Modelo de implementação dos algoritmos V&V e V&V com realimentação considerando o processamento de 256 amostras em paralelo.	69
Figura 4.32: Relação entre a BER e a SNR teórica e para os algoritmos V&V e V&V com realimentação e processamento paralelo de 256 amostras.	69

Figura 4.33: Resultados obtidos para os dois algoritmos com uma SNR de 10dB, processamento paralelo de 256 amostras e variando a largura de linha do laser: a) - erro de fase; b) - BER.....	70
Figura 4.34: Largura de linha tolerada pelos algoritmos de estimação de fase para diferentes valores de paralelismo.....	70
Figura 4.35: Modelo do transmissor WDM com dois canais SP-QPSK implementado no OSIP.....	72
Figura 4.36: Exemplo ilustrativo do canal de transmissão considerado para a simulação do sistema WDM 2xSP-QPSK.	72
Figura 4.37: Modelo do recetor para dois canais SP-QPSK implementado no OSIP.....	72
Figura 4.38: Ilustração do espectro do sinal após a multiplexagem dos dois canais SP-QPSK.	73
Figura 4.39: Relações entre a BER e a SNR para o sistema WDM 2xSP-QPSK.....	74
Figura 4.40: Modelo do transmissor implementado no OSIP para um canal PDM-QPSK.	75
Figura 4.41: Exemplo ilustrativo da multiplexagem e desmultiplexagem dos canais WDM e do canal de transmissão.	75
Figura 4.42: Modelo do recetor ótico implementado no OSIP para um canal PDM-QPSK.	75
Figura 4.43: Ilustração do espectro do sinal após a multiplexagem dos dois canais PDM-QPSK.	76
Figura 4.44: Relações entre a BER e a SNR para o sistema WDM 2xPDM-QPSK.	77
Figura 4.45: Modelo do transmissor com multiplexagem de polarização implementado no OSIP para o sistema PDM-QPSK.....	78
Figura 4.46: Modelo do canal de transmissão para simular a CD e a PMD.	78
Figura 4.47: Modelo do recetor com diversidade de polarização implementado no OSIP para o sistema DP-QPSK.....	79
Figura 4.48: Equalização do sinal passo-a-passo: a) – sinal de entrada do equalizador; b) – sinal após a compensação da CD; c) – sinal após a compensação da PMD; d) – sinal após a correção de fase.....	80
Figura 4.49: Resultados da BER obtidos para o sistema PDM-QPSK.	80
Figura A.1: Exemplo ilustrativo da constelação do formato de modulação BPSK.....	91
Figura A.2: Exemplo ilustrativo da constelação do formato de modulação QPSK.	92
Figura A.3: Exemplo ilustrativo da constelação do formato de modulação 16QAM.	94
Figura A.4: Exemplo ilustrativo da constelação do formato de modulação 32QAM.	95
Figura A.5: Exemplo ilustrativo da constelação do formato de modulação 64QAM.	96
Figura A.6: Curvas ilustrativas da probabilidade de erro de símbolo em função da SNR para os formatos de modulação BPSK, QPSK, 16QAM, 32QAM e 64QAM.....	97
Figura A.7: Curvas ilustrativas da probabilidade de erro de bit em função da SNR para BPSK, QPSK, 16QAM, 32QAM e 64QAM.	97
Figura B.1: Parâmetros de configuração da ADC no OSIP.	99
Figura B.2: Esquema de funcionamento da ADC no OISP.....	99
Figura B.3: Ícone da ADC no OSIP.....	100
Figura B.4: Parâmetros de configuração do componente Amplifier no OSIP.	100
Figura B.5: Ícone do Amplifier no OSIP.	100
Figura B.6: Parâmetros de configuração do componente Coding no OSIP.....	101

Figura B.7: Diagrama de funcionamento do componente Coding.....	101
Figura B.8: Mapeamento da sequência binária na constelação: (a) BPSK; (b) QPSK; (c) 16QAM; (d) 32QAM; (e) 64QAM.	102
Figura B.9: Ícone do componente Coding no OSIP.	102
Figura B.10: Parâmetros de configuração do componente Decoding no OSIP.	103
Figura B.11: Diagrama de funcionamento do componente Decoding.....	103
Figura B.12: Ícone do componente Decoding no OSIP.....	103
Figura B.13: Parâmetros de configuração do componente Downsampling.	104
Figura B.14: Ícone do componente Downsampling no OSIP.	104
Figura B.15: Parâmetros de configuração do componente ReSample.	104
Figura B.16: Ícone do componente ReSample no OSIP.	104
Figura B.17: Parâmetros de configuração do componente DSNR.....	105
Figura B.18: Ícone do componente DSNR no OSIP.....	105
Figura B.19: Parâmetros de configuração do componente DSP.....	105
Figura B.20: Modelo de implementação dos algoritmos para o DSP: (a) – para uma polarização; (b) – para duas polarizações; id - identificador.	106
Figura B.21: Ícone do componente DSP no OSIP.	106
Figura B.22: Ícone do componente Inverter no OSIP.....	107
Figura B.23: Exemplo da conversão para série de um sinal binário em sequências paralelas de 2 bits.....	107
Figura B.24: Parâmetros de configuração do componente PtoS.	107
Figura B.25: Ícone do componente PtoS no OSIP.	107
Figura B.26: Exemplo da paralelização de um sinal binário em sequências de 2 bits... ..	108
Figura B.27: Parâmetros de configuração do componente StoP.	108
Figura B.28: Ícone do componente StoP no OSIP.	108
Figura B.29: Ícone do componente TIA no OSIP.	108
Figura B.30: Diagrama de funcionamento do componente PolAlign.....	109
Figura B.31: Ícone do componente PolAlign no OSIP.....	109
Figura B.32: Modelo de implementação do modulador IQ.....	110
Figura B.33: Parâmetros de configuração do componente IQ_Mod.	110
Figura B.34: Ícone do componente IQ_Mod no OSIP.	110
Figura B.35: Parâmetro de configuração do componente BPD.	111
Figura B.36: Ícone do componente BPD no OSIP.	111
Figura B.37: Parâmetro de configuração do componente DGD.....	111
Figura B.38: Ícone do componente DGD no OSIP.....	112
Figura B.39: Parâmetros de configuração do componente DOSNR.....	112
Figura B.40: Ícone do componente DOSNR no OSIP.....	112
Figura B.41: Ícone do componente Optical90Hybrid no OSIP.	113
Figura B.42: Ícone do componente PBC_S no OSIP: (a) - PBC; (b) - PBS.	113
Figura B.43: Parâmetros de configuração do componente PolarizationRot.....	113
Figura B.44: Ícone do componente PolarizationRot no OSIP.	114
Figura B.45: Ícone do componente Polarizer no OSIP.....	114
Figura B.46: Parâmetros de configuração do componente WDM_Mux.	115
Figura B.47: Princípio de funcionamento do algoritmo do componente WDM_Mux.	115
Figura B.48: Ícone do componente WDM_Mux no OSIP.	115
Figura B.49: Parâmetros de configuração do componente WDM_Demux.....	116

Figura B.50: Diagrama de funcionamento do algoritmo do componente WDM_Demux.	116
Figura B.51: Ícone do componente WDM_Demux no OSIP.	116
Figura B.52: Parâmetros de configuração do componente PDL.....	117
Figura B.53: Ícone do componente PDL no OSIP.	117
Figura B.54: Parâmetro de configuração do componente PhaseRot.....	117
Figura B.55: Ícone do componente PhaseRot no OSIP.....	117
Figura B.56: Parâmetro de configuração do componente BERT.....	118
Figura B.57: Ícone do componente BERT no OSIP.....	118
Figura B.58: Parâmetro de configuração do componente EVM.....	118
Figura B.59: Ícone do componente EVM no OSIP.....	119
Figura B.60: Parâmetros de configuração do componente SavetoFile.....	119
Figura B.61: Ícone do componente SavetoFile no OSIP.	120
Figura B.62: Parâmetros de configuração do componente UConstellation.	120
Figura B.63: Ícone do componente UConstellation no OSIP.	120
Figura B.64: Modelo do transmissor adotado para o componente Transmitter.	121
Figura B.65: Parâmetros de configuração do componente Transmitter.	121
Figura B.66: Ícone do componente Transmitter no OSIP.	122
Figura B.67: Modelo do recetor adotado para o componente Receiver.	122
Figura B.68: Parâmetros de configuração do componente Receiver.	122
Figura B.69: Ícone do componente Receiver no OSIP.	123
Figura C.1: Ilustração do primeiro passo de configuração da interface gráfica.	125
Figura C.2: Ilustração da janela de desenho da interface gráfica com o GUIDE: a) – barra de componentes; b) – área de desenho; c) – barra de ferramentas.	125
Figura C.3: Esboço da interface a criar.....	126
Figura C.4: Esboço da interface pretendida no editor do GUIDE.	126
Figura C.5: Exemplo ilustrativo da janela de configuração de cada objeto introduzido.	127
Figura C.6: Exemplo ilustrativo do cabeçalho do ficheiro de código gerado para a configuração da interface gráfica.	127
Figura C.7: Exemplo do código de configuração da leitura dos valores de cálculo.	128
Figura C.8: Exemplo ilustrativo do código de configuração da operação de soma na calculadora.	128
Figura C.9: Exemplo ilustrativo do código de configuração do botão de cálculo.	129
Figura C.10: Exemplo ilustrativo da função responsável pela leitura, cálculo e atualização dos parâmetros da interface gráfica.	129
Figura C.11: Janela de interface com o utilizador resultante das configurações definidas.	130

Índice de tabelas

Tabela 3.1: Conteúdo da variável global NUM para as versões do OSIP: a) - versão 3.1; b) - versão 2012.....	43
Tabela 3.2: Lista dos componentes adicionados e suas funcionalidades na nova versão do OSIP.	44
Tabela 3.3: Componentes alterados na nova versão do OSIP.....	45
Tabela 3.4: Funções alteradas na nova versão do simulador.	45
Tabela 4.1: Configurações do transmissor e do recetor.	51
Tabela 4.2: Parâmetros de configuração do transmissor e do recetor para testar o desempenho dos algoritmos TDE e FDE.	55
Tabela 4.3: Parâmetros de configuração do transmissor e recetor para testar o algoritmo de compensação da PMD.	58
Tabela 4.4: Parâmetros de configuração do transmissor e do recetor para o teste dos algoritmos de correção de fase.	64
Tabela 4.5: Configurações do transmissor e do recetor para a simulação do sistema WDM 2xSP-QPSK.	73
Tabela 4.6: Configurações do transmissor e do recetor para a simulação do sistema WDM com dois canais DP-QPSK.....	76
Tabela 4.7: Configurações dos componentes do transmissor e do recetor para o sistema PDM-QPSK.....	79

1 Introdução

1.1 Perspetiva histórica das Comunicações óticas

O aumento do número de serviços de telecomunicações disponíveis aos utilizadores e a sua grande aceitação tem vindo a exigir, ao longo dos anos, o aumento da capacidade das redes de telecomunicações. Os sistemas de transmissão óticos representam um papel fundamental nesta evolução das redes, tendo sofrido ao longo dos anos várias atualizações que visam o aumento da sua eficiência. Desta forma podem ser identificadas várias gerações que se distinguem pelas especificações exigidas e pelos componentes e/ou tecnologias disponíveis [1]. A primeira geração de sistemas óticos foi estudada a partir de 1970 e utilizava um laser de Arsenieto de Gálio (GaAs) que operava na faixa dos $0.8\mu\text{m}$, permitindo a transmissão de 45 Mb/s num único canal, com um espaçamento entre repetidores de 10 km [2-4]. Os sinais eram modulados em intensidade, devido à elevada largura espectral dos lasers [5], e transportados em fibras multimodo. Os sistemas de deteção eram bastante simples, recorrendo à deteção direta. Os primeiros exemplos foram implementados em *Los Angeles* e em *Chicago*, sendo posteriormente demonstrados sistemas similares na *Europa* e no *Japão* [6]. Estes sistemas já apresentavam um enorme espaçamento entre repetidores relativamente aos sistemas com cabo coaxial, aproximadamente 1 km. Ainda durante a década de 1970 foi possível desenvolver recetores e fontes que operam a comprimentos de onda próximos de $1.3\mu\text{m}$, permitindo reduzir a atenuação, passando a ser próxima de 1dB/km, e assim aumentar a distância entre repetidores [2, 6]. Para este valor a fibra ótica apresenta o mínimo da dispersão cromática e desta forma tornava-se na zona de emissão mais desejada na altura. A segunda geração ficou disponível a partir de 1980 e corresponde à introdução de fibras monomodo, estas permitem apenas a propagação de um único modo, anulando o efeito da dispersão intermodal [1]. Foi assim possível atingir ritmos de transmissão superiores a 100Mbit/s [2]. Estes novos sistemas foram disponibilizados comercialmente em 1987 com taxas de transmissão de 1.7Gbit/s e espaçamento entre repetidores de 50km [2]. A principal limitação residia na atenuação, que tipicamente tomava valores de 0.5dB/km [3]. Para colmatar este problema surgia a necessidade de alterar o comprimento de onda dos lasers para $1.55\mu\text{m}$, ao qual corresponde o mínimo de atenuação [6]. Uma das adversidades para este comprimento de onda é a elevada dispersão registada. Do seu estudo surge a terceira geração dos sistemas óticos, que recorre a lasers com um comprimento de onda de $1.55\mu\text{m}$ e apresenta uma atenuação de aproximadamente 0.2dB/km. A dispersão era reduzida recorrendo a fibras de dispersão deslocada, que eram projetadas de modo a apresentarem um mínimo de dispersão para os $1.55\mu\text{m}$ [2, 4]. Em 1985 foram realizadas as primeiras experiências onde se registaram ritmos de transmissão de 4Gbit/s com um espaçamento entre repetidores de 100km. Os sistemas desta geração foram disponibilizados comercialmente a partir de 1990 e eram capazes de atingir taxas de transmissão até 10 Gb/s com repetidores espaçados de

distâncias entre 60 a 70 km [2]. A grande revolução nos sistemas óticos surge com o desenvolvimento de lasers com uma largura espectral reduzida, permitindo a implementação de sistemas coerentes [5]. Técnicas de modulação como a modulação digital de fase (PSK), modulação digital de frequência (FSK), modulação digital de amplitude (ASK) e a modulação em fase e quadratura (QAM) são técnicas usadas em sistemas coerentes [7]. A utilização de sistemas coerentes permitiu aumentar consideravelmente a sensibilidade dos recetores óticos, comparativamente com os sistemas de intensidade modulada com deteção direta (IM/DD), contudo apresentam uma complexidade acrescida devido à necessidade de controlo dos osciladores locais usados na modulação e desmodulação [7]. Por outro lado, estes sistemas vieram aumentar a sensibilidade e seletividade dos recetores, permitindo realizar de forma mais eficiente a multiplexagem e demultiplexagem da informação por divisão do comprimento de onda (WDM). Os sistemas coerentes juntamente com a introdução dos amplificadores óticos, nomeadamente os de fibra dopada com Érbio, EDFA, caracterizam a quarta geração [2, 4]. Esta geração para além de permitir um grande aumento da capacidade da fibra possibilitou também o aumento da distância entre regeneradores. Um dos problemas relacionados com esta geração é penalidade verificada nos sistemas devido à dispersão cromática. Por outro lado também exigiu o estudo de métodos de compensação dos efeitos não-lineares, que passaram a representar uma grande importância devido ao número de canais presentes na fibra, logo a potência transportada ser elevada [4]. Com o aparecimento das fibras de compensação de dispersão (DCF) foi possível gerir a dispersão acumulada e desta forma viabilizar os sistemas WDM [1, 3]. Com estes avanços, os sistemas de transmissão ótica sofreram, num curto espaço de tempo, evoluções assinaláveis, tanto ao nível das taxas de transmissão como ao nível da distância de propagação. Em 1996, começaram a ser instalados os primeiros sistemas de transmissão transatlânticos a operar a taxas de transmissão de 5 Gb/s, com distâncias de 11.300km [2]. Também nesta altura foi demonstrado que era possível transmitir informação a 1Tbit/s por fibra [4]. Na quinta geração dos sistemas de comunicação ótica aumentou-se a gama de comprimentos de onda válidos para os sistemas WDM. Passou a utilizar-se a gama de comprimentos de onda de 1.30 μ m a 1.65 μ m, que juntamente com amplificação de *Raman* possibilitaram transmissões com distâncias e ritmos de transmissão elevados. Nas últimas décadas a evolução dos sistemas coerentes permitiu o desenvolvimento de técnicas de compensação digital e formatos de modulação avançados. Estes avanços em conjunto com o aperfeiçoamento dos filtros óticos e com a modulação nas componentes de polarização permitiram aumentar a capacidade e eficiência dos sistemas óticos. Em 2007 foram demonstrados sistemas com ritmos de transmissão da ordem dos 25Tbit/s e com uma eficiência espectral de 3.2 b/s/Hz [3].

1.2 Motivação

Nos últimos anos tem-se assistido a um grande aumento da largura de banda nas redes, que é resultado do aumento das necessidades dos utilizadores [8]. Serviços de voz e dados ou mesmo internet de alta velocidade exigem o uso de tecnologias cada vez mais sofisticadas. Como consequência, torna-se necessário desenvolver novos equipamentos e tecnologias [7, 8]. As soluções adotadas visam reduzir o custo

necessário por bit de informação e maximizar a distância de transmissão, sem que seja necessário o uso de regeneradores. A procura de maior capacidade e funcionalidade dos sistemas óticos, levou a um aumento substancial na complexidade dos sistemas de comunicação nas últimas décadas. A evolução da investigação pode ser representada pela Figura 1.1, resultado de uma pesquisa sobre as publicações realizadas nos últimos anos relacionadas com comunicações óticas na base de dados [9].



Figura 1.1: Evolução das publicações relacionadas com comunicações óticas nos últimos anos.

A Figura 1.1 apresenta uma forte evolução na investigação das comunicações óticas, registando-se em 2011 a publicação de aproximadamente 10.000 artigos. No decorrer deste ano o número de artigos publicados já excedem metade desse valor (aproximadamente 8000 artigos até de 12 de Junho de 2012). A investigação realizada visa a procura de componentes com baixas perdas, amplificadores com baixo ruído, o estudo de formatos de modulação multinível, técnicas de compensação digital e de novos tipos de fibras, que proporcionem menor atenuação e que reduzam o impacto das não linearidades, permitindo assim transportar potências mais elevadas [10]. Uma das consequências é o crescente aumento da complexidade dos novos dispositivos e técnicas de transmissão levando a que a análise e projeto dos sistemas se tornem cada vez mais difíceis. É assim importante utilizar ferramentas de análise e projeto assistidas por computador. A transmissão de sinais digitais por fibra ótica representam um bom exemplo da crescente complexidade dos sistemas de transmissão. Nas últimas décadas, o produto taxa-de-transmissão \times distância aumentou várias ordens da grandeza [2], atingindo capacidades agregadas por fibra ótica na ordem das dezenas de terabits por segundo. No entanto, à medida que a capacidade de transmissão nas fibras óticas aumenta, cada vez mais fenómenos têm influência relevante (ruído gerado pelos amplificadores óticos, não linearidade da fibra ótica e dos componentes opto-eletrónicos, dispersão cromática (CD), flutuações na polarização do feixe ótico, etc) e as técnicas de transmissão e de receção mais complexas, estas últimas baseadas em formatos de modulação digital híbrida (amplitude e fase) e deteção coerente [7]. A simulação é atualmente reconhecida como a ferramenta principal para análise e projeto dos sistemas de comunicações óticas. Vários simuladores foram desenvolvidos na área dos sistemas de comunicações óticas, da qual se destaca a ferramenta comercial VPI photonics[®] [11], com uma grande aceitação nos meios académicos devido ao seu baixo custo para ensino

e permitir a simulação de praticamente todos os sistemas óticos [12, 13]. No entanto, este simulador apresenta como desvantagem o facto do utilizador não poder consultar e alterar os modelos dos componentes de acordo com as suas necessidades. Torna-se assim num software de simulação que, apesar de ter muitas potencialidades, limita o seu âmbito a nível de ensino e de investigação. O desenvolvimento do simulador OSIP [14] surge na tentativa de colmatar essas necessidades. Este sendo um software livre e de fácil utilização tornou-se, nos últimos anos, numa ferramenta didática e para fins de investigação. Tem mostrado enormes potencialidades e apresentado alguma evolução ao longo dos anos. A sua fácil adaptação a novos cenários torna-o num software de simulação bastante promissor.

1.3 Objetivos

O objetivo central deste trabalho é melhorar o *software* de simulação de sistemas de comunicações óticas com base na ferramenta computacional Matlab[®]. Pretende-se estudar a plataforma atual e partindo desta desenvolver novos algoritmos de controlo visando a implementação de novos métodos de simulação, dos quais se destaca o método de *Monte-Carlo*. No processo de atualização pretende-se manter a compatibilidade com qualquer versão da ferramenta Matlab[®], isto é, a mesma compatibilidade que se registava na versão anterior. O objetivo principal é a atualização dos componentes existentes para permitir a simulação de sistemas coerentes, o que implica o desenvolvimento de novos modelos para o transmissor, o canal e o recetor e a sua integração no simulador. Também se pretende o estudo de algoritmos de compensação digital bem como a sua implementação. Por fim através da simulação de sistemas reais pretende-se o teste das funcionalidades e componentes adicionados bem como a análise do desempenho dos algoritmos de equalização digital.

1.4 Publicações do autor

Durante a elaboração do presente trabalho foi submetido um artigo na revista do Departamento de Eletrónica Telecomunicações e Informática (DETUA) que se intitula “Simulador de comunicações óticas: OSIP2012” [15], com o objetivo de dar a conhecer algumas das potencialidades da nova versão do OSIP. Também foi publicado um artigo na conferência SEONS (Symposium on Enabling Optical Networks and Sensors) sobre o desempenho dos algoritmos de processamento digital em recetores coerentes [16].

1.5 Estrutura da dissertação

O documento encontra-se dividido em cinco capítulos. No presente capítulo é introduzido o trabalho a realizar e a motivação do autor. No segundo capítulo são apresentados os vários módulos dos sistemas óticos coerentes: transmissor, canal de transmissão e recetor. São ilustrados vários modelos do transmissor e do recetor assim como também os conceitos associados à modulação, desmodulação e propagação na

fibra ótica. Também são estudados alguns algoritmos para compensação dos efeitos lineares que resultam da propagação nas fibras óticas bem como algoritmos de estimação e correção de fase. No capítulo 3 é apresentada a estrutura atual do simulador, a qual representa o seu método de operação. São também apresentadas as funcionalidades introduzidas, de que são exemplo os algoritmos de controlo, métodos de simulação e componentes. Por outro lado são identificadas as funções e componentes já existentes na versão anterior que foram atualizados. No capítulo 4 são validadas as funcionalidades introduzidas no simulador. É realizada a análise ao desempenho do recetor e implementados e testados os algoritmos de compensação digital. São também abordados alguns modelos de sistemas reais, PDM-QPSK (Modulação digital de fase quaternária com multiplexagem na polarização) e WDM com e sem diversidade de polarização visando a análise do seu comportamento e o estudo conjunto de todos os algoritmos de compensação implementados. Por fim, no capítulo 5, são apresentadas algumas conclusões retiradas durante a elaboração do trabalho e são identificados alguns dos tópicos que poderão ser realizados em trabalhos futuros.

2 Sistemas Óticos Coerentes

Os recetores coerentes são o culminar da evolução que os sistemas óticos têm sofrido ao longo dos tempos. Atualmente é o método de deteção mais avançado, permitindo converter para o domínio elétrico toda a informação do sinal no domínio ótico. A preservação da fase, da frequência, da amplitude e mesmo da polarização permite adotar formatos de modulação de elevada eficiência espectral. Esta é uma das grandes vantagens deste tipo de sistemas. Assim, uma sequência de N bits pode ser codificada num símbolo, sendo possível transmitir o sinal a um ritmo que é N vezes inferior ao ritmo binário. Esta gestão espectral possibilita o aumento do ritmo de transmissão dos sinais sem que seja necessária maior largura de banda. Por outro lado, com o uso de um laser local para realizar o batimento com o sinal recebido é possível projetar sistemas com elevada sensibilidade e seletividade. Estes fatores são bastante importantes para os sistemas atuais, permitindo por exemplo, a implementação de multiplexagem no comprimento de onda (WDM). Para além disso, possibilita a implementação de algoritmos de processamento digital de sinal (DSP) com o objetivo de compensar os efeitos lineares e não lineares sofridos ao longo da propagação no canal. Aliando esta capacidade de compensação às características referidas anteriormente é possível identificar duas novas vantagens: a redução da complexidade dos sistemas de transmissão e o aumento da distância de transmissão sem que seja necessário regeneradores. A redução da complexidade dos sistemas é um dos pontos-chave para o aumento da eficiência. Sendo a compensação realizada totalmente no domínio digital, no recetor, é possível remover as habituais fibras de compensação de dispersão (DCFs) e os componentes óticos de compensação da dispersão dos modos de polarização (PMD). A Figura 2.1 apresenta o modelo de um sistema ótico coerente constituído pelo conjunto transmissor, canal e recetor.

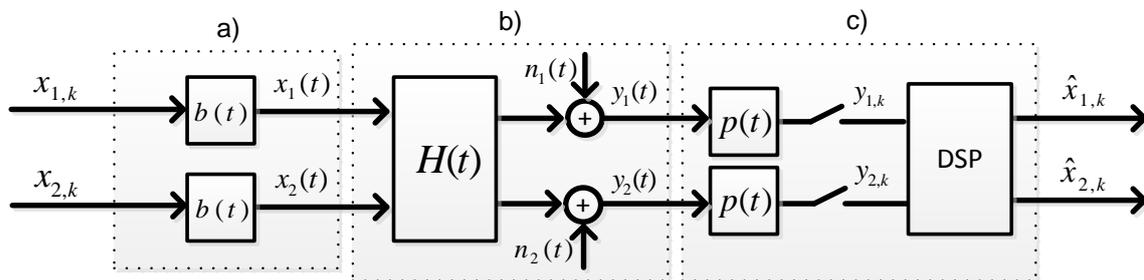


Figura 2.1: Exemplo de um modelo de um sistema de transmissão ótico, adaptado de [17]: a) Transmissor, b) Canal de transmissão, c) Recetor.

A Figura 2.1-a) representa um transmissor ótico com multiplexagem na polarização, que é responsável por receber a informação binária e codificá-la de acordo com o formato desejado. As variáveis $x_{1,k}$ e $x_{2,k}$ representam os sinais, de cada polarização, após a codificação. Estes são modulados no domínio ótico e enviados pelo canal de transmissão

Figura 2.1-b). O modelo do canal de transmissão representa os efeitos da CD, da PMD, da atenuação resultante da propagação e a adição de ruído. O sinal é adquirido no recetor e convertido para o domínio elétrico Figura 2.1-c). De seguida é amostrado e processado digitalmente, sendo a sequência binária descodificada ($\hat{x}_{1,k}$ e $\hat{x}_{2,k}$) [7, 17-21]. Os blocos que constituem o sistema de transmissão são estudados individualmente nos temas seguintes.

2.1 Transmissor

Como referido o transmissor pode ser visto como o conjunto de dois módulos, o primeiro responsável pelo mapeamento da informação binária e o segundo pela modulação ótica dos sinais. Ao longo do presente trabalho serão utilizados diferentes formatos de modulação, sendo o modelo do transmissor constante. A sua função de transferência é representada na Figura 2.1-a). Na secção 2.1.1 são apresentados os moduladores óticos e na 2.1.2 a arquitetura da componente digital do transmissor. Por fim na secção 2.1.3 são estudados os modelos de um transmissor ótico com e sem diversidade de polarização.

2.1.1 Componente ótica

Um dos componentes chave dos transmissores óticos é o modulador. Para melhor compreender o seu funcionamento é necessário introduzir primeiro os componentes fundamentais na modulação externa dos sinais: o modulador de fase (PM) e o modulador de Mach-Zehnder (MZM). Estes estão representados na Figura 2.2 e Figura 2.3 respetivamente, representando E_{in} e E_{out} os campos óticos à entrada e saída dos moduladores. Os moduladores em estudo baseiam-se na variação do índice de refração efetivo dos materiais com a tensão aplicada. Desta forma o sinal ótico recebido pode ser modulado em fase ou em amplitude, este último no caso do MZM [7, 22].

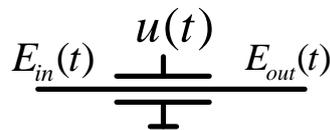


Figura 2.2: Modelo do modulador de fase, [7].

Dos moduladores considerados o mais simples é o PM. A tensão aplicada aos elétrodos $u(t)$ provoca a alteração do valor do índice de refração do guia de onda, variando assim o atraso de fase induzido e como tal da fase [7]:

$$\varphi_{PM} = \frac{u(t)}{v_\pi} \pi, \quad (2.1)$$

O valor v_π é a voltagem que é necessária aplicar aos terminais do elétrodo para que a rotação de fase resultante seja de 180° . Esta rotação provocada pela tensão aplicada traduz-se na relação entre o sinal de entrada e saída [7]:

$$E_{out}(t) = E_{in}(t)e^{j\varphi_{PM}} = E_{in}(t)e^{j\frac{u(t)}{v_\pi}\pi} \quad (2.2)$$

O MZM, Figura 2.3, é constituído por dois moduladores de fase em paralelo, cujas saídas são combinadas e interferem entre si. Funciona assim pelo princípio da interferência controlada pela modulação da fase da portadora ótica. A onda eletromagnética à entrada do modulador é dividida por dois caminhos distintos, nos quais é aplicada uma determinada tensão elétrica ($u_1(t), u_2(t)$), de modo a alterar o valor do índice de refração no guia de onda, variando o atraso de fase induzido em cada um dos braços do MZM. Na junção dos dois braços do modulador, a diferença de fase entre as duas ondas conduz a uma interferência construtiva ou destrutiva [7, 10].

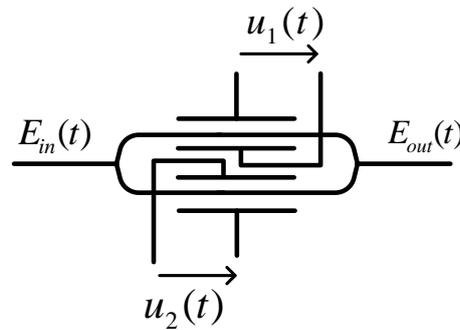


Figura 2.3: Modelo do modulador Mach-Zehnder, [7].

Desprezando as perdas por inserção, a sua função de transferência pode escrever-se [7, 10]:

$$E_{out}(t) = \frac{1}{2} E_{in}(t) (e^{j\varphi_1} + e^{j\varphi_2}) \quad (2.3)$$

Na equação (2.3) φ_1 e φ_2 representam o atraso de fase em cada um dos moduladores de fase e são dadas por [7, 10]:

$$\varphi_1 = \frac{u_1(t)}{v_\pi} \pi, \varphi_2 = \frac{u_2(t)}{v_\pi} \pi. \quad (2.4)$$

A combinação das tensões no MZM pode levar a dois casos particulares, que correspondem aos casos das tensões serem iguais, $u_1(t) = u_2(t) = u(t)$, ou simétricas, $u_1(t) = -u_2(t) = u(t)/2$. Caso sejam iguais o modulador comporta-se como um modulador de fase puro, sendo a função de transferência idêntica à (2.2). No caso de

serem simétricas, este opera no estado denominado na literatura de *push-pull* sendo a sua função de transferência [7, 10]:

$$\begin{aligned}
 E_{out}(t) &= \frac{1}{2} E_{in}(t) (e^{j\phi_1} + e^{-j\phi_1}) \\
 &= \frac{1}{2} E_{in}(t) \left(e^{j\frac{u}{2v_\pi}\pi} + e^{-j\frac{u}{2v_\pi}\pi} \right) \\
 &= E_{in}(t) \cos\left(\frac{u(t)}{2v_\pi}\pi\right).
 \end{aligned} \tag{2.5}$$

Neste caso o sinal é modulado em intensidade com uma característica sinusoidal. A transferência de potência pode escrever-se:

$$\begin{aligned}
 P_{out}(t) &= E_{out}(t) \cdot E_{out}^*(t) \\
 &= P_{in}(t) \cdot \cos^2\left(\frac{u(t)}{2v_\pi}\pi\right) \\
 &= P_{in}(t) \left[\frac{1}{2} + \frac{1}{2} \cos\left(\frac{u(t)}{v_\pi}\pi\right) \right].
 \end{aligned} \tag{2.6}$$

É visível na equação (2.6) que a potência transmitida corresponde à potência do sinal de entrada modulada e que a voltagem $u(t)$ foi escolhida de modo a que a tensão v_π provoque uma transição do máximo para o mínimo de transferência de potência, ou vice-versa. À tensão de modulação aplicada ao MZM pode estar também associada uma componente DC que definirá o ponto de funcionamento do modulador. A Figura 2.4 ilustra os comportamentos do modulador em potência e campo óticos.

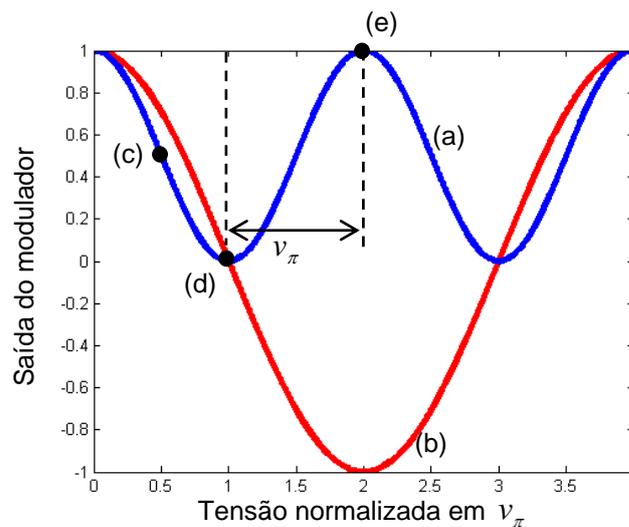


Figura 2.4: Ilustração das saídas do modulador MZM: (a) – potência ótica; (b) - campo ótico; (c) – ponto de quadratura; (d) – mínimo de transmissão; (e) – máximo de transmissão.

O ajuste do valor da polarização do modulador permite definir o ponto de funcionamento, por exemplo, no mínimo de transmissão ou no ponto de quadratura [7, 10].

Atendendo aos moduladores apresentados, um transmissor M-PSK pode ser implementado com qualquer um deles e um OOK com um MZM. Pretendendo-se modular os sinais com vários formatos, é necessário projetar um transmissor versátil. Considerando a combinação dos MZMs e PMs é possível obter vários modelos. A configuração frequentemente adotada recorre a dois MZMs e um PM, resultando o modulador IQ (IQM), Figura 2.5 [7, 22-24].

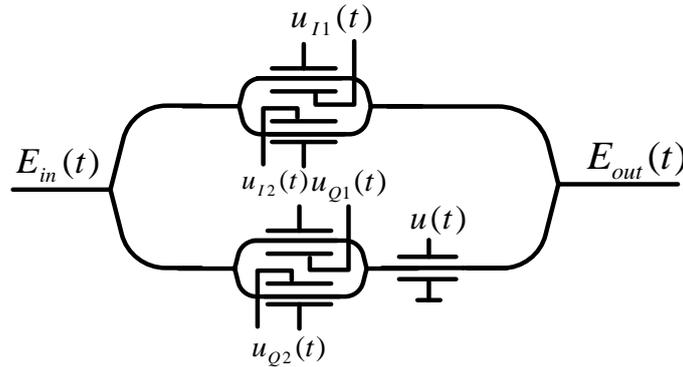


Figura 2.5: Esquema representativo do modulador IQ [22].

O sinal de luz incidente no modulador, $E_{in}(t)$, é dividido em duas partes de igual potência, sendo cada uma delas modulada por um MZM independente. Um dos ramos do MZM tem um PM cuja tensão de controlo é habitualmente especificada para uma rotação de -90° , portanto $u(t) = -\frac{V}{2}$. Assim um dos braços do modulador é responsável pela modulação da componente em fase e o outro da componente em quadratura. Após combinados os sinais dos dois braços do IQM, o sinal resultante, E_{out} , corresponde ao sinal incidente com a modulação de amplitude e fase definida pelas tensões dos MZMs [7, 25]. Nos cálculos admitiu-se que os sinais eram divididos e combinados com *couplers* de 3dB. A função transferência do IQM pode escrever-se:

$$\begin{aligned} \frac{E_{out}}{E_{in}} &= \frac{1}{2} \left(\frac{1}{2} \left(e^{j\frac{V_{I1}}{V_\pi}\pi} + e^{j\frac{V_{I2}}{V_\pi}\pi} \right) - \frac{1}{2} \left(e^{j\frac{V_{Q1}}{V_\pi}\pi} + e^{j\frac{V_{Q2}}{V_\pi}\pi} \right) e^{j\frac{u}{V_\pi}\pi} \right) \\ &= \frac{1}{4} \left(e^{j\frac{V_{I1}}{V_\pi}\pi} + e^{j\frac{V_{I2}}{V_\pi}\pi} \right) - \frac{1}{4} \left(e^{j\frac{V_{Q1}}{V_\pi}\pi} + e^{j\frac{V_{Q2}}{V_\pi}\pi} \right) e^{-j\frac{\pi}{2}}. \end{aligned} \quad (2.7)$$

Tendo em consideração o caso referido anteriormente, funcionamento em *push-pull*, considera-se as relações $V_I/2 = V_{I1} = -V_{I2}$ e $V_Q/2 = V_{Q1} = -V_{Q2}$, resultando:

$$\frac{E_{out}}{E_{in}} = \frac{1}{2} \left(\cos \left(\frac{V_I}{2V_\pi} \pi \right) + j \cos \left(\frac{V_Q}{2V_\pi} \pi \right) \right). \quad (2.8)$$

Pela equação (2.8) pode-se verificar que os dois ramos são modulados em intensidade, sendo um deles rodado de 90° antes da combinação dos dois sinais. Considerando agora as tensões de polarização de cada um dos MZMs pode escrever-se a função de transferência:

$$\frac{E_{out}}{E_{in}} = \frac{1}{2} \left(\cos \left(\frac{V_I + V_{DC,I}}{2V_\pi} \pi \right) + j \cos \left(\frac{V_Q + V_{DC,Q}}{2V_\pi} \pi \right) \right). \quad (2.9)$$

Os conceitos apresentados em cima aplicam-se à modulação do sinal no domínio ótico. Deve notar-se que as tensões de controlo do IQM resultam da codificação do sinal binário, que é realizada digitalmente.

2.1.2 Componente digital

A componente eletrónica do transmissor é responsável pelo mapeamento da informação binária e por gerar as tensões necessárias para o modulador, reduzindo a complexidade ótica. Atualmente, considerando a versatilidade dos equipamentos, é usual que se recorra a DSPs. O processamento é assim realizado no domínio digital e os sinais resultantes convertidos para o domínio analógico, utilizando para o efeito DACs [26, 27]. A arquitetura do DSP considerada encontra-se na Figura 2.6.

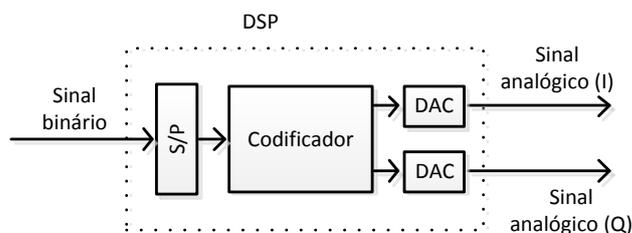


Figura 2.6: Exemplo da arquitetura do DSP para o mapeamento dos sinais: S/P - conversor série para paralelo, DAC – conversor digital para analógico.

O DSP deverá paralelizar a sequência binária, tendo em conta o número de bits a mapear por símbolo. Uma implementação simples poderá basear-se em linhas de atraso. De seguida deverá mapear essa informação e gerar os valores digitais, correspondentes aos valores analógicos necessários à modulação do sinal. Estes serão convertidos para o domínio analógico utilizando DACs. O mapeamento dos sinais pode ser realizado utilizando simplesmente LUTs (Look Up Table), um método rápido e eficiente [26]. Tendo em conta estes aspetos, para alterar o formato de modulação do transmissor, basta alterar dois parâmetros no DSP, o número de bits a paralelizar e as LUTs. A escolha do formato de modulação para um determinado cenário deverá ter em consideração a

dispersão acumulada e os efeitos da PMD. Os formatos multinível permitem obter um ritmo de transmissão de símbolos inferior ao da sequência de dados tornando-se vantajoso para os efeitos referidos e permitindo a implementação de processamento digital com menores velocidades de processamento. O ruído também representa um peso elevado na escolha da modulação a utilizar. Formatos de ordem mais elevada apresentam os símbolos mais próximos tornando os sistemas menos tolerantes a ruído e a outros tipos de distorções.

2.1.3 Modelo do transmissor ótico

O modelo de um transmissor ótico que representa a função transferência da Figura 2.1-a) é apresentado na Figura 2.7.

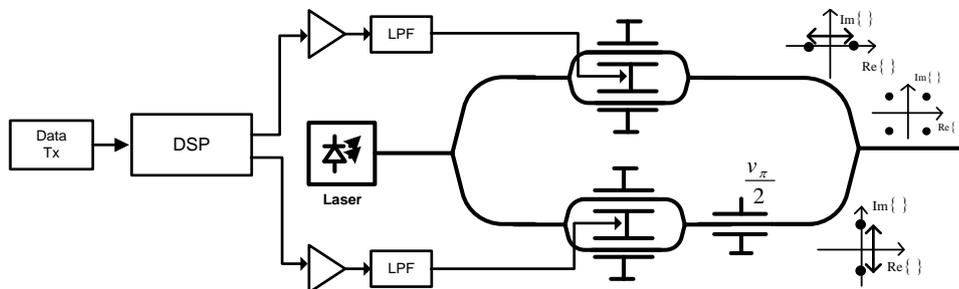


Figura 2.7: Ilustração do modelo de um transmissor ótico, adaptado de [27]: DSP – processamento digital de sinal, LPF – filtro passa-baixo.

A Figura 2.7 apresenta o gerador de informação binária, o DSP responsável pelo mapeamento e codificação do sinal e o IQM. A informação binária é adquirida no DSP e codificada, sendo gerados os valores analógicos para a modulação no IQM. Tendo em consideração que as DACs do DSP podem não ter a capacidade de fornecer a corrente de controlo ao modulador introduzem-se amplificadores. Por outro lado, é necessário limitar a largura de banda do sinal a modular devido às limitações do modulador ou mesmo do canal de transmissão, dependendo da aplicação, e reduzir o ruído na banda desejada. Desta forma para gerir a largura de banda ocupada pelo sinal a transmitir são utilizados filtros passa-baixo.

A Figura 2.8 representa um modelo de um transmissor ótico com multiplexagem na polarização. O feixe ótico é dividido em dois sinais de igual potência e modulado em dois transmissores independentes, um por cada polarização, semelhantes ao da Figura 2.7. Posteriormente os dois sinais são combinados com um PBC (Polarization beam combiner).

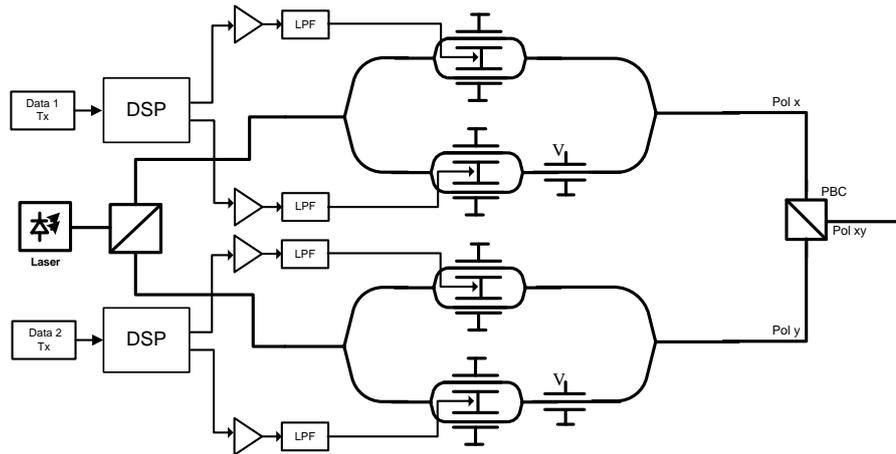


Figura 2.8: Ilustração do modelo de um transmissor ótico com diversidade de polarização, adaptado de [17, 27]: DSP – processamento digital de sinal, LPF – filtro passa-baixo, PBC – Combinador de polarizações.

A Figura 2.8 sugere que existem duas fontes binárias independentes, uma para cada polarização. Esta solução é frequentemente adotada em simulação. Na prática a fonte de informação pode ser a mesma, sendo desmultiplexada e direcionada para cada um dos moduladores. Um bom exemplo é admitir que os bits pares da sequência são transmitidos na polarização x e os ímpares na polarização y [27, 28].

2.2 Propagação na fibra

A propagação de um campo E num meio dispersivo segundo a direção z pode ser descrita pela equação não-linear de *Schroedinger* (NLSE) [10, 17]:

$$\frac{dE}{dz} + \frac{i}{2} \beta_2(z) \frac{d^2 E}{dt^2} - \frac{1}{6} \beta_3(z) \frac{d^3 E}{dt^3} + \frac{\alpha(z)}{2} E = i\gamma |E|^2 E, \quad (2.10)$$

Os parâmetros z e t representam as variáveis espacial e temporal respetivamente e α a atenuação do sinal na fibra. Este relaciona a potência do sinal de entrada, P_{in} , com a potência do sinal de saída, P_{out} , através da expressão [2, 6]:

$$P_{out} = P_{in} e^{-\alpha L}, \quad (2.11)$$

Sendo que, L representa o comprimento da fibra em km . Portanto a potência decai de forma exponencial de acordo com a constante α expressa em km^{-1} ou de acordo com a equação (2.12) em dB/km [2, 6].

$$\alpha(dB/km) \approx 4.343\alpha \quad (2.12)$$

Na equação (2.10) o termo β_2 representa o coeficiente de dispersão da velocidade de grupo (GVD) e relaciona-se com o parâmetro de dispersão D pela equação (2.13). O valor típico de D , para fibras SSMF a $1550nm$, ronda os $17 ps/nm.km$ [10]. O coeficiente β_3 refere-se à dispersão de segunda ordem, que não vai ser considerado no decorrer deste trabalho [10].

$$D = -\frac{2\pi c}{\lambda^2} \beta_2 \quad (2.13)$$

O segundo termo da equação (2.10) representa a dependência do índice de refração da potência, o seu peso depende do coeficiente não-linear γ cujas unidades são $W^{-1}km^{-1}$ [17]. Alguns dos efeitos associados a este termo são a modulação de fase resultante da potência do próprio sinal (SPM) e a modulação de fase provocada pelos sinais adjacentes (XPM) [2, 10, 17].

O canal de transmissão é representado na Figura 2.1-b), que para efeitos do presente trabalho é considerado linear. Desta forma a função de transferência $H(t)$ representa a dispersão cromática (CD), a dispersão dos modos de polarização (PMD) e a atenuação sofrida pelo sinal. Adicionalmente modela-se o ruído no canal através das variáveis $n_1(t)$ e $n_2(t)$. Os comportamentos da CD e da PMD serão estudados mais detalhadamente no subtópicos que se seguem.

2.2.1 Dispersão Cromática

A dispersão cromática é um efeito linear que ocorre na fibra ótica. A sua principal consequência é o alargamento temporal dos sinais, provocando interferência entre símbolos (ISI). As suas consequências são tão superiores quanto maior for a razão entre o alargamento dos impulsos e o período de símbolo. O alargamento observado é causado pela dependência da velocidade de grupo com a frequência das componentes do campo. Este fenómeno é tipicamente designado na literatura por GVD (dispersão da velocidade de grupo) [2, 18]. A sua dependência do material, dispersão material, e da geometria do guia, dispersão do guia de onda, permite dimensionar fibras que visam a gestão da dispersão e que são designadas de DCF [10]. O grande objetivo destas fibras é compensar a dispersão acumulada ao longo de uma ligação. O aumento da eficiência dos efeitos não-lineares e a maior atenuação são dois fatores limitativos do uso deste tipo de fibras. Atualmente já é possível evitar o uso deste tipo de fibras e compensar a CD digitalmente no recetor [29, 30].

Considerando o meio linear, o efeito da dispersão cromática no impulso $A(z,t)$ pode descrever-se pela equação diferencial [29]:

$$\frac{dA(z,t)}{dz} = j \frac{D\lambda^2}{4\pi c} \frac{d^2 A(z,t)}{dt^2}, \quad (2.14)$$

O parâmetro z é a distância de propagação, t é a variável de tempo, D é o coeficiente de dispersão, c a velocidade da luz e λ o comprimento de onda do feixe. A solução da equação (2.14) é exponencial, aplicando a transformada de *Fourier* para obter a solução no domínio da frequência obtém-se a função de transferência [10, 18, 29]:

$$G(z, \omega) = \exp\left(-j \frac{D\lambda^2 z}{4\pi c} \omega^2\right), \quad (2.15)$$

Sendo ω a frequência angular. Na equação (2.15) é possível constatar o efeito da componente de frequência no atraso referido anteriormente.

2.2.2 Dispersão dos Modos de Polarização

Da simetria circular da fibra é possível degenerar, de um modo transversal, dois modos de polarização com características semelhantes [2, 31]. Contudo, a existência de pequenas irregularidades resultantes do processo de fabrico, de natureza mecânica ou mesmo variações de temperatura causam assimetrias [22, 31]. Os dois modos de polarização resultantes, também designados na literatura por PSPs (estados principais de polarização), têm propriedades distintas [17, 32]. As diferentes velocidades de grupo resultantes causam um atraso entre as componentes de polarização (DGD) que tipicamente toma o valor de $0.1 \text{ ps}/\sqrt{\text{km}}$ e se manifesta como um comportamento dispersivo [32]. Este fenómeno resultante nas fibras designa-se por dispersão dos modos de polarização (PMD) e é uma das grandes limitações nos sistemas óticos. No caso particular do valor da DGD ser constante para um dado comprimento de onda diz-se que a PMD é de primeira ordem [31, 32]. Um dos grandes problemas associados à PMD é a sua natureza estocástica. O DGD e os PSPs podem variar na ordem dos milissegundos ou dias, sendo complexa a descrição matemática do fenómeno [31, 33, 34]. A forma mais simples de o descrever é dividir a fibra em secções e considerar que cada secção tem os PSPs constantes e que entre secções estes variam aleatoriamente [31]. Assim, desprezando as perdas dependentes da polarização, cada secção seria representada pela matriz de *Jones* (2.16) e a função de transferência da PMD pelo produto da função de transferência das várias secções (2.17) [35, 36].

$$J_i(\omega) = \begin{bmatrix} u_i(\omega) & v_i(\omega) \\ -v_i^*(\omega) & u_i^*(\omega) \end{bmatrix} \quad (2.16)$$

$$T(\omega) = \prod_{i=1}^M R_i D_i(\omega) \quad (2.17)$$

Na equação (2.17) M representa o número de secções consideradas, R_i o termo da DGD (2.18) e D_i o termo correspondente à rotação dos PSPs (2.19) [35].

$$R_i = \begin{bmatrix} \exp\left(j\omega \frac{\tau_{DGD}}{2}\right) & 0 \\ 0 & \exp\left(-j\omega \frac{\tau_{DGD}}{2}\right) \end{bmatrix} \quad (2.18)$$

$$D_i = \begin{bmatrix} \exp\left(j\frac{\varphi}{2}\right) & 0 \\ 0 & \exp\left(-j\frac{\varphi}{2}\right) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (2.19)$$

Nas equações (2.18) e (2.19) ω representa a frequência angular, τ_{DGD} o atraso de grupo entre as duas polarizações, φ o ângulo de rotação na vertical e θ o ângulo de rotação na horizontal. Os ângulos φ e θ estão uniformemente distribuídos entre $\varphi \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$ e $\theta \in [0; 2\pi]$, [35].

2.2.3 Efeitos não-lineares

A propagação em fibras monomodo (SMF) é essencialmente confinada ao núcleo, que normalmente tem áreas bastante reduzidas [2, 10]. Desta forma, a elevada intensidade de sinal provoca variações no índice de refração da fibra e consequentemente na fase do sinal ótico [7, 10]. Este fenómeno é não-linear e apresenta uma forte dependência da potência. As causas deste efeito podem ser agrupadas em duas categorias: as que resultam da propagação de um único canal e as que resultam da interação entre dois ou mais canais WDM [10, 22]. Apesar de estes coexistirem nos sistemas atuais, a sua influência depende do ritmo de símbolos do sinal. Para valores inferiores a 10Gbaud predominam as não-linearidades entre componentes do próprio canal e para valores superiores as resultantes da interação entre canais [10]. Para melhor compreender o fenómeno considera-se um sinal WDM com três componentes E_1 , E_2 e E_3 . Deve notar-se que estes podem representar diferentes canais ou as várias componentes de um único canal. As equações que traduzem a evolução das componentes de campo são idênticas, para o caso de E_1 pode escrever-se [10]:

$$\begin{aligned} \frac{dE_1}{dz} + \frac{i}{2}\beta_2(z)\frac{d^2E_1}{dt^2} - \frac{1}{6}\beta_3(z)\frac{d^3E_1}{dt^3} + \frac{\alpha(z)}{2}E_1 \\ = \underbrace{i\gamma|E_1|^2 E_1}_{\text{SPM}} + \underbrace{2i\gamma(|E_2|^2 + |E_3|^2)E_1}_{\text{(I)XPM}} + \underbrace{i\gamma E_2^2 E_3^*}_{\text{(I)FWM}} \end{aligned} \quad (2.20)$$

No segundo membro da equação (2.20) identificam-se três termos não-lineares, que correspondem aos efeitos SPM, XPM e FWM [10]. O SPM resulta da modelação de

fase não-linear autoinduzida, isto é, apenas depende da sua própria potência [2, 10, 17]. Por outro lado, o XPM resulta da interferência com os canais adjacentes, dependendo assim da potência destes [2, 10, 17]. Caso o sinal tenha apenas um canal este fenómeno pode resultar da interferência entre as várias componentes do sinal, sendo designado nesse caso de IXPM. Por fim, o último termo representa o FWM, que resulta do batimento entre os canais adjacentes, transferindo parte da sua energia para uma nova componente de frequência [2, 10]. Considerando que o sinal tem três portadoras, ω_1 , ω_2 e ω_3 , da ocorrência deste efeito resulta uma quarta que obedece à relação $\omega_4 = \omega_1 \pm \omega_2 \pm \omega_3$ [2]. De forma semelhante ao caso anterior, este pode também resultar da combinação das várias componentes de frequência de um único canal, sendo nesse caso designado de IFWM [10]. Atualmente já é possível compensar digitalmente os efeitos não-lineares recorrendo a series de *Volterra* [37] ou algoritmos de propagação inversa [38].

2.3 Recetor coerente

O recetor ótico, representado na Figura 2.1-c), deve ser capaz de recuperar a informação modulada em amplitude, fase, frequência e polarização [7]. De uma forma geral deve implementar a função inversa do transmissor. Os recetores coerentes digitais permitem também equalizar o canal de transmissão. A Figura 2.9 apresenta um modelo de um recetor ótico coerente.

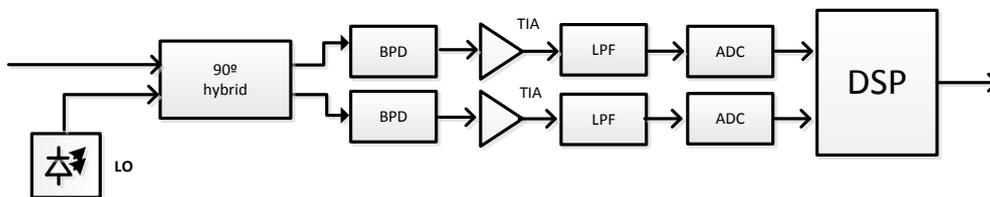


Figura 2.9: Exemplo do modelo de um recetor ótico coerente, adaptado de [39, 40]: BPD – fotodíodos balanceados, TIA – amplificador de trans-impedância, LPF – Filtro passa-baixo, ADC – conversor analógico-digital, DSP – processamento digital de sinal.

O sinal é adquirido no recetor e combinado com o laser local, princípio semelhante aos recetores de radiofrequência. A principal diferença relativamente aos sistemas de rádio é a natureza dos sinais, que neste caso encontram-se no domínio ótico. O sinal resultante é detetado por fotodíodos balanceados e assim é possível representar o sinal no domínio elétrico nas suas componentes de fase e quadratura. Os fotodíodos balanceados permitem eliminar o ruído de intensidade do laser local e detetar toda a sua potência. Tendo em conta que as ADCs adquirem o sinal em tensão é necessário um conversor corrente-tensão. Os níveis de excursão do sinal podem ser bastante reduzidos, pelo que, para que estes estejam acima do limiar de deteção das ADCs pode ser necessário amplificação. Assim, são utilizados amplificadores de trans-impedância que possibilitam a conversão do sinal e a sua amplificação. Os filtros passa-baixo são bastante importantes para que da operação de amostragem não resulte sobreposições espectrais. Estes filtros também limitam a quantidade de ruído e incluem resposta em frequência dos fotodíodos. É frequente que a resposta dos fotodíodos seja adequada

para limitar a largura de banda tanto do sinal como do ruído, situação em que os filtros passa-baixo não têm um correspondente físico, isto é, não precisam de ser implementados. É utilizado um DSP para igualar o canal de transmissão e decodificar o sinal. Para melhor compreender o recetor é necessário estudar os seus componentes principais. O primeiro componente a estudar é o 2×4 90° híbrido. Este permite detetar as componentes em fase e quadratura do sinal através do batimento entre o laser local e o sinal recebido. Considerando que os sinais nas suas entradas são descritos por (2.21), a potência desejada à saída de cada uma das portas ($n=0,1,2,3$) do componente é descrita na equação (2.22) [7].

$$E_{in1} = |E_{in1}|e^{j\phi_1}, E_{in2} = |E_{in2}|e^{j\phi_2} \quad (2.21)$$

$$P_{out,n}(t) = E_{out,n}(t) \cdot E_{out,n}^*(t) = \frac{1}{4}|E_{in1}|^2 + \frac{1}{4}|E_{in2}|^2 + \frac{1}{2}|E_{in1}||E_{in2}|\cos[\phi_1(t) - \phi_2(t) - n \cdot 90^\circ + \psi]; n = 0,1,2,3 \quad (2.22)$$

Na equação (2.21) $|E_{in1}|$ e $|E_{in2}|$ representam a informação de intensidade dos sinais e ϕ_1 e ϕ_2 a informação de fase. Já na equação (2.22) ψ representa uma rotação arbitrária de fase e $P_{out,n}$ a potência em cada porta de saída. Para garantir as relações de potência necessárias a função de transferência do componente assume a forma [7]:

$$\begin{bmatrix} E_{out1} \\ E_{out2} \\ E_{out3} \\ E_{out4} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} e^{j\psi_{11}} & e^{j\psi_{12}} \\ e^{j\psi_{21}} & je^{j\psi_{22}} \\ e^{j\psi_{31}} & -e^{j\psi_{32}} \\ e^{j\psi_{41}} & -je^{j\psi_{42}} \end{bmatrix} \begin{bmatrix} E_{in1} \\ E_{in2} \end{bmatrix}, \quad (2.23)$$

Os parâmetros $E_{in,i}$ e $E_{out,i}$ representam as entradas e saídas do componente e $\psi_{j,i}$ os coeficientes de fase, que devem satisfazer a relação $\psi_{11} - \psi_{12} = \psi_{21} - \psi_{22} = \psi_{31} - \psi_{32} = \psi_{41} - \psi_{42} = \psi$. Várias implementações deste componente são possíveis [41], mas não serão estudadas no âmbito deste trabalho. A interface entre o meio ótico e elétrico é realizada por fotodíodos, cuja função de transferência é dada por [7]:

$$I_{out} = R \cdot E_{inc} \cdot E_{inc}^*, \quad (2.24)$$

sendo que, R representa a responsividade dos fotodíodos [A/W] e E_{inc} o feixe de luz incidente no fotodíodo. Um fotodíodo balanceado (BPD) é constituído por dois fotodíodos, Figura 2.10, como tal, a corrente de saída é dada pela subtração das correntes detetadas em cada um dos fotodíodos (2.25).

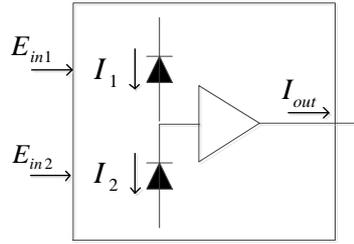


Figura 2.10: Modelo de um fotodiodo balanceado.

$$I_{out} = I_1 - I_2 = R(E_{in1} \cdot E_{in1}^* - E_{in2} \cdot E_{in2}^*). \quad (2.25)$$

Considerando o conjunto do 2x4 90° híbrido e dos BPD o sinal obtido no domínio elétrico para a componente de fase e quadratura é dado pelas equações (2.26) e (2.27), respetivamente [7], sendo que a característica de transferência do 2x4 90° híbrido considerada é representada pela equação (2.26) [7, 18].

$$\begin{bmatrix} E_{out1} \\ E_{out2} \\ E_{out3} \\ E_{out4} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} E_s + E_{lo} \\ E_s + jE_{lo} \\ E_s - E_{lo} \\ E_s - jE_{lo} \end{bmatrix} \quad (2.26)$$

$$\begin{aligned} I_I(t) &= R(E_{out1} \cdot E_{out1}^* - E_{out3} \cdot E_{out3}^*) \\ &= R\sqrt{P_s P_{lo}} a(t) e_s e_{lo} \cos[\Delta\omega t + \varphi_n(t) + \varphi_0 + \varphi(t)] + i_{shI} \end{aligned} \quad (2.27)$$

$$\begin{aligned} I_Q(t) &= R(E_{out2} \cdot E_{out2}^* - E_{out4} \cdot E_{out4}^*) \\ &= R\sqrt{P_s P_{lo}} a(t) e_s e_{lo} \sin[\Delta\omega t + \varphi_n(t) + \varphi_0 + \varphi(t)] + i_{shQ} \end{aligned} \quad (2.28)$$

Nas equações (2.26), (2.27) e (2.28) E_{lo} representa o campo do oscilador local, P_s a potência do sinal recebido, P_{lo} a potência do oscilador local, $a(t)$ a informação de intensidade, $\Delta\omega$ o desvio de frequência entre o laser local e o do transmissor, φ_n o ruído de fase, φ_0 a fase inicial, $\varphi(t)$ a informação de fase e $i_{shI/Q}$ o ruído quântico [7]. Considera-se que o ruído de intensidade do laser LO é completamente cancelado pelo balanceamento, e que o ruído de intensidade do laser emissor é desprezável. Das equações (2.27) e (2.28) pode verificar-se que o oscilador local permite amplificar o sinal recebido, contribuindo para uma melhor deteção do mesmo. Por outro lado, a seletividade do recetor é definida pelo oscilador local e pelo filtro passa-baixo. Após a amplificação e filtragem o sinal detetado é amostrado, normalmente a duas amostras por símbolo, e processado digitalmente.

A Figura 2.11 ilustra um modelo de um recetor ótico coerente com diversidade de polarização.

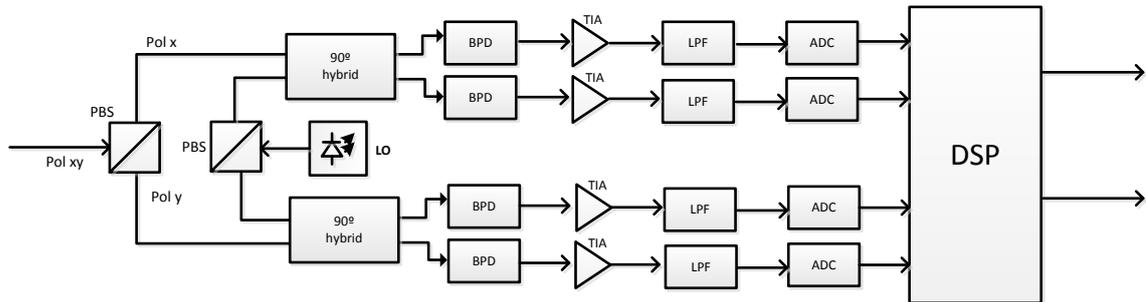


Figura 2.11: Exemplo do modelo de um recetor ótico coerente com diversidade de polarização, adaptado de [39, 40]: PBS – divisor de polarização, BPD – fotodiodos balanceados, TIA – amplificador de trans-impedância, LPF – Filtro passa-baixo, ADC – conversor analógico-digital, DSP – processamento digital de sinal.

O recetor é formado por dois recetores idênticos ao da Figura 2.9. O sinal é recebido num PBS, que o divide nas duas componentes de polarização. O mesmo acontece com o oscilador local, sendo dividido em dois sinais de igual potência. Após misturados com a respetiva componente do laser local e convertidos para o domínio elétrico, os sinais provenientes das duas polarizações são amostrados e processados simultaneamente no DSP, onde o canal de transmissão é igualado e a sequência binária descodificada. Tendo em vista a compensação do canal de transmissão, é usual utilizarem-se duas amostras por símbolo nas ADCs [40]. Considerando apenas os efeitos lineares, o DSP deverá ser capaz de igualar a CD, a PMD e corrigir a fase do sinal (ver Figura 2.12).

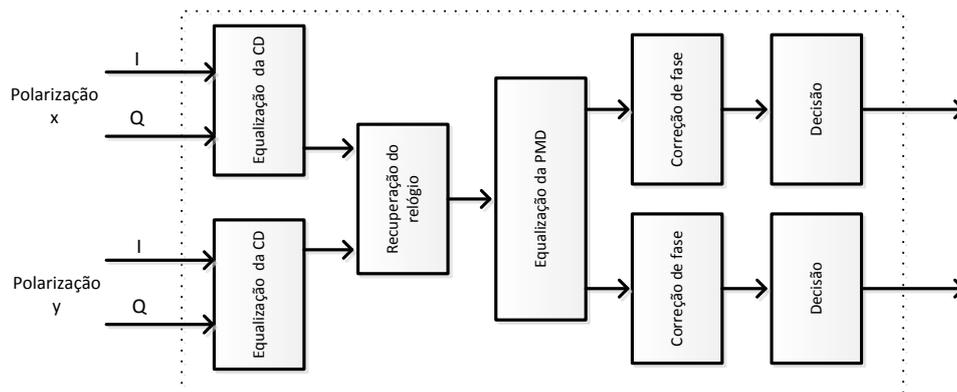


Figura 2.12: Ilustração do processamento digital de sinal realizado num recetor coerente, adaptado de [18, 22].

Após a compensação são decididos quais os símbolos recebidos e a sequência binária é descodificada.

2.4 Processamento Digital de Sinal

Neste tema são apresentadas algumas das soluções possíveis para a compensação dos principais fatores limitadores da eficiência dos sistemas óticos atuais. Existem dois métodos de compensação: o primeiro que permite estimar a função de transferência do canal de forma “cega”, isto é, apenas tem em conta algumas propriedades estatísticas do sinal que está a ser transmitido, e o segundo que estima a transferência do canal com base numa sequência de símbolos conhecida [30]. Neste último caso, é necessário o envio periódico de uma sequência de treino para estimar o canal. A periodicidade de envio da sequência de treino induz uma limitação no acompanhamento das variações do canal. No caso de as sequências estarem espaçadas de um intervalo temporal elevado é difícil acompanhar as alterações do canal. No caso contrário, se o espaçamento for pequeno, o sistema torna-se ineficiente [30]. As técnicas “cegas” apesar de serem mais simples não garantem a adaptação ao canal e portanto que a compensação se processa corretamente. Ainda assim estas técnicas são usualmente aplicadas. Os conceitos apresentados nos subtemas seguintes estão relacionados com técnicas “cegas” considerando a ausência de ruído e sistemas lineares [30].

2.4.1 Compensação da dispersão Cromática

Como referido na secção “2.2.1 Dispersão Cromática”, a resposta em frequência da CD é dada pela equação (2.15). Tendo em vista a sua equalização, dois métodos podem ser adotados: no domínio do tempo (TDE) ou no domínio da frequência (FDE) [36, 42]. A escolha do compensador depende do valor máximo admissível para a dispersão acumulada ao longo da propagação. Se for baixa, normalmente as abordagens são TDE, caso contrário os filtros de equalização tornam-se muito complexos, sendo as abordagens FDE mais simples [30].

No TDE procura-se dimensionar um filtro FIR que inverta a resposta impulsional da CD. O filtro necessário é projetado de acordo com o valor máximo da dispersão acumulada admissível no canal. Recorrendo à transformada de *Fourier* é possível converter a equação (2.15) para o domínio do tempo [29]:

$$g(z, t) = \sqrt{\frac{c}{jD\lambda^2 z}} \exp\left(j \frac{\pi c}{D\lambda^2 z} t^2\right). \quad (2.29)$$

Na equação (2.29), c representa a velocidade da luz, D o parâmetro de dispersão, z o comprimento da fibra, λ o comprimento de onda do sinal e t o parâmetro temporal. A resposta do compensador digital deverá ser [29]:

$$g_c(z, t) = \sqrt{\frac{jc}{D\lambda^2 z}} \exp(-j\phi(t)), \quad \phi(t) = \frac{\pi c}{D\lambda^2 z} t^2. \quad (2.30)$$

A resposta impulsiva do filtro (2.30) apresenta alguns problemas de implementação prática: resposta não é causal e não é limitada nos domínios do tempo e da frequência. A solução é truncar a resposta à frequência de *Nyquist*. Considerando que o período de amostragem é T , a frequência de *Nyquist* é $\omega_n = \frac{\pi}{T}$. Para não provocar *aliasing* a frequência angular da resposta não deverá exceder esse valor [29]:

$$\omega = \frac{d\phi(t)}{dt} = \frac{2\pi c}{D\lambda^2 z} t. \quad (2.31)$$

Da limitação da frequência angular resulta a limitação da resposta impulsiva [29]:

$$|t| \leq \frac{|D|\lambda^2 z}{2cT}. \quad (2.32)$$

A resposta impulsional é agora finita, sendo possível a implementação digitalmente. O filtro resultante poderá ser implementado com linhas de atraso e terá N coeficientes. O cálculo dos coeficientes terá de ter em conta a característica [29, 43]:

$$a_k = \sqrt{\frac{jcT^2}{D\lambda^2 z}} \exp\left(-j \frac{\pi cT^2}{D\lambda^2 z} k^2\right), -\left\lfloor \frac{N}{2} \right\rfloor \leq k \leq \left\lfloor \frac{N}{2} \right\rfloor, e \ N = 2 \times \left\lfloor \frac{|D|\lambda^2 z}{2cT^2} \right\rfloor + 1, \quad (2.33)$$

Onde $\lfloor x \rfloor$ representa o maior número inteiro não superior a x e N o número de coeficientes do filtro.

As abordagens FDE exigem a conversão do sinal recebido para o domínio da frequência, isto é, o cálculo da FFT. Este cálculo torna a implementação do algoritmo complexa, sendo necessárias técnicas de sobreposição [30, 42]. Assim, a sequência de símbolos recebida é dividida em blocos de N_c amostras sobre as quais é calculada a FFT. O sinal no domínio da frequência é depois multiplicado pelos coeficientes da função de transferência inversa da CD [42]. Considerando que a resolução da FFT é finita, a função de transferência aplicada corresponde à função inversa da equação (2.15) amostrada [43]. Visto que o sinal provém das ADCs, a frequência máxima representável é a frequência de *Nyquist* ω_n , logo os coeficientes do filtro FDE podem escrever-se [43]:

$$b_k = \exp\left(-j \frac{D\lambda^2 z}{\pi c} \left(\frac{k}{N_c} \omega_n\right)^2\right), -\frac{N_c}{2} \leq k \leq \frac{N_c}{2} - 1. \quad (2.34)$$

Aplicada a função inversa da FFT (IFFT) os primeiros e últimos N_e símbolos são removidos, permanecendo apenas $N_0 = N_c - 2N_e$. Os $2N_e$ símbolos adicionais permitem eliminar a interferência entre blocos e caracterizar a resposta impulsional da CD [42]. Os símbolos resultantes são recombinaados à medida que são processados. A saída do

igualador corresponde ao sinal recombinado. A Figura 2.13 exemplifica o funcionamento do algoritmo.

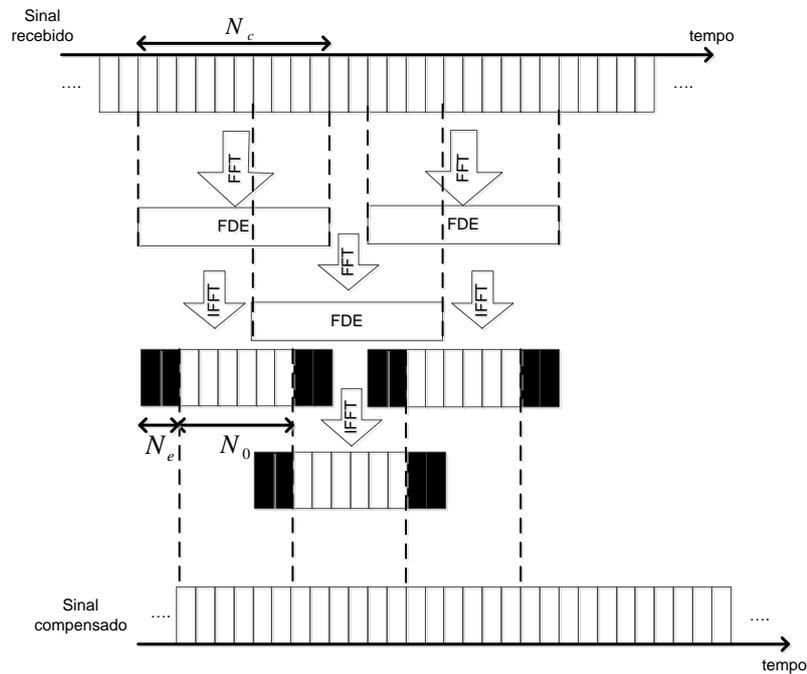


Figura 2.13: Exemplo ilustrativo da técnica de compensação FDE, retirado de [42].

O número de amostras a sobrepor é um fator importante no desempenho deste algoritmo e como tal deve ser corretamente determinado em função de cada cenário [42]:

$$N_e > F_s S(f_{\max}) = \frac{cLDf_{\max} F_s}{f_c^2}. \quad (2.35)$$

Na equação (2.35) c é a velocidade da luz, L o comprimento da fibra, D o parâmetro de dispersão, f_{\max} a máxima frequência do sinal, f_c a frequência central e $S(f)$ a função de transferência que traduz o atraso de grupo (2.36) [42].

$$S(f) = \frac{cLdf}{f_c^2} \quad (2.36)$$

Deve-se ter em atenção que a zona de guarda utilizada, isto é, o número de amostras sobrepostas não deve ser muito elevada pois limita o desempenho do compensador. Um aspeto fundamental a ter em conta na implementação dos igualadores é o número mínimo de coeficientes. Estes deverão ser superiores ao comprimento do canal em dispersão [36]:

$$N_{CD} = \left\lceil \frac{\tau_{CD}}{T} \right\rceil, \quad (2.37)$$

A operação $\lceil x \rceil$ representa o menor número inteiro não inferior a x , o parâmetro T representa o período de amostragem e τ_{CD} o comprimento do canal em dispersão, em segundos [36]:

$$\tau_{CD} = \frac{c}{f_c^2} |D| L \Delta f. \quad (2.38)$$

Na equação (2.38) Δf é a largura de banda do sinal.

2.4.2 Compensação da dispersão dos Modos de Polarização

O algoritmo de equalização da PMD deve estimar de forma eficiente os parâmetros que caracterizam as imperfeições do canal dependentes da polarização, equação (2.17). A estimação destes valores torna-se complexa devido à natureza estocástica da PMD [31]. Assim os métodos mais eficientes fazem uso de filtros adaptativos [31, 44]. Esta abordagem permite projetar filtros mais eficientes e com menor número de coeficientes [40]. É de realçar que estes também são capazes de compensar a CD. A Figura 2.14 apresenta o modelo de um filtro adaptativo.

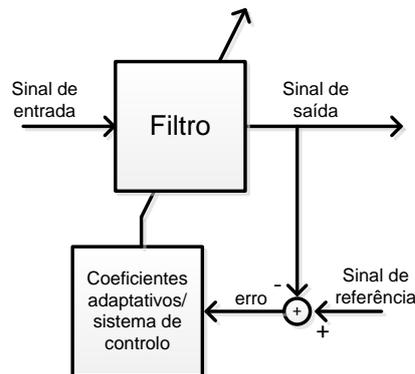


Figura 2.14: Exemplo ilustrativo do modelo de um filtro adaptativo, adaptado de [44].

De acordo com os sinais de saída e de referência é gerado um sinal de erro que permite ajustar os coeficientes do filtro. O filtro frequentemente utilizado é múltiplas entradas-múltiplas saídas (MIMO), que é designado na literatura como filtro *butterfly* [44, 45], Figura 2.15.

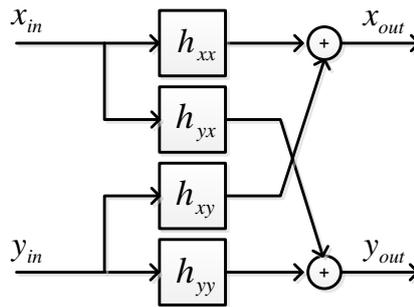


Figura 2.15: Modelo ilustrativo de um filtro MIMO em estrutura *butterfly*, [44, 45].

As saídas do filtro MIMO podem escrever-se [44, 45]:

$$\begin{aligned} x_{out} &= x_{in} * h_{xx} + y_{in} * h_{xy} \\ y_{out} &= x_{in} * h_{yx} + y_{in} * h_{yy} \end{aligned} \quad (2.39)$$

Sendo que h_{xx} , h_{xy} , h_{yx} e h_{yy} representam os coeficientes dos filtros, x_{in} e y_{in} os sinais de entrada, x_{out} e y_{out} as saídas do compensador e $*$ a operação de convolução. O ajuste dos coeficientes depende dos algoritmos de controlo utilizados. Destacam-se dois: o algoritmo CMA (*Constant Modulus Algorithm*) e o algoritmo LMS (*Least Mean Square*) [44].

O CMA é um algoritmo “cego” que faz uso do módulo constante do sinal para compensar a distorção e assim gerar o sinal de ajuste dos filtros [44]. O método foi inicialmente desenvolvido para formatos de modulação M-PSK dado que apresentam módulo unitário [46]. Apesar do seu funcionamento não ser tão eficiente pode ser adaptado para outros formatos de modulação, como apresenta *Irshaad Fatadin* em [46] para o caso do 16QAM. Considerando o formato QPSK e o filtro da Figura 2.15 o sinal de erro gerado é [29, 44]:

$$\begin{aligned} \mathcal{E}_{CMA,x} &= 1 - |x_{out}|^2 \\ \mathcal{E}_{CMA,y} &= 1 - |y_{out}|^2 \end{aligned} \quad (2.40)$$

A equação (2.40) calcula o erro do sinal relativamente ao módulo unitário forçando o sinal a convergir para esse valor. As condições de ajuste dos coeficientes podem escrever-se [29, 44]:

$$\begin{aligned} h_{xx} &= h_{xx} + \mu \mathcal{E}_{CMA,x} x_{out} x_{in}^* \\ h_{xy} &= h_{xy} + \mu \mathcal{E}_{CMA,x} x_{out} y_{in}^* \\ h_{yx} &= h_{yx} + \mu \mathcal{E}_{CMA,y} y_{out} x_{in}^* \\ h_{yy} &= h_{yy} + \mu \mathcal{E}_{CMA,y} y_{out} y_{in}^* \end{aligned} \quad (2.41)$$

Onde μ representa o parâmetro de convergência e $*$ o complexo conjugado. A estabilidade e convergência do algoritmo estão em grande parte dependentes deste parâmetro. Se o seu valor for muito pequeno o algoritmo demora muito tempo a convergir, caso seja elevado converge rapidamente mas também se instabiliza facilmente. Assim, a escolha deste parâmetro terá de ser um compromisso entre estabilidade e tempo de convergência, tipicamente toma o valor entre 10^{-3} e 10^{-1} [44]. Outro fator importante na convergência do algoritmo prende-se com a inicialização dos coeficientes dos filtros. Para um bom funcionamento, os coeficientes devem ser todos inicializados a zero exceto os coeficientes centrais de h_{xx} e h_{yy} , que devem ser inicializados a um [29]. Apesar da simplicidade este algoritmo apresenta um grande inconveniente, a incerteza na convergência das suas saídas. Na verdade, não é possível garantir para que porta de saída do compensador converge cada polarização [47], apenas se pode garantir que as duas saídas do igualador são ortogonais e assim evitar que a mesma polarização esteja nas duas saídas [47]. Este resultado consegue-se através da adaptação das equações de atualização dos coeficientes do filtro à simetria da matriz de Jones, equação (2.16). As equações resultantes são [47]:

$$\begin{aligned} h_{xx} &= h_{xx} + \mu \varepsilon_{CMA,x} x_{out} x_{in}^* \\ h_{xy} &= h_{xy} + \mu \varepsilon_{CMA,x} x_{out} y_{in}^* \\ h_{yx} &= -h_{xy}^* \\ h_{yy} &= h_{xx}^* \end{aligned} \quad (2.42)$$

O LMS é igualmente um algoritmo “cego” e faz uso do símbolo descodificado, definido como R , como referência no cálculo do erro [44, 48]. Assim os sinais de erro podem escrever-se [44, 49]:

$$\begin{aligned} \varepsilon_{LMS,x} &= R_x - x_{out} \\ \varepsilon_{LMS,y} &= R_y - y_{out} \end{aligned} \quad (2.43)$$

Deve notar-se que neste caso o sinal de erro contém informação de fase e amplitude. As equações de atualização dos coeficientes dos filtros são [44]:

$$\begin{aligned} h_{xx} &= h_{xx} + \mu \varepsilon_{LMS,x} x_{out}^* \\ h_{xy} &= h_{xy} + \mu \varepsilon_{LMS,x} y_{out}^* \\ h_{yx} &= h_{yx} + \mu \varepsilon_{LMS,y} x_{out}^* \\ h_{yy} &= h_{yy} + \mu \varepsilon_{LMS,y} y_{out}^* \end{aligned} \quad (2.44)$$

É assim necessário um circuito de decisão, sendo o algoritmo designado de DD-LMS [44, 50]. A consequência é a redução da eficiência do algoritmo e por outro lado o aumento da complexidade da implementação [44, 50]. Uma vantagem desta implementação é a compensação da fase do sinal não sendo necessário um circuito à posteriori.

No projeto dos filtros é necessário ter em conta o número mínimo de coeficientes do filtro. Este valor pode ser estimado pela equação [36]:

$$N_{PMD} = \left\lceil \frac{\tau_{DGD}}{T} \right\rceil. \quad (2.45)$$

Na equação (2.45) T representa o período de amostragem e τ_{DGD} representa o valor médio da DGD. Este valor pode ser estimado de acordo com o comprimento da fibra como referido na secção “2.2.2 Dispersão dos Modos de Polarização”, considerando que a fibra é a única fonte deste atraso de grupo.

2.4.3 Estimação e correção de fase

Não apresentando os lasers um comportamento ideal, do batimento entre o sinal recebido e o oscilador local resulta uma diferença de fase e frequência. Estes desfasamentos continuam a ser um grave problema para a eficiência dos sistemas coerentes [7, 51]. Visando a compensação do seu efeito um elevado número de métodos foram propostos [52-57], em que os mais frequentes são baseados em malhas sem realimentação devido à sua relativa facilidade de implementação para elevados ritmos de transmissão e estabilidade. Neste tema apenas se irá abordar os métodos de estimação e correção de fase. O ruído de fase (PN) tem-se mostrado um dos grandes limitadores da eficiência destes algoritmos [7, 58]. Este está diretamente associado à largura de linha dos lasers cujos valores típicos variam de 100kHz a 10MHz [58]. O método mais abordado é o *Viterbi* e *Viterbi* (V&V) que recorre a uma não-linearidade de quarta ordem para eliminar o formato de modulação (2.46), no caso do QPSK [8, 40, 59].

$$\phi[k] = \frac{1}{2N+1} \sum_{n=-N}^N \arg\{x_{in}^4[k+n]\} \quad (2.46)$$

Na equação (2.46) $\phi[k]$ representa a fase média estimada para um conjunto de $2N+1$ amostras. Estimado este valor, é possível compensar o sinal rodando a constelação do valor simétrico. A Figura 2.16 apresenta um modelo de implementação do algoritmo.

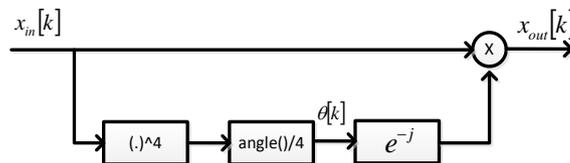


Figura 2.16: Modelo representativo do algoritmo V&V.

Uma vez que este algoritmo mede para cada bloco de amostras o erro de fase absoluto, não apresenta grande tolerância a rotações de fase. Atendendo ao sinal

modulado e com ruído de fase, equação (2.47), ao elevar à quarta ordem obtém-se a equação (2.48).

$$x_{in}[k] = \alpha e^{j(2k+1)\frac{\pi}{4} + \varphi_k} \quad (2.47)$$

$$x_{in}^4[k] = \alpha^4 e^{j(2k+1)\pi + 4\varphi_k} = \alpha^4 e^{j(\pi + 4\varphi_k)} \quad (2.48)$$

Nas equações (2.47) e (2.48) α representa um fator de amplitude e φ_k o ruído de fase. Despresando o termo de π , que pode ser previamente compensado, a fase detetada é $4\varphi_k$. Como o intervalo de medida é de $-\pi$ a π o valor máximo admissível em módulo para φ_k é de $\frac{\pi}{4}$ [60]. Desta forma, apenas tolera rotações de fase de $\pm \frac{\pi}{4}$ em torno do ponto ideal. Esta limitação pode ser vista de outro modo, sendo o algoritmo “cego” não tem capacidade de distinguir as transições entre quadrantes, isto é, entre zonas de decisão. Para reduzir este problema, uma boa adaptação ao algoritmo de *Viterbi* é apresentado na Figura 2.17 [7].

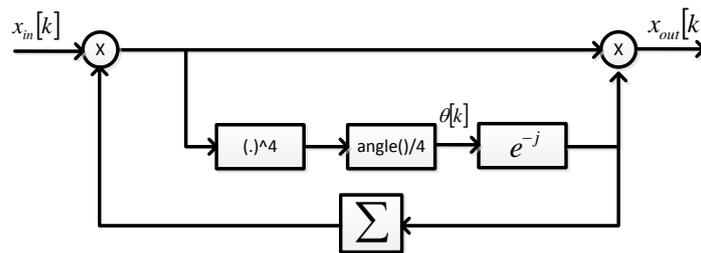


Figura 2.17: Exemplo representativo do algoritmo V&V [7].

Apesar de ser utilizada uma malha de realimentação, o circuito ao realimentar a fase cumulativa torna-se mais robusto ao ruído de fase e consegue seguir melhor a deslocação da constelação. Neste caso, o circuito tolera qualquer valor de fase desde que a transição instantânea não seja superior em módulo a $\frac{\pi}{4}$. Considerando a paralelização do circuito, apenas existe uma linha de realimentação que corresponde à média das fases estimadas. Esta abordagem apesar de apresentar um elevado desempenho em ambientes de simulação, é frequentemente evitada visto que o atraso na malha de realimentação pode limitar o desempenho e a estabilidade do compensador em implementações reais [57]. Contudo, o desempenho deste algoritmo pode ser superior ao do algoritmo V&V sem linha de realimentação.

3 OSIP 2012

Neste capítulo é apresentada a nova versão do simulador OSIP (Optical Simulator Platform), OSIP 2012. Este foi desenvolvido na linguagem de programação Matlab[®], devido ao conjunto alargado de funções pré-definidas e à sua flexibilidade e simplicidade na manipulação numérica de matrizes e cálculo científico, e é constituído por um conjunto de funções que ao serem executadas sequencialmente geram a interface gráfica e implementam as funcionalidades dos componentes. Uma das grandes vantagens deste simulador é a facilidade que apresenta para consultar o código que descreve cada componente ou subsistema ótico, bem como adicionar novos componentes. Também, a sua alargada compatibilidade com versões do Matlab[®] torna-o numa ferramenta bastante atrativa. Por outro lado, apresenta-se como uma ferramenta interativa com uma vasta biblioteca de modelos que tem sido desenvolvida e atualizada pelos alunos e docentes ao longo dos últimos anos. Ao longo deste capítulo são abordadas as funcionalidades e as características do simulador. Na secção 3.1 é explicada a arquitetura de funcionamento do simulador, sendo dados alguns exemplos de modelos de execução de funcionalidades, organização de ficheiros e ambiente gráfico. Na secção 3.2 são descritas as novas funcionalidades do simulador. São apresentados alguns modelos de algoritmos de controlo e identificados os componentes introduzidos e modificados.

3.1 Arquitetura do simulador

Como foi referido o simulador foi desenvolvido na plataforma Matlab[®]. Sendo esta uma linguagem de programação simples e versátil permite adaptar o simulador a cenários muito diversos. Este ponto é muito importante pois para além de permitir aos utilizadores o desenvolvimento dos seus próprios componentes, possibilita também sucessivas atualizações. Assim, o *software* de simulação estará sempre preparado para estudar os sistemas mais recentes.

O simulador encontra-se organizado num conjunto de variáveis globais sobre as quais são realizadas todas as operações, desde adicionar, remover, configurar componentes ou mesmo realizar ligações. Os componentes e funcionalidades do simulador encontram-se distribuídos segundo uma árvore de diretórios, que está representada na Figura 3.1.



Figura 3.1: Organização das diretorias do simulador OSIP.

O código referente aos componentes, núcleo, etc, encontra-se em diretórios dedicados proporcionando uma maior compreensão da estrutura do simulador e logo uma melhor aprendizagem deste. O diretório *help* armazena todas as funções de ajuda à execução do simulador, sendo que nesta versão está vazia. Dada a importância atribuída à componente gráfica do simulador, os ícones de cada componente são armazenados na diretoria *icons*, enquanto que, o código referente às funcionalidades dos componentes se encontra na diretoria *modules*. Funções auxiliares, são exemplo as funções de cálculo de filtros, ajuste de escalas de frequência e tempo e de suporte à interface gráfica, são armazenadas na pasta *utilities*. A execução de simulações implica a criação de ficheiros temporários os quais são armazenados na diretoria *Temp*, estes que no fim da simulação são eliminados. É exemplo a função “*osipsim.m*” que agrupa todos os componentes que estão na área de desenho e executa a simulação. Já a *Paramdialog* armazena as funções de suporte à criação de janelas de configuração dos componentes. As funções responsáveis pela gestão, escalonamento e comunicação entre processos do simulador encontra-se no diretório *kernel*. Este torna-se assim, de entre todos os diretórios, o mais importante. Sobre ele cai a responsabilidade de agrupar e integrar todas as funcionalidades inerentes ao simulador. Esta estratégia de organização das funcionalidades leva também a que o mesmo processo seja adotado para a organização de todos os componentes. Estes estão organizados segundo quinze categorias, as quais são apresentadas na Figura 3.2.

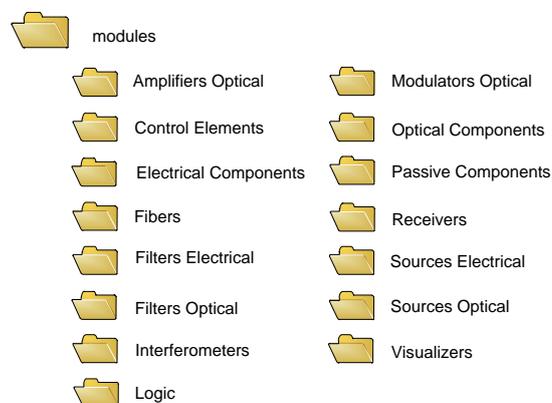


Figura 3.2: Conteúdo da diretoria *modules*.

A divisão dos ficheiros permite simplificar a compreensão do simulador e por outro lado facilitar o processo de integração de novas funcionalidades. Sempre que um componente é introduzido no ambiente de trabalho, é identificada a sua localização e o ficheiro que o descreve é carregado. O método torna-se assim bastante simples. Atualmente o simulador encontra-se bastante atualizado, com um total de 132 componentes.

Os sinais e variáveis responsáveis pela configuração dos componentes, passagem de parâmetros e registo da informação do ambiente de simulação, nomeadamente ligações entre componentes, são armazenados sobre a forma de estruturas. Estas em Matlab[®] são bastante simples de criar e manipular e permitem obter um ambiente de simulação simples, versátil e facilmente compreensível. As estruturas utilizadas são cinco: *OSIP_GLOBALS*, *OSIP_PARAM*, *OSIP_SCHEME*, *OsipHandles* e *NUM*. A cada uma delas está associada uma funcionalidade, na *OSIP_GLOBALS* está registada toda a informação geral, são exemplo a diretoria de trabalho, posicionamento da janela e tamanho dos componentes. Para facilitar as operações de criação de novos componentes, inicialização de variáveis e configuração das simulações a informação relacionada com a configuração de cada componente introduzido está armazenada na variável *OSIP_PARAM*. O registo desta informação em variáveis globais aumenta a eficiência e organização do simulador, portanto procura-se que toda a informação esteja sintetizada de forma clara, compreensível e em poucas variáveis. O posicionamento e as ligações de todos os componentes presentes na área de desenho estão associados à variável *OSIP_SCHEME*. A informação relativa às referências de interface gráfica de cada componente, e portanto informação crítica, está associada à estrutura *OsipHandles*. É assim facilitado o acesso a funções de configuração da interface com o utilizador, de que é exemplo, a configuração dos componentes. A última estrutura, e não menos importante, é a *NUM*. Esta armazena os parâmetros de configuração da simulação e torna-se, no ponto de vista do utilizador, na variável com maior importância visto que permite o seu acesso na interface gráfica através dos parâmetros numéricos.

A combinação de todas as variáveis globais, e funções, que representam os componentes, operações auxiliares e mesmo críticas do simulador, criam o ambiente de simulação que é apresentado na Figura 3.3.

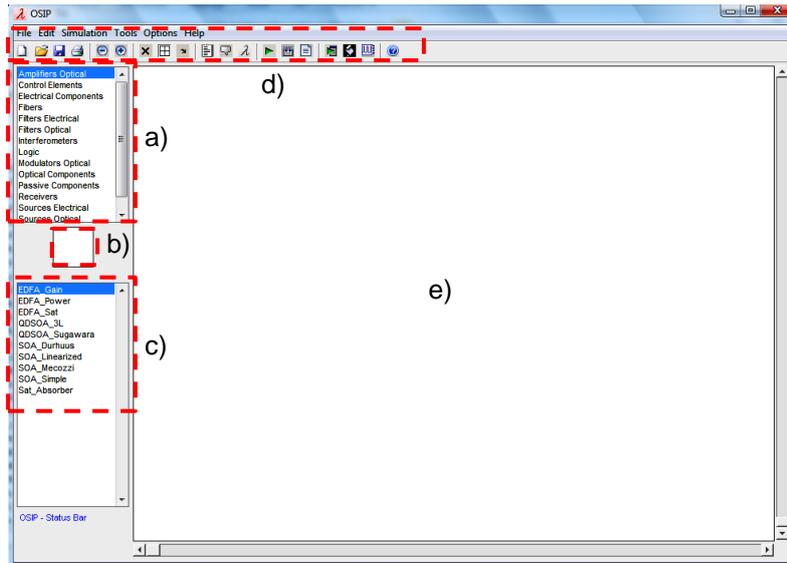


Figura 3.3: Ilustração do ambiente de simulação do OSIP: a) – categorias de componentes; b) – pré-visualização do componente; c) – componentes da categoria seleccionada; d) – barra de menus; e) – área de desenho.

A janela de trabalho apresentada na Figura 3.3 apresenta um aspeto semelhante às janelas do *Windows* e foi desenvolvida com auxílio da funcionalidade GUIDE do Matlab[®]. Esta permite desenvolver de forma simplificada janelas de interação com o utilizador, tornando o uso do simulador mais intuitivo. Apesar de esta ferramenta facilitar a construção de interfaces gráficas, exige o desenvolvimento de algoritmos de controlo. Para facilitar a interação com o utilizador, todas as categorias de componentes são identificadas e listadas (ver Figura 3.3–a)) assim como os seus componentes (ver Figura 3.3–b)). Desta forma, a escolha dos componentes a utilizar torna-se bastante intuitiva. Após a seleção do componente, o seu ícone aparece na zona de pré-visualização (ver Figura 3.3–c)), facilitando o processo de introdução do componente na área de desenho. O simulador apresenta também uma barra de menus (ver Figura 3.3–d)) onde são disponibilizadas todas as ferramentas necessárias à simulação. São exemplo a criação de novas folhas de desenho, iniciar uma simulação e *zoom*. É objetivo concentrar todas as funcionalidades do simulador em zonas restritas e de forma gráfica de modo a que seja disponibilizada uma grande área de desenho e que a experiência do utilizador se torne mais intuitiva. Deve notar-se que apesar da ideia que transparece ao utilizador é de que as operações são realizadas em ambiente gráfico tal não é verdade. Existem um conjunto de funções que executam, de forma transparente ao utilizador, as funcionalidades desejadas. É exemplo a função de início de simulação. Esta gera automaticamente um ficheiro de código Matlab[®] cuja organização é apresentada na Figura 3.4.

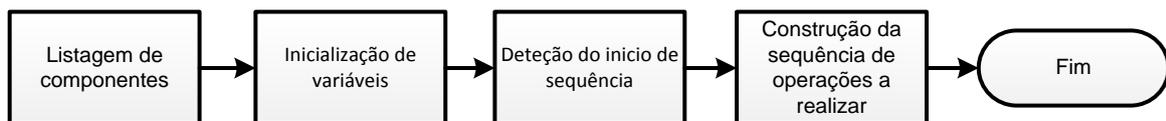


Figura 3.4: Estrutura da função de execução da simulação.

O primeiro passo realizado pela função de controlo é percorrer as variáveis globais para identificar quais são os componentes que estão na área de desenho, identificando-os em cabeçalho no código posteriormente gerado. De seguida inicializa as variáveis temporárias para simular as ligações, isto é, propagação de sinais, e para passagem de parâmetros de configuração. São agora identificadas as sequências de componentes a simular e o código correspondente é criado. O uso de estruturas de dados permite simplificar as operações necessárias neste processo, não sendo perceptível ao utilizador a execução de todos os passos intermédios. Terminadas as operações, a função é guardada e executada. Este ficheiro de código pode ser consultado e executado pelo utilizador. Sobre ele podem ser realizadas todo o tipo de modificações, permitindo ao utilizador adaptar o ambiente de simulação às funções desejadas. O simulador proporciona assim um ambiente gráfico que permite desenhar os modelos de forma intuitiva e por outro lado uma forma eficiente de representação em código Matlab[®] das funções esquematizadas. Ainda associado ao desenvolvimento de janelas gráficas, deve ter-se em consideração as janelas de configuração de componentes e da simulação, Figura 3.5.

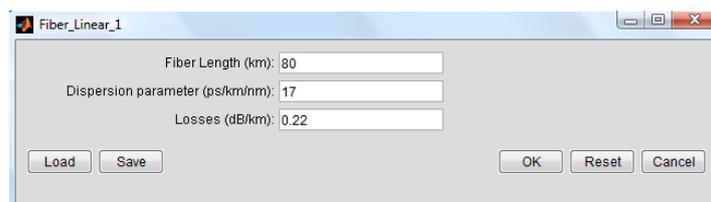


Figura 3.5: Exemplo de uma janela de configuração da fibra linear.

A Figura 3.5 apresenta a janela de configuração da fibra linear. Apesar de este exemplo ser um pouco restritivo, as janelas de configuração das portas dos componentes e mesmo dos parâmetros numéricos são semelhantes. Como referido anteriormente os valores de configuração do componente são, após o fecho da janela, armazenados na variável global *OSIP_PARAM* e a criação das janelas de configuração é realizada com auxílio das funções presentes no diretório *paramdialog*. Existem outras funcionalidades com interface gráfica, das quais se destacam as operações de gravação, abertura de ficheiros previamente gravados e de impressão. De salientar que o simulador apresenta um número bastante elevado de funcionalidades, não sendo possível referir a sua totalidade neste documento.

Outro aspeto importante a considerar é a distinção dos tipos de sinais possíveis de utilizar no simulador. Estes encontram-se divididos em quatro grupos: os óticos, os elétricos, os lógicos e os numéricos. A cada tipo de sinal é atribuída uma cor no ambiente de simulação (azul para o ótico, vermelho para o lógico, verde para o elétrico e rosa para o numérico), permitindo uma fácil distinção dos sinais e uma melhor compreensão dos circuitos a simular. Assim cada componente será caracterizado não só pelas suas portas de entrada e saída, mas também pelo seu tipo. Existem também portas de controlo superiores e inferiores que permitem facilitar a compreensão de algumas das funcionalidades desejadas. O modelo de um componente é apresentado na Figura 3.6.

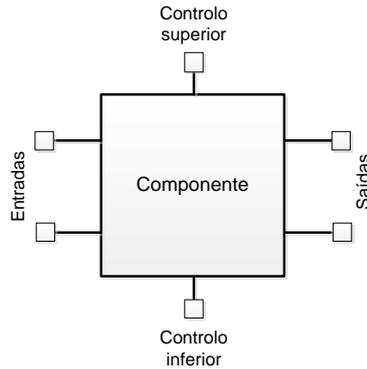


Figura 3.6: Modelo de um componente no OSIP.

Atendendo ao modelo, os portos de entrada são apresentados à esquerda, os de saída à direita e os de controlo em cima e em baixo do componente. Estes não são exatamente portos de entrada, mas funcionam como entradas de sinal. Um bom exemplo da sua utilização é um modulador, Figura 3.7. Neste caso estas portas representam fielmente os sinais de controlo do componente.

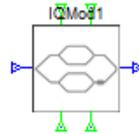


Figura 3.7: Exemplo ilustrativo do modulador IQ.

Neste exemplo os portos de controlo assumem uma natureza elétrica, na verdade estes podem representar outros tipos de sinais (óticos, lógicos ou numéricos). Outro aspeto que se deve ter em consideração é a organização das portas no ambiente de simulação, Figura 3.8.

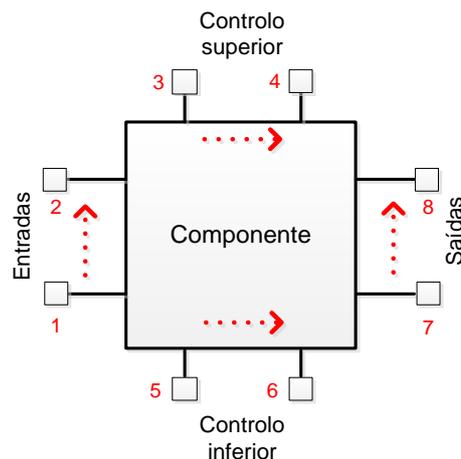


Figura 3.8: Organização das portas do componente no ambiente de simulação.

A contagem do número das portas é realizada de baixo para cima e inicia-se nas portas de entrada, seguindo-se as portas de controlo superior e inferior. Nestas, a

contagem realiza-se da esquerda para a direita, sendo que as portas superiores são as primeiras a ser consideradas. Por fim são enumeradas as portas de saída, sendo a contagem realizada de baixo para cima. Para facilitar ao utilizador a manipulação dos componentes, é possível alterar graficamente o número e tipo de portas de entrada e de saída. Para proceder a esta configuração basta aceder às definições do componente e seleccionar *Port definitions* ou pressionar *shift+click* sobre o ícone. Concluído um destes passos, é apresentada a janela de configuração das portas do componente, Figura 3.9.

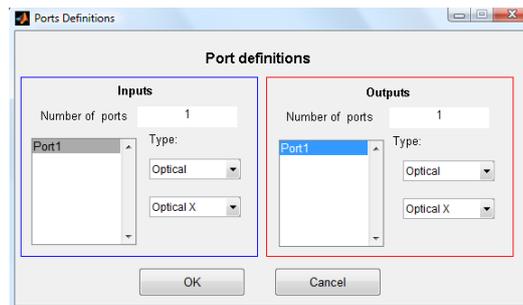


Figura 3.9: Janela de configuração das portas de entrada/saída de um componente.

O utilizador pode assim seleccionar o número de portas de entrada e de saída do componente. Caso o número atribuído seja nulo significa que o componente não apresenta portas de entrada e/ou saída sendo as restantes funcionalidades desativadas. Outro aspeto a ter em consideração é que apenas é possível alterar o número de portas caso o componente tenha permissão para isso. Esta permissão é atribuída no ato de criação do componente. Para cada porta o utilizador pode depois escolher qual o tipo de sinal.

Para facilitar as operações críticas do simulador e tornar as operações de carregamento e manipulação de cada componente mais simplificada, as funções referentes às funcionalidades dos componentes adicionados devem seguir uma estrutura pré-definida, que é apresentada na Figura 3.10.

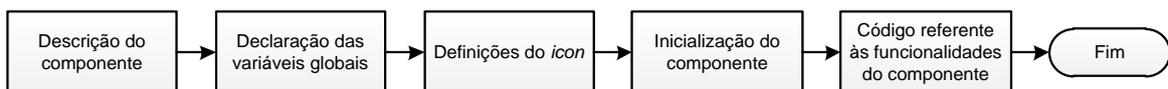


Figura 3.10: Exemplo da estrutura das funções descritivas dos componentes.

A primeira tarefa a definir é descrever o componente. Esta informação é bastante útil aos utilizadores do sistema, pois pode ser consultada através do comando *help* do Matlab® e permite ao utilizador saber como funciona cada componente. De seguida devem definir-se quais as variáveis globais a utilizar, possibilitando a passagem de parâmetros e consequentemente uma configuração globalizada. As definições do ícone e a inicialização do componente estão intrinsecamente ligadas, apesar de executadas em intervalos temporais distintos. Aquando da introdução do componente as definições do ícone são carregadas, das quais se destacam o número, o tipo de portas e os parâmetros de configuração. Posteriormente estes valores são importados e o componente é inicializado, podendo ser visualizado corretamente no ambiente de trabalho do OSIP. O código que descreve a função de cada componente deve ser otimizado de forma a

garantir a eficiência e operacionalidade do ambiente de simulação. Esta abordagem requer um conhecimento mais pormenorizado da arquitetura do simulador. Para facilitar ao utilizador comum a introdução dos seus próprios componentes é disponibilizada uma opção de integração de componentes em ambiente gráfico. Para fazer uso desta opção deve seleccionar-se a opção *create new module* no separador *tools* da barra de ferramentas. Seleccionada a opção aparece a janela de configuração da Figura 3.11.

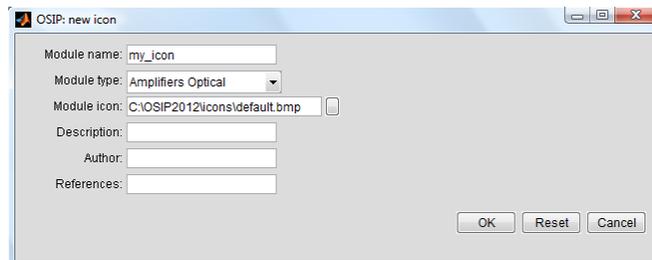


Figura 3.11: Ilustração do primeiro passo do método de adicionar graficamente novos componentes.

Basta agora definir o nome do componente, a sua categoria, a localização do seu ícone e a sua descrição. O processo de introdução do componente contém mais dois passos, nos quais se definem as características do componente. São exemplo o número e tipo de portas. Terminadas as configurações é aberta uma janela do Matlab[®] para escrever o código referente às funções do componente. O método torna-se assim bastante simples e intuitivo, não sendo necessário o conhecimento pormenorizado do funcionamento do simulador. Ainda assim, deve notar-se que os procedimentos necessários são os mesmos que para a construção manual de um componente. O simulador também permite agrupar um conjunto de componentes de forma a obter-se uma perspectiva de mais alto nível e também de simplificar os esquemas que se pretendem simular. Para tal deve seleccionar-se o conjunto de componentes que se pretende agrupar e seleccionar a opção *group icons* no separador *tools* da barra de ferramentas. O processo de agrupamento é iniciado com a janela da Figura 3.12.

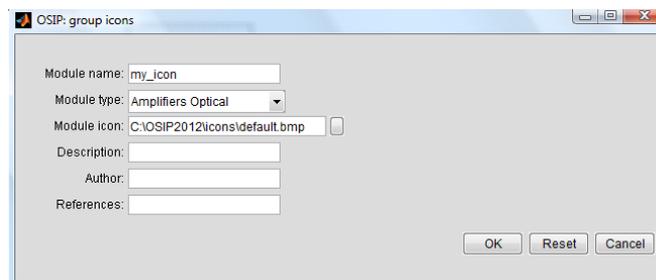


Figura 3.12: Ilustração do primeiro passo do método de agrupar graficamente os componentes.

Este processo é idêntico ao de adicionar componentes no ambiente gráfico. Deve definir-se o nome do componente resultante, a sua categoria, a localização do seu ícone e por fim configurar as portas. Após todas as configurações é apresentado um ficheiro de código Matlab[®] que descreve o componente final, sendo possível ajustar manualmente as configurações.

Outro aspeto fundamental a ter em consideração no simulador é a configuração da simulação. Esta pode ser vista como o conjunto de 3 passos: configuração dos parâmetros de simulação, definição das constantes físicas e configuração dos componentes. Quando se introduz um componente é gerada automaticamente uma janela de configuração, idêntica à da Figura 3.5, permitindo ao utilizador configurar corretamente cada componente. As constantes físicas são definidas no *kernel* do simulador e não necessitam de ser constantemente configuradas. Ainda assim, é importante conhecer os seus valores para comparar com cálculos teóricos. Estes podem ser consultados acedendo à opção *Simulation* na barra de ferramentas e selecionando *Physical constants*. Ao selecionar a opção o simulador abre a função do *kernel* responsável por definir as constantes físicas, Figura 3.13.

```

1  function declare_phyconst()
2  $DECLARE_PHYCONST Global variable to store useful physical constants.
3  $
4  $ Use this function to set useful physical constants in OSIP platform. In
5  $ you modules define global NUM.PhysicalConst and access any physical value you
6  $ need.
7  $
8  $
9  $ Optical SIMulation Platform (OSIP - www.av.it.pt/osip)
10
11  global NUM
12
13  NUM.PhysicalConst.c0 = 2.99792458e8;    %light speed in vacuum [m/s]
14  NUM.PhysicalConst.h = 6.626176e-34;    %Planck constant [J.s]
15  NUM.PhysicalConst.hs = 1.0545887e-34;  %normalized Plack constant [J.s/pi]
16  NUM.PhysicalConst.q = 1.602189e-19;    %electron charge [C]
17
18  NUM.PhysicalConst.k = 1.380662e-23;    %Boltzman constant [J/K]
19  NUM.PhysicalConst.kT290 = 4.00357e-21; %kT for T=290K [J]
20  NUM.PhysicalConst.kT290q = 0.02585;    %kT/q for T=290K [eV]
21
22  NUM.PhysicalConst.epsilon_0 = 1/36*pi*1e-9; %vacuum dielectric permittivity [F/m]
23  NUM.PhysicalConst.miu_0 = 4*pi*1e-7;    %vacuum magnetic permeability [H/m]
24  NUM.PhysicalConst.Z_0 = 376.7;         %vacuum intrinsic impedance [ohm]
25
26  NUM.PhysicalConst.m0 = 9.10938188e-31; %electron mass
27

```

Figura 3.13: Função do *kernel* do simulador que define as constantes físicas da simulação.

Por fim a configuração dos parâmetros de simulação é realizada selecionando no separador *Simulation* da barra de ferramentas a opção *Numerical parameters*, sendo a janela de configuração da Figura 3.14 apresentada.

Figura 3.14: Janela de configuração dos parâmetros de configuração da simulação.

Na janela apresentada na Figura 3.14 pode definir-se o ritmo da sequência binária a gerar e o seu comprimento, a frequência de amostragem e o comprimento de onda de referência. Adicionalmente é possível definir o número de blocos que se pretende simular e qual o número máximo de erros que é admissível na simulação. O número máximo de erros apenas é tomado em consideração se a opção *use* estiver selecionada. Definidas todas as configurações o simulador está apto para simular os sistemas desejados.

Para maior facilidade na exportação dos esquemas desenhados o simulador disponibiliza uma função para a sua impressão. Para utilizar esta função basta selecionar a opção *print* no separador *file* da barra de ferramentas, sendo gerada a janela da Figura 3.15.

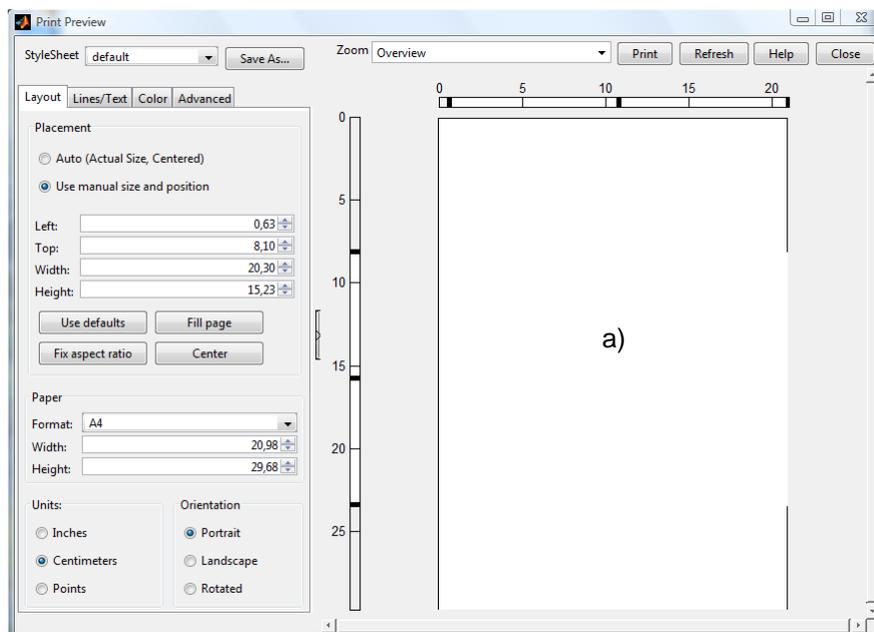


Figura 3.15: Janela de impressão dos esquemas presentes na área de desenho: a) – zona de pré-visualização da área de desenho.

No caso apresentado a área de desenho estava em branco, logo não é possível verificar nenhum esquema na zona de pré-visualização (ver Figura 3.15-a)). Na janela da Figura 3.15 é possível definir todos os aspetos de configuração da página e posteriormente imprimir em papel, formato XPS (XML Paper Specification) ou PDF (Portable Document Format).

3.2 Novas funcionalidades

A versão atual do OSIP, OSIP 2012, foi desenvolvida partindo da versão anterior, OSIP 3.1. As diferenças entre as duas são bastantes, resultando a incompatibilidade dos ficheiros de simulação. A versão 3.1 do OSIP estava direcionada à modulação em intensidade, permitindo apenas a simulação de formatos de modulação a 1 *bit* por símbolo. Esta é uma das grandes limitações desta versão. Por outro lado, a abordagem de simulação que estava a ser adotada tornava bastante difícil realizar simulações

quando se pretendia detetar um número reduzido de erros. Poderiam ser necessárias sequências bastante longas tornando as simulações computacionalmente exigentes. Estas limitações aliadas à necessidade de simular sistemas coerentes foram os principais focos de desenvolvimento no simulador. Para melhor compreender as diferenças apresenta-se na Figura 3.16 os parâmetros de configuração da simulação nas duas versões do simulador.

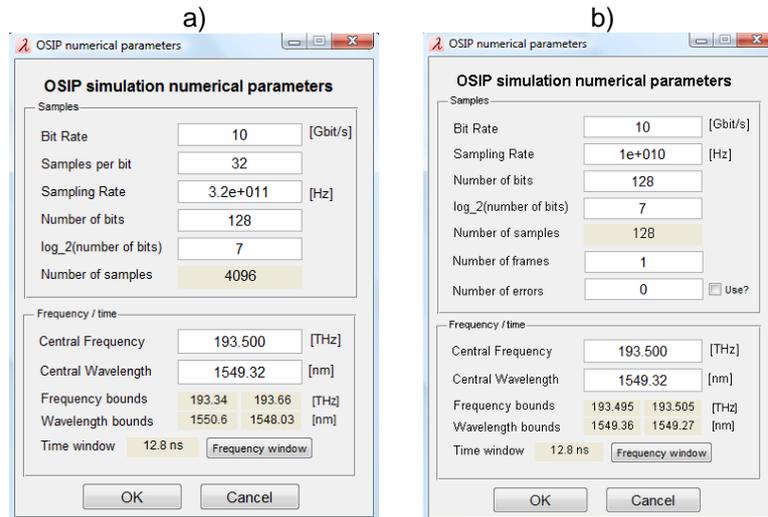


Figura 3.16: Comparação dos parâmetros de configuração das duas versões do OSIP: a) - versão 3.1; b) - versão 2012.

São visíveis na Figura 3.16 grandes diferenças. Na versão anterior do simulador o número de amostras era calculado tendo em conta a relação entre a frequência de amostragem e ritmo de *bits*, sendo esta fixa e sempre maior ou igual a um. Atualmente é permitido que a frequência de amostragem seja inferior ao ritmo de *bits*. Apesar de aparentemente não parecer correto, na verdade é possível. Sendo que um dos objetivos é a atualização para simulação de sistemas coerentes, o simulador passa a ter capacidade de lidar com ritmos de símbolos que serão inferiores aos de *bit*. Desta forma a alteração realizada faz todo o sentido. Deve também notar-se que o campo *Samples per bit* foi eliminado. Na verdade, pretendendo-se que o simulador seja capaz de funcionar corretamente para vários formatos de modulação, não seria possível que este parâmetro fosse de configuração geral. Esta afirmação pode-se verificar facilmente, visto que o simulador apenas admite uma frequência de amostragem, e como tal caso seja pretendido simular em simultâneo vários formatos apenas seria possível definir, nos parâmetros globais, um dos casos. A abordagem adotada permite que este processo se torne mais intuitivo sendo que no caso de formatos de ordem superior à primeira deve ser utilizado um componente de *resample*, que faz a amostragem dos sinais de acordo com o número de amostras pretendidas. Claro que este valor deve ser calculado de acordo com a relação:

$$N_A = \frac{\text{frequência de amostragem}}{\text{taxa de símbolos}}. \quad (3.1)$$

Na equação (3.1) N_A representa o número de amostras a utilizar, a taxa de símbolos deve ser calculada de acordo com a equação:

$$\text{taxa de símbolos} = \frac{\text{taxa de bits}}{\log_2(M)}, \quad (3.2)$$

sendo M o número de símbolos da constelação considerada. Este cálculo deve ser realizado para cada formato de modulação que se pretende utilizar e assim configurar corretamente a simulação.

O método de *Monte-Carlo* é dos métodos de simulação bastante atrativos uma vez que possibilita a divisão da sequência de dados a simular em blocos mais pequenos os quais são simulados sequencialmente, reduzindo as exigências computacionais em termos de memória e, em geral, no tempo de computação. À medida que os vários blocos são simulados, é realizada a soma cumulativa dos erros ocorridos. Com este método, o número de *bits* definidos nos parâmetros numéricos correspondem ao comprimento de cada bloco e o valor definido no campo *Number of frames* o número total de blocos a utilizar. O número total de bits gerados na simulação corresponde ao produto dos dois. Esta é uma das grandes vantagens da nova versão do simulador. Associado a este método pode desenvolver-se outras técnicas de simulação de que é exemplo a paragem da simulação quando é detetado ou ultrapassado o número de erros desejado. Para ativar esta funcionalidade basta seleccionar a opção *use* e o número máximo de erros admissível deve ser definido no campo *Number of errors*. Para poder combinar estas funcionalidades é necessário um algoritmo de controlo, que é representado na Figura 3.17.

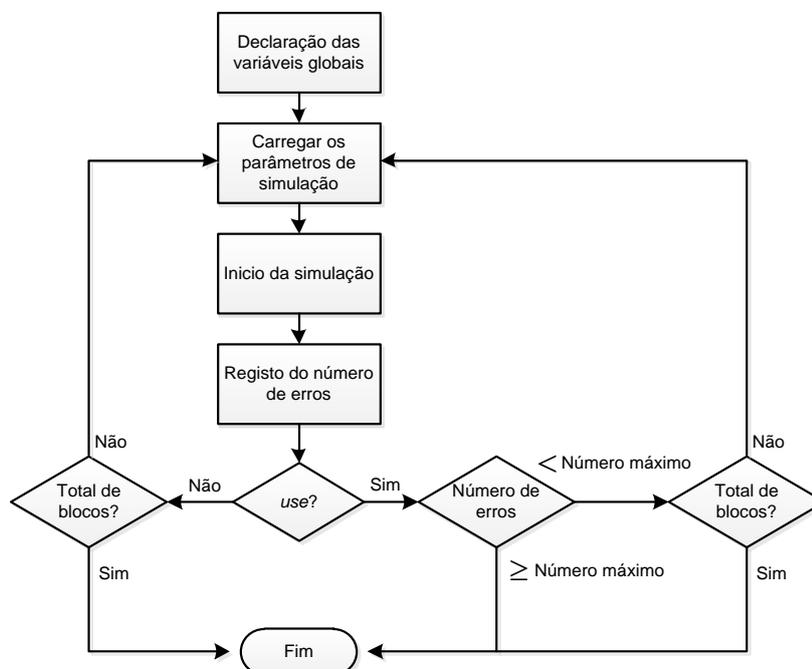


Figura 3.17: Diagrama do algoritmo de controlo do método de simulação.

O algoritmo recorre à recursividade de forma a poder simular sequencialmente todos os blocos definidos. De uma forma geral, o simulador apenas termina a simulação quando são simulados o conjunto de blocos definidos nos parâmetros. Caso seja ativada a opção para admitir um número máximo de erros, assim que este valor é detetado a simulação é interrompida e os resultados obtidos em cada bloco podem ser consultados na linha de comandos do Matlab[®]. A exigência do método não só necessita do algoritmo de controlo como também de um conjunto de variáveis que permitam armazenar os resultados ao longo da simulação de cada bloco. Para esse fim foi necessário proceder à alteração da variável global *NUM*. A Tabela 3.1 apresenta o conteúdo desta variável na versão anterior e atual do simulador.

a)	b)
- bitRate	- bitRate
- sampRate	- sampRate
- centerFreq	- centerFreq
- nBits	- nBits
- nSamp	- nSamp
- t	- frames
- f	- step
- EmptySignal	- BERTresults
- PhysicalConst	- isBERTstop
	- BERTstop
	- EVM
	- t
	- f
	- EmptySignal
	- PhysicalConst

Tabela 3.1: Conteúdo da variável global *NUM* para as versões do OSIP: a) - versão 3.1; b) - versão 2012.

Foram adicionados 6 novos parâmetros: *frames*, *step*, *BERTresults*, *isBERTstop*, *BERTstop* e *EVM*. Todos eles estão associados à simulação por blocos, sendo alguns configurados pelos parâmetros numéricos e outros manipulados por componentes externos. No caso das variáveis *BERTresults* e *EVM* são alteradas pelos componentes BERT e EVM respetivamente. O registo da informação é realizada bloco a bloco, sendo que, o número de erros detetado vai sendo passado entre blocos cumulativamente. Desta forma a variável *BERTresults* é constituída dois campos: *BER* e *Nerrors*. A primeira regista a percentagem de erros por cada bloco e a segunda o número total de erros detetados até ao momento. Do mesmo modo, a variável *EVM* regista os valores do EVM bloco a bloco. Assim para poder utilizar este método é necessário introduzir os componentes BERT e EVM. O parâmetro *step* indica qual o bloco que está a ser simulado em cada momento, enquanto que, a variável *frames* o número total de blocos. As restantes variáveis servem de controlo à condição de paragem, *isBERTstop* indica se o método vai ser utilizado e *BERTstop* o número máximo de erros admissível. Este ajuste das variáveis globais da estrutura *NUM* fez com que as simulações de versões anteriores não sejam compatíveis com a versão atual.

Face à adaptação do simulador para sistemas coerentes também foram inseridos 31 novos componentes, os quais são apresentados, juntamente com as suas

funcionalidades, na Tabela 3.2. A sua descrição encontra-se no Apêndice B – Descrição dos componentes adicionados.

<p>ADC: amostragem dos sinais</p> <p>Amplifier: amplificação dos sinais elétricos</p> <p>Coding: mapeamento de bits em símbolos</p> <p>DSNR: definir um valor de SNR</p> <p>DSP: processamento digital de sinal</p> <p>Decoding: descodificar os símbolos em bits</p> <p>Downsampling: decimador</p> <p>Inverter: inversor do sinal</p> <p>PtoS: converter um sinal paralelo para série</p> <p>Resample: Amostragem do sinal</p> <p>StoP: converter um sinal série para paralelo</p> <p>TIA: amplificador de trans-impedância</p> <p>IQ_Mod: modulador IQ</p> <p>BPD: fotodíodo balanceado</p> <p>DGD: atraso entre as componentes de polarização</p> <p>DOSNR: definir o valor da OSNR</p>	<p>Optical90Hybrid: recuperação das componentes óticas I e Q</p> <p>PBC_S: combinador e divisor de polarizações</p> <p>PDL: perdas dependentes da polarização</p> <p>PhaseRot: rotação de fase</p> <p>PolarizationRot: rotação das polarizações</p> <p>Polarizer: filtrar uma polarização</p> <p>WDM_Demux: demultiplexador de canais WDM</p> <p>WDM_Mux: multiplexador de canais WDM</p> <p>Receiver: recetor definido por software</p> <p>Transmitter: transmissor definido por software</p> <p>BERT: cálculo da BER</p> <p>EVM: cálculo do EVM</p> <p>SavetoFile: exportar sinais para um ficheiro</p> <p>Uconstellation: ver constelação do sinal (ótico e elétrico)</p> <p>PolAlign: Alinhamento das polarizações</p>
---	---

Tabela 3.2: Lista dos componentes adicionados e suas funcionalidades na nova versão do OSIP.

O elevado número de componentes óticos adicionados levou à criação de uma categoria dedicada aos componentes óticos. Os componentes introduzidos permitem manipular os sinais no domínio lógico, elétrico e ótico, possibilitando desta forma o projeto de sistemas de transmissão e receção para formatos de modulação BPSK, QPSK, 16QAM, 32QAM e 64 QAM, bem como para sistemas com diversidade de polarização, permitindo o aumento da eficiência da transmissão. Outra funcionalidade chave da nova versão do OSIP é a possibilidade de simular a propagação e a interação de vários canais óticos existentes nos sistemas óticos com multiplexagem por divisão de comprimento de onda, WDM. Cada um destes canais pode transmitir informação com características diferentes, como formatos de modulação e taxas de transmissão. Também, para fazer face aos fenómenos presentes nas comunicações foi adicionado um DSP. A possibilidade de introduzir processamento de sinal permite simular um sistema ótico na sua totalidade: modulação, transmissão, receção e processamento digital do sinal. O utilizador pode agora desenvolver e aplicar algoritmos de compensação para efeitos lineares e não-lineares. Estes podem ser testados em tempo real, utilizando para o efeito o componente DSP, o que se torna numa mais-valia para o simulador ou em ambiente *offline*, utilizando o componente SavetoFile para armazenar os sinais após a digitalização nas ADCs. Para facilitar o teste de alguns algoritmos foram desenvolvidos alguns componentes dos quais são exemplo *PolarizationRot* e *DGD* para simular a PMD (ver equação (2.17)) e *PhaseRot* para analisar o comportamento dos algoritmos de estimação e correção de fase.

Na versão 3.1 do OSIP todos os componentes eram configurados manualmente. Na nova versão pretende-se que a sua configuração não seja muito exigente, principalmente quando os circuitos de simulação são de grande dimensão e complexidade. Assim para facilitar a configuração foram introduzidos em alguns componentes identificadores. Estes são valores numéricos maiores ou iguais a um. O objetivo é permitir a partilha de

informação entre os componentes e desta forma configurarem-se automaticamente. Um bom exemplo são os blocos de *Coding* e *Decoding*, definindo o mesmo identificador nos dois componentes o bloco de *Decoding* ajusta-se automaticamente para a descodificação do sinal. A passagem dos parâmetros dá-se através de uma estrutura interna *MODParam*, a Figura 3.18 ilustra a transferência de informação entre alguns componentes.

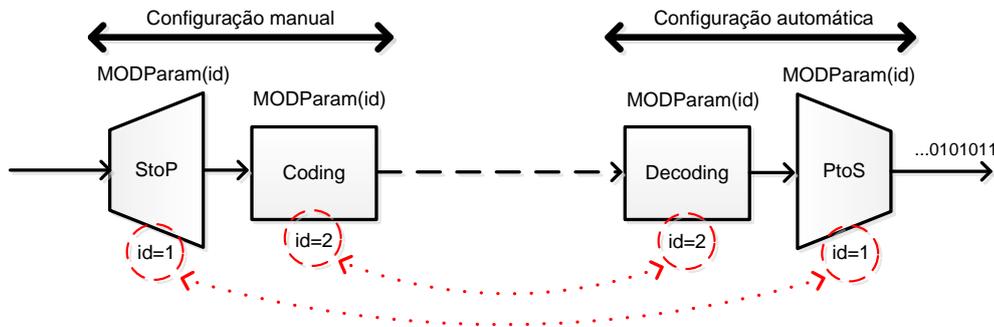


Figura 3.18: Exemplo ilustrativo da passagem de parâmetros entre componentes: id – identificador do componente.

Para cada identificador é criada uma nova entrada na tabela *MODParam* onde são registadas as variáveis que vão ser partilhadas. O método é simples e eficiente, permitindo por exemplo distinguir os vários canais multiplexados por divisão do comprimento de onda (WDM). Caso se pretenda um sistema WDM com muitos canais, para cada transmissor associa-se um identificador, sendo parâmetros de configuração de cada canal armazenados numa tabela permitindo a que outros componentes se configurem automaticamente, reduzindo bastante o número de configurações necessárias à simulação. Outra vantagem é a possibilidade de implementar sistemas multimodo. Apesar de este tópico não estar no âmbito deste trabalho, a possibilidade de distinguir os vários modos permite a implementação destes sistemas.

Tendo em conta todas as modificações foram atualizadas algumas das funções e componentes já existentes. A Tabela 3.3 apresenta uma lista de todos os componentes que foram atualizados e a Tabela 3.4 de todas as funções alteradas.

<ul style="list-style-type: none"> - Fibra linear - Scope - Diagrama de olho - Filtro elétrico - Espectro elétrico

Tabela 3.3: Componentes alterados na nova versão do OSIP.

<ul style="list-style-type: none"> - write_mfile.m - run_simulation.m - numerical_parameters.m - numerical_parameters.fig - about_osip.fig - declare_num.m
--

Tabela 3.4: Funções alteradas na nova versão do simulador.

No caso da fibra linear foi atualizada a função de transferência da dispersão cromática, nos restantes componentes foi necessário alterar os algoritmos de forma a suportarem o ritmo de símbolos e o funcionamento simultâneo com uma ou mais

amostras por símbolo. As funções atualizadas pertencem todas ao *kernel* do simulador. Dada a nova estrutura de simulação os algoritmos de suporte à simulação tiveram de ser modificados, para seguirem o algoritmo apresentado na Figura 3.17. Na Tabela 3.4 dois ficheiros têm a extensão “.fig”, o que significa que se tratam de funções de suporte à interface gráfica. As suas alterações exigiram a introdução de novos algoritmos de controlo aos componentes gráficos, como foi referido aquando da apresentação da simulação por blocos. As funções adicionadas correspondem aos novos campos dos parâmetros numéricos identificados anteriormente (ver Figura 3.16-b)). O algoritmo de controlo para os parâmetros adicionados é exemplificado na Figura 3.19.

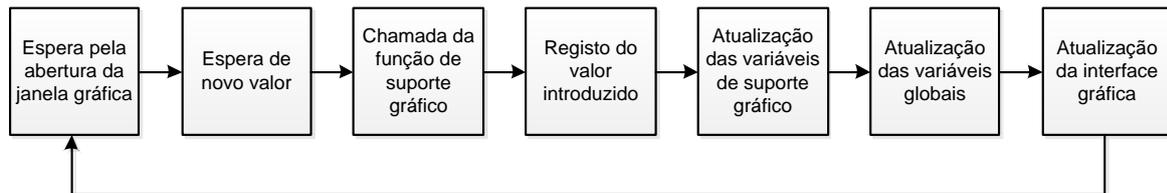


Figura 3.19: Ilustração do algoritmo de controlo para cada parâmetro de configuração gráfico.

O algoritmo conceptualmente é bastante simples e como tal eficiente, permitindo a leitura correta de todos os parâmetros numéricos e assim a configuração da simulação. Quando o utilizador pretende alterar o valor dos parâmetros numéricos, a janela de configuração é aberta, dando-se início ao processo de espera pela introdução de novos parâmetros. Quando é introduzido um novo parâmetro é gerada uma interrupção que chama a função de suporte à componente gráfica, sendo os novos termos armazenados na estrutura gráfica do botão alterado. Terminado este processo, as variáveis de suporte gráfico são atualizadas e é invocada uma nova função que faz com que os valores registados anteriormente sejam armazenados na variável global *NUM* e a interface gráfica atualizada. O utilizador pode agora ver os valores que introduziu nos respetivos campos sem que se aperceba de todo este processo. Deve notar-se que a cada botão de configuração apresentado na Figura 3.16-b) corresponde uma função que responde à interrupção gerada pela introdução de novos parâmetros, a qual deve interagir com as variáveis gráficas e as estruturas globais de modo a atualizar todos os parâmetros envolvidos. Esta janela de configuração, bem como os respetivos botões, são construídos com auxílio da função *GUIDE* do Matlab®. Esta permite à semelhança de outras linguagens de programação, de que são exemplo o C e o Visual Basic, gerar interfaces gráficas. O processo de criação das janelas é bastante simples sendo assistido por uma interface gráfica. Assim apenas é necessário adicionar os algoritmos de controlo para a interligação dos vários botões introduzidos, garantindo assim o bom funcionamento da janela gráfica. Um exemplo muito simples do uso desta função do Matlab® para criar janelas gráficas é apresentado no Apêndice C – Criar uma interface gráfica utilizando o *GUIDE*.

4 Validação do Simulador

No seguimento do desenvolvimento de algoritmos dedicados a elementos de controlo e a componentes é importante o teste do seu desempenho e como tal a sua validação. Nesta secção pretende-se abordar alguns cenários que possibilitem o teste dos componentes e também o desenvolvimento e implementação de novas funcionalidades. Estas estão relacionadas com os algoritmos apresentados nas secções anteriores e que visam a compensação dos efeitos lineares que ocorrem na fibra e a estimação e correção de fase. Na secção 4.1 pretende-se estudar o desempenho do recetor. Este aspeto é fundamental pois permite concluir se este se comporta como esperado teoricamente. A implementação e validação dos algoritmos referidos são realizadas na secção 4.2. Após a validação dos algoritmos está-se apto para simular alguns modelos, sendo na secção 4.3 abordados os sistemas WDM e na 4.4 um sistema modulado nas duas componentes de polarização.

4.1 Análise do desempenho do recetor com o ruído

Um aspeto importante a estudar é o desempenho dos componentes, dando especial ênfase ao recetor. É assim necessário estudar o envio e receção da informação de modo a ter a certeza que o sinal está a ser corretamente transmitido e decodificado. Os modelos do transmissor e do recetor adotados encontram-se na Figura 4.1 e na Figura 4.2 respetivamente.

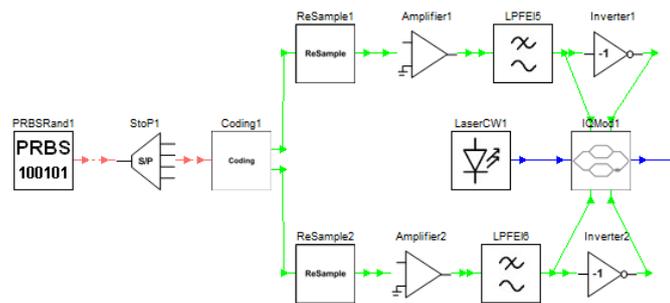


Figura 4.1: Modelo do transmissor ótico implementado no OSIP para as simulações com ruído.

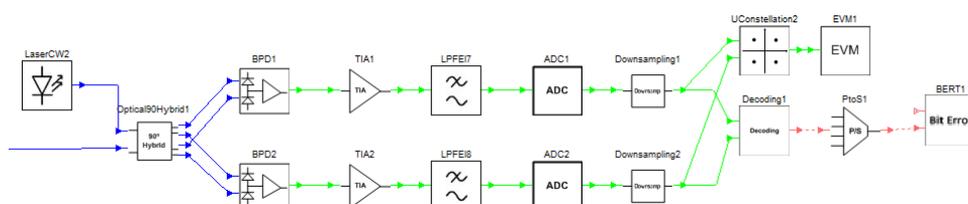


Figura 4.2: Modelo do recetor ótico implementado no OSIP para as simulações com ruído.

O sinal a transmitir pode ser modulado com um formato de modulação arbitrário (BPSK, QPSK, 16QAM, 32QAM ou 64QAM), cuja configuração é realizada nos componentes de codificação e conversão série para paralelo. Após a modulação, o sinal resultante é diretamente aplicado ao recetor. Isto implica que as simulações sejam realizadas sem fibra, o que é um fator fundamental pois apenas se pretende testar o comportamento dos componentes para várias potências de ruído. Antes da conversão opto-eletrónica, o sinal recebido é misturado com o laser local de modo a ser decomposto nas suas componentes em fase e quadratura, e posteriormente é amostrado nas ADCs sendo retiradas duas amostras por símbolo, limite imposto pelo teorema de *Nyquist* para preservar a informação do sinal. Em seguida, o sinal é decodificado e a sequência binária analisada. O desempenho dos componentes do transmissor e do recetor vai ser estudado com base em curvas BER vs SNR e EVM vs SNR, cujos resultados obtidos devem estar conforme os apresentados por *Rishad Ahmed Shafik* em [61] e os valores teóricos, cuja dedução se encontra no “Apêndice A – Cálculo teórico da SER e BER”. Para seguir o mesmo raciocínio as medidas a realizar deverão ter em conta a SNR e portanto adicionou-se o componente DSNR para controlar a potência do ruído à entrada do circuito de decisão. Este componente permite adicionar ruído de acordo com a relação entre a SNR à sua entrada e à sua saída. As ADCs e DACs apresentam uma resolução de 8bits e os filtros elétricos são ajustados ao ritmo de símbolos do sinal. A tensão v_{π} dos MZMs é definida a 5V, sendo o ganho dos amplificadores ajustado para 13.98dB. Os lasers do transmissor e do recetor têm uma frequência de emissão de 193.5THz (1549.32nm). O comprimento da sequência de dados gerada é ajustado de acordo com o formato de modulação e é suficientemente longa para detetar o número de erros pretendido (100 vezes superior ao número de bits necessários para detetar a probabilidade de erro esperada teoricamente). A Figura 4.3 apresenta os resultados obtidos para a relação entre o EVM e a SNR.

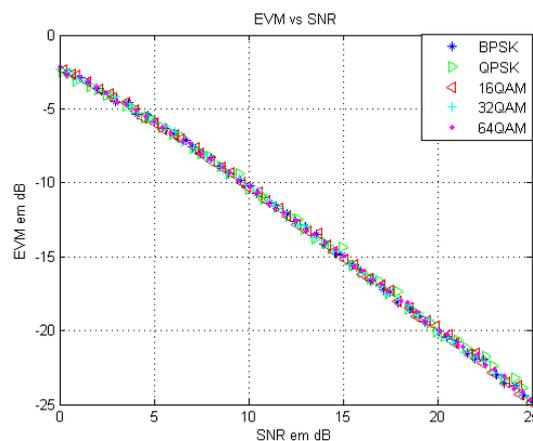


Figura 4.3: Relação entre o EVM e a SNR obtida variando a SNR à entrada do circuito de decisão.

O EVM e a SNR apresentam uma diferença de aproximadamente 0.1dB, pelo que respeita a relação apresentada em [61]. Na verdade, esta relação não se mantém para valores de SNR inferiores a aproximadamente 5dB. Esta diferença registada está associada ao equalizador de amplitudes utilizado no circuito de deteção, que para valores

elevados de ruído deixa de ajustar corretamente o sinal. Este valor é calculado com base na potência média do sinal ideal e do sinal recebido. Considerando o caso limite, SNR de 0dB, a potência média do ruído é igual à potência média do sinal ideal e desta forma a potência média do sinal recebido será o dobro da média ideal. Assim o fator de equalização estará 3dB abaixo do valor desejado e portanto o EVM também deverá ser aproximadamente 3dB inferior ao valor da SNR. Estes valores são medidos na prática. Pode-se verificar na Figura 4.3 que para uma SNR de 0dB obtém-se o EVM de aproximadamente -2.5dB. Registaram-se também as curvas da BER em função da SNR, as quais se encontram na Figura 4.4.

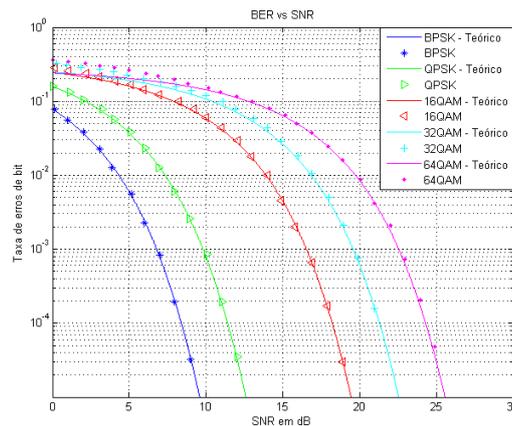


Figura 4.4: Relação entre BER e SNR obtida variando o valor da SNR à entrada do circuito de decisão.

As relações obtidas na prática sobrepõem-se às curvas teóricas. Na verdade, para os casos de 16QAM, 32QAM e 64QAM, para baixos valores de SNR os resultados obtidos na prática não coincidem com os resultados teóricos. Este desfasamento está relacionado com a aproximação considerada no cálculo das curvas teóricas da BER. Neste cálculo considera-se que apenas ocorre um erro de bit por símbolo, o que apenas é válido para valores de SNR relativamente elevados. Nesse caso o deslocamento dos símbolos é reduzido e portanto estes apenas podem transitar para áreas de decisão adjacentes. Considerando que o sinal é codificado com código de *Gray* só ocorre um erro de bit por símbolo. Estas aproximações fazem com que o modelo obtido não apresente os valores desejados nas zonas em que o ruído assume potências mais elevadas. De qualquer forma, é possível verificar que as curvas obtidas na prática apresentam o mesmo comportamento que em [61].

4.2 Implementação de algoritmos DSP

Nesta secção pretende-se implementar algoritmos DSP para compensação dos efeitos lineares que ocorrem nas fibras óticas e analisar o seu desempenho. Na subsecção 4.2.1 estuda-se dois exemplos de algoritmos de compensação de dispersão cromática nos domínios do tempo e da frequência. Na subsecção 4.2.2 é abordado o algoritmo CMA para compensação das não idealidades dependentes da polarização,

PMD, e na subsecção 4.2.3 são considerados os algoritmos de estimação e correção de fase.

4.2.1 Compensação da dispersão cromática

O estudo da compensação da dispersão cromática divide-se em três temas. No primeiro compensa-se a dispersão com um filtro FIR no domínio do tempo, método designado na literatura por TDE, e no segundo aplica-se um filtro no domínio da frequência, que é designado por FDE. Para estes primeiros tópicos a compensação é realizada em condições ideais, isto é, na ausência de ruído. No último tema pretende-se analisar o desempenho dos dois algoritmos na presença de ruído do canal. Os dois tipos de filtros considerados são projetados de acordo com o valor máximo da dispersão acumulada admissível. Após a sua implementação os algoritmos são introduzidos no DSP sendo a compensação realizada em tempo real de simulação. Também seria possível guardar o sinal após a operação de amostragem e realizar o processamento posteriormente, num ambiente *offline*.

4.2.1.1 Compensação com filtro no domínio do tempo

Nesta secção pretende-se projetar e testar em condições ideais o filtro FIR para compensação da dispersão cromática no domínio do tempo. Os coeficientes do filtro são calculados com base na máxima dispersão acumulada permitida e aplicando a equação (2.33). Para o estudo do desempenho do algoritmo consideraram-se dois casos: 500km e 1000km de fibra SSMF. Nos dois, o parâmetro de dispersão é 17 ps/nm.km e o comprimento de onda é de 1549.32 nm . A sequência binária é gerada a 100 Gbit/s , com um comprimento de 2^{14} bits , e modulada com o formato QPSK. O modelo utilizado para o transmissor e recetor é apresentado na Figura 4.5 e na Figura 4.6 respetivamente.

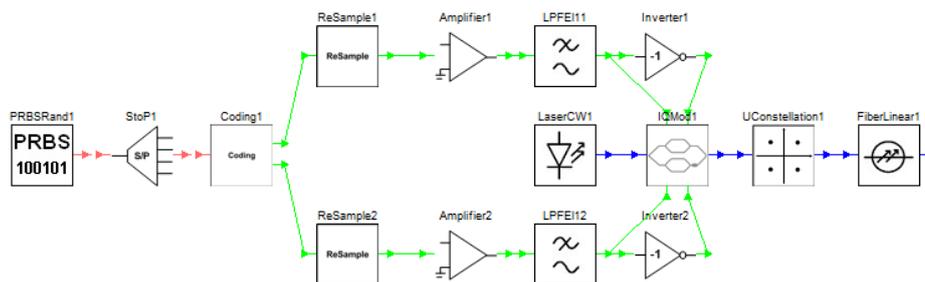


Figura 4.5: Modelo do transmissor óptico implementado no OSIP para estudar o desempenho dos algoritmos de compensação da dispersão cromática.

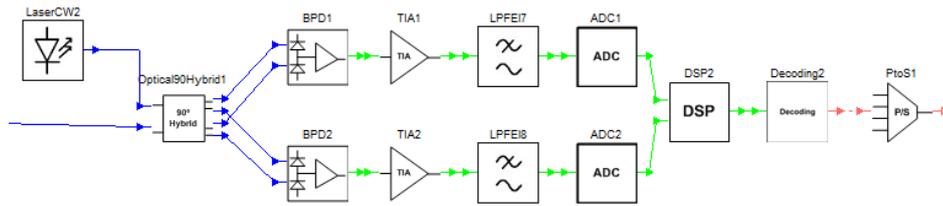


Figura 4.6: Modelo do recetor ótico implementado no OSIP para estudar o desempenho dos algoritmos de compensação da dispersão cromática.

A sequência binária, após codificação e modulação no domínio ótico, é transmitida na fibra e o sinal é posteriormente adquirido no recetor. Este realiza o batimento entre o sinal recebido e o laser local, dividindo o sinal nas suas componentes em fase e quadratura. Cada uma destas componentes é amostrada numa ADC cuja frequência de amostragem é o dobro do ritmo de símbolos do sinal. O sinal digitalizado é processado no DSP, onde se aplicam os filtros que invertem a resposta impulsional da dispersão cromática, e é posteriormente decodificado. As restantes configurações do transmissor e do recetor encontram-se na Tabela 4.1.

Transmissor	Recetor
- Filtro de Bessel a 50GHz	- Filtro de Bessel a 50GHz
- DAC de 8bits	- ADC de 8bits
- Ganho de 13.98dB no amplificador	- 2 amostras/símbolo
- Laser a 193.5THz	- Laser a 193.5THz
- V_{π} de 5V	

Tabela 4.1: Configurações do transmissor e do recetor.

Deve notar-se que os filtros elétricos do transmissor e do recetor são ajustados para o ritmo de símbolos do sinal. O primeiro cenário de estudo considera 500km de fibra SSMF, e os respetivos resultados são apresentados na Figura 4.7.

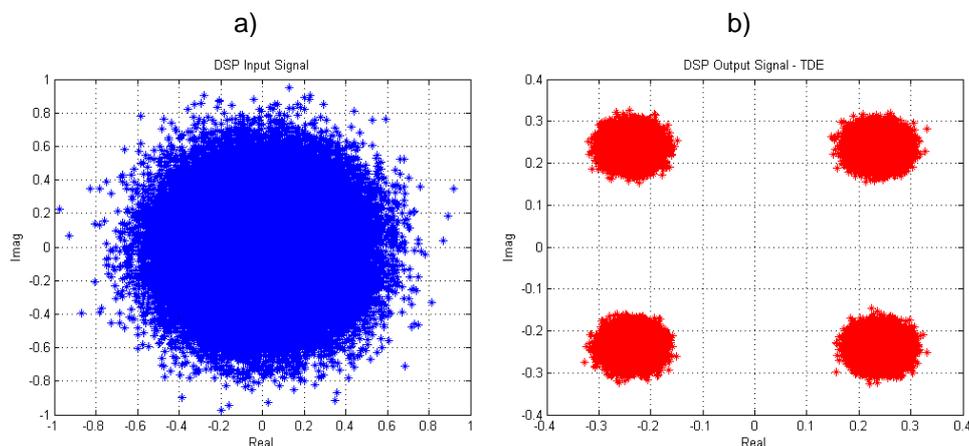


Figura 4.7: Resultados da compensação da dispersão cromática no domínio do tempo para 500km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.

Neste cenário a dispersão acumulada é $D_{ac} = 8500 \text{ ps/nm}$ ao que corresponde um alargamento do impulso de 6.8ns, este valor é bastante elevado comparando com o período de símbolo, 20ps. É assim justificado o fato do sinal recebido, Figura 4.7-a), se encontrar completamente dominado pela interferência entre símbolos. Ainda assim, o sinal é recuperado corretamente, Figura 4.7-b), apesar de ainda apresentar uma interferência entre símbolos residual. Esta é resultado da limitação da resolução do filtro de compensação e pode ser reduzida aumentando a frequência de amostragem das ADCs. Na verdade esse resultado só é possível visto que resulta num aumento do número de coeficientes e como tal uma melhor aproximação da inversa da resposta impulsional da dispersão cromática. Neste caso o filtro tem 681 coeficientes, este valor deve ser tomado em consideração e comparado com o comprimento do canal em dispersão, equação (2.37). Desse cálculo resulta que são necessários no mínimo 680 coeficientes. Verifica-se assim que o número de coeficientes utilizado é suficiente e como tal os resultados são válidos.

Atendendo agora ao caso de 1000km de fibra, a dispersão acumulada é 2 vezes superior ao caso anterior, $D_{ac} = 17000 \text{ ps/nm}$, ao que corresponde um alargamento do impulso de 13.6ns. A Figura 4.8 apresenta os resultados obtidos.

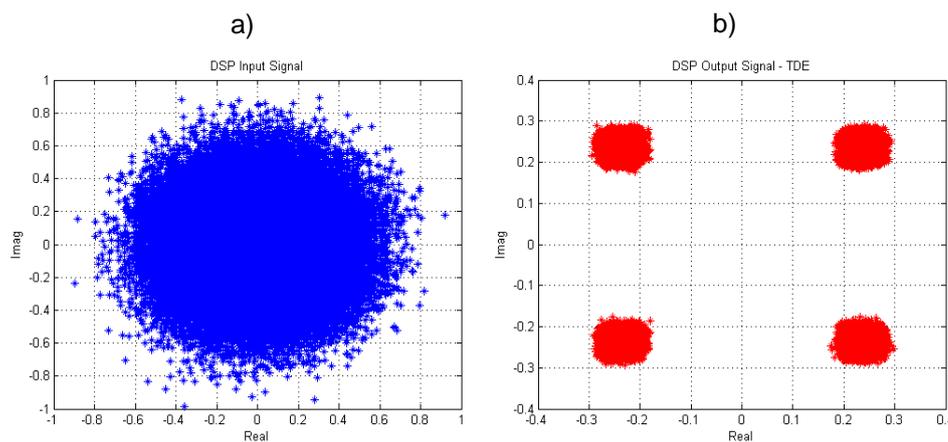


Figura 4.8: Resultados da compensação da dispersão cromática no domínio do tempo para 1000km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.

Dado o alargamento do impulso a constelação do sinal recebido apresenta-se completamente dominada pela interferência entre símbolos. Contudo, é possível verificar que o algoritmo continua a compensar corretamente a dispersão. Neste caso o filtro apresenta 1361 coeficientes, considerando a equação (2.37) são necessários 1360 coeficientes pelo que o valor obtido está correto. É possível verificar que apesar da dispersão neste caso ser bastante superior o algoritmo apresenta um melhor desempenho na compensação. Na verdade deve-se ao facto do filtro se adaptar melhor à resposta da dispersão devido ao maior número de coeficientes.

Este método não funciona corretamente para comprimentos de fibra baixos, considerando o mesmo ritmo de símbolos, uma vez que o número de coeficientes resultantes é inferior ao comprimento do canal em dispersão. Deste modo o equalizador não tem toda a informação necessária para compensar o sinal. Um aspeto importante e

que se deve ter em conta é a interferência que se verifica no sinal compensado. Esta irá limitar o desempenho dos sistemas por via deste ficar menos imune a ruído, etc. Para formatos de modulação de ordem mais elevada, por exemplo 64QAM, é necessário ter especial atenção a esta interferência visto que poderá provocar um elevado número de erros, podendo a implementação tornar-se mesmo irrealizável.

4.2.1.2 Compensação com filtro no domínio da frequência

O projeto do filtro no domínio da frequência deverá ter em conta a máxima dispersão acumulada admissível na propagação e os conceitos introduzidos na secção 2.4.1. Tal como no tópico anterior consideram-se dois casos: 500km e 1000km de fibra SSMF. Em ambos os casos o parâmetro de dispersão é 17 ps/nm.km e o comprimento de onda do sinal 1549.32 nm . O sinal binário é gerado a 100 Gbit/s , com um comprimento de 2^{14} bits , e modulado com o formato QPSK. Os modelos do transmissor e recetor são ilustrados na Figura 4.5 e Figura 4.6 respetivamente. Neste cenário, os sinais são amostrados na ADC com uma frequência que é o dobro do ritmo de símbolos do sinal. O sinal digitalizado é processado digitalmente no DSP, onde se aplicam os filtros que invertem a função de transferência da dispersão cromática no domínio da frequência. Os restantes parâmetros de configuração dos componentes são apresentados na Tabela 4.1. Tendo em conta que o Matlab[®] não apresenta grandes limitações de cálculo da FFT, esta é calculada diretamente sobre todo o sinal amostrado, não sendo aplicadas as técnicas de sobreposição introduzidas na secção “2.4.1 Compensação da dispersão Cromática”. O primeiro cenário a ser estudado é para 500km de fibra SSMF, cujos resultados encontram-se na Figura 4.9.

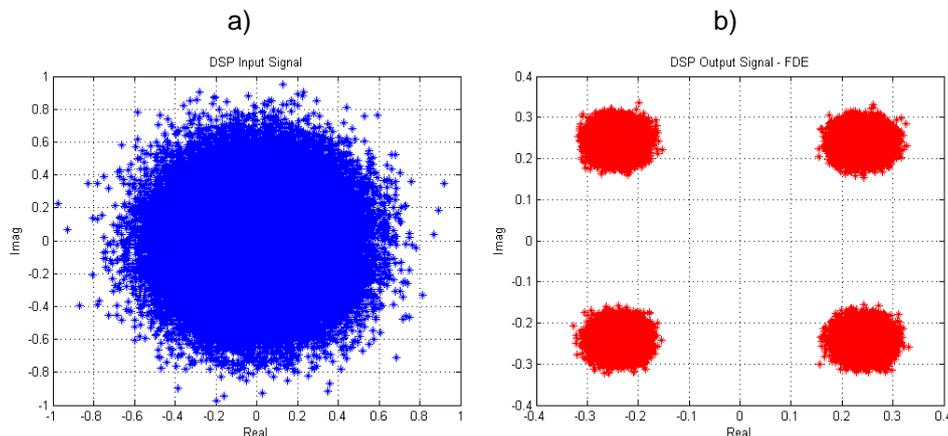


Figura 4.9: Resultados da compensação da dispersão cromática no domínio da frequência para 500km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.

A dispersão acumulada é $D_{ac} = 8500 \text{ ps/nm}$ ao que corresponde um alargamento do impulso de 6.8 ns , que é bastante elevado comparado com o período de símbolo, 20 ps . Mesmo assim o igualador é capaz de compensar a distorção observada no sinal, apresentando ainda assim alguma interferência residual. Comparando a constelação da

Figura 4.9-b) com a da Figura 4.7-b) pode verificar-se que em termos da interferência resultante da compensação o desempenho dos dois métodos é semelhante.

Atendendo agora ao segundo caso, 1000km de fibra SSMF, a dispersão acumulada é agora 2 vezes superior ao caso anterior, $D_{ac} = 17000 ps/nm$, ao que corresponde um alargamento do impulso de 13.6ns. A Figura 4.10 apresenta os resultados obtidos.

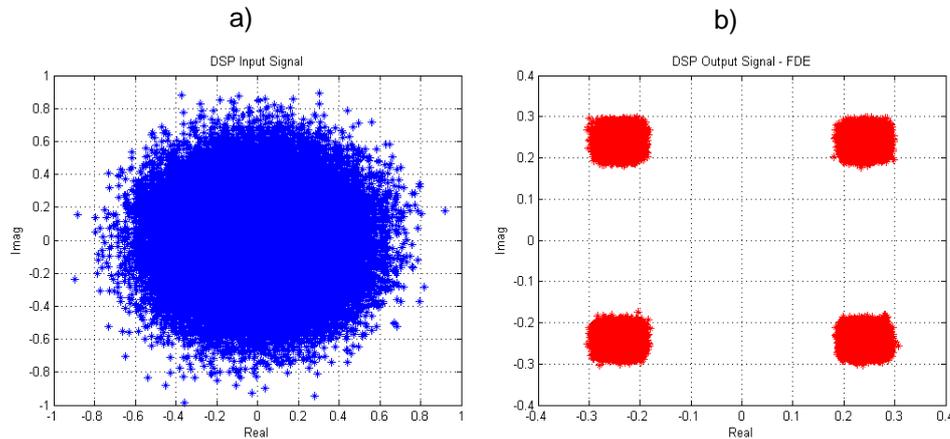


Figura 4.10: Resultados da compensação da dispersão cromática no domínio da frequência para 1000km de fibra: a) - sinal à entrada do DSP; b) - sinal à saída do DSP.

Dado o alargamento temporal do sinal a constelação do sinal recebido apresenta-se completamente dominada pela interferência entre símbolos. Contudo, é possível verificar que o algoritmo continua a compensar a dispersão com um desempenho semelhante ao algoritmo no domínio do tempo (ver Figura 4.7-b)). Apesar de este método apresentar níveis de interferência ligeiramente inferiores ao caso anterior, estes não devem ser desprezados pois podem representar um fator limitativo à aplicação de formatos de modulação de ordem mais elevada.

4.2.1.3 Desempenho dos algoritmos TDE e FDE

Um aspeto importante a estudar é o impacto destes algoritmos no sistema, e como tal, o seu desempenho. Caso estes apresentem uma penalidade elevada podem tornarem-se inviáveis à implementação. Para esse efeito mantiveram-se os modelos da Figura 4.5 e da Figura 4.6. Para o estudo considerou-se 500km de fibra SSMF, cujo parâmetro de dispersão é $17 ps/nm.km$. A sequência binária é gerada a 50Gbit/s com um comprimento de onda de 1549.32nm e modulado com o formato QPSK. As restantes configurações são apresentadas na Tabela 4.2.

Transmissor	Recetor
- DAC de 8bits	- ADC de 8bits
- Ganho de 13.98dB no amplificador	- 2 amostras/símbolo
- Laser a 193.5THz	- Laser a 193.5THz
- V_{π} de 5V	

Tabela 4.2: Parâmetros de configuração do transmissor e do recetor para testar o desempenho dos algoritmos TDE e FDE.

As configurações são idênticas às dos temas anteriores. Os filtros TDE e FDE são dimensionados de acordo com a dispersão acumulada e a frequência de amostragem. No caso do FDE é calculada a FFT sobre todo o sinal amostrado, não sendo assim aplicadas as técnicas de sobreposição. Para testar o desempenho dos algoritmos consideraram-se vários cenários de ritmos de amostragem. Estes possibilitam uma maior ou menor resolução dos filtros de equalização e como tal definem a quantidade da dispersão cromática residual resultante. Desta forma os filtros de Bessel do transmissor e do recetor são ajustados de acordo com a frequência de amostragem no recetor. Introduzindo o componente DSNR controla-se a potência do ruído no recetor obtendo-se os resultados apresentados na Figura 4.11.

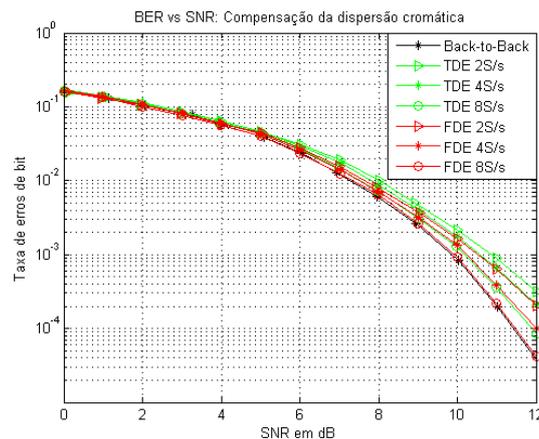


Figura 4.11: Desempenho dos algoritmos de compensação da dispersão cromática, TDE e FDE, com o ruído.

Os resultados indicam que o algoritmo FDE apresenta melhor desempenho que o TDE, sendo a penalidade introduzida inferior. Como esperado, a penalidade introduzida depende da resolução dos filtros aplicados, sendo que, para o algoritmo FDE com 8 amostras por símbolo não se regista qualquer penalidade relativamente às simulações sem fibra. Já no caso do TDE com 8 amostras por símbolo ainda se regista uma penalidade que na gama representada pode tomar valores de aproximadamente 0.5dB para uma BER de 10^{-4} . Contudo, esta comparação direta não é correta pois os dois filtros apresentam dimensões diferentes, ainda assim os resultados estão de acordo com [43, 62, 63]. Para os dois igualadores, verifica-se que para baixos valores de SNR não é introduzida penalidade. Na verdade sendo a variância do ruído nessa zona bastante superior à resultante da dispersão residual, esta torna-se desprezável. Deve notar-se que

o limite de *Nyquist* define que, para preservar a informação do sinal basta amostrar o sinal ao dobro da sua componente de frequência mais elevada. Na verdade verifica-se essa relação, sendo ainda assim registada uma penalidade que poderá ser superior a 1dB. Estas penalidades são semelhantes às registadas em [43, 63-65] e devem-se à interferência entre símbolos verificada no sinal após a compensação. O seu efeito será tão superior quanto maior for a SNR e poderá ser reduzido aumentando a frequência de amostragem das ADCs (ver Figura 4.11). Essa melhoria é possível de se observar pois o aumento da resolução dos filtros possibilita uma maior aproximação da resposta impulsional e como tal a redução da interferência residual. Deve notar-se que em implementações práticas o filtro no domínio da frequência pode apresentar algumas limitações no desempenho devido ao tempo de computação necessário para o cálculo da FFT e respetiva IFFT. Da mesma forma, o filtro temporal apenas será implementável na prática se o número de coeficientes necessário não for muito elevado, pois este está limitado pelos recursos existentes nos DSPs.

4.2.2 Compensação da dispersão dos modos de polarização

Nesta secção pretende-se implementar e testar o algoritmo de equalização da PMD, considerando que as condições de teste são ideais, isto é, ausência de ruído. O algoritmo estudado é o CMA e os conceitos relacionados com o seu projeto encontram-se na secção 2.4.2. O esquema de implementação do compensador é apresentado na Figura 4.12.

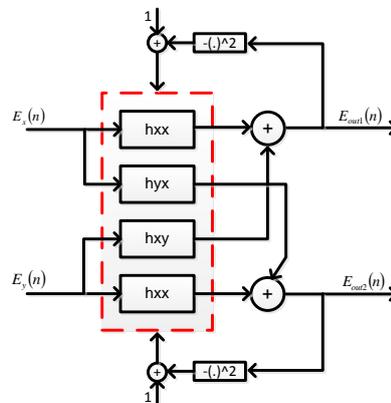


Figura 4.12: Modelo de implementação do algoritmo CMA.

Por cada símbolo que é processado é calculado um sinal de erro relativamente ao módulo unitário, posteriormente de acordo com este sinal e com os valores residentes em memória do filtro FIR são implementadas as equações que permitem, em cada iteração, atualizar os coeficientes dos filtros. O algoritmo converge quando da filtragem resulta um sinal de módulo unitário. Isto significa que, deixam de ocorrer oscilações no sinal de erro e portanto a função de custo dos coeficientes anula-se, representando os coeficientes dos filtros a função de transferência inversa do canal de transmissão. Estes coeficientes são todos inicializados a zero exceto os centrais de h_{xx} e h_{yy} que são inicializados a um.

Para testar o algoritmo os modelos do transmissor, canal e recetor são apresentados na Figura 4.13, na Figura 4.14 e na Figura 4.15 respetivamente.

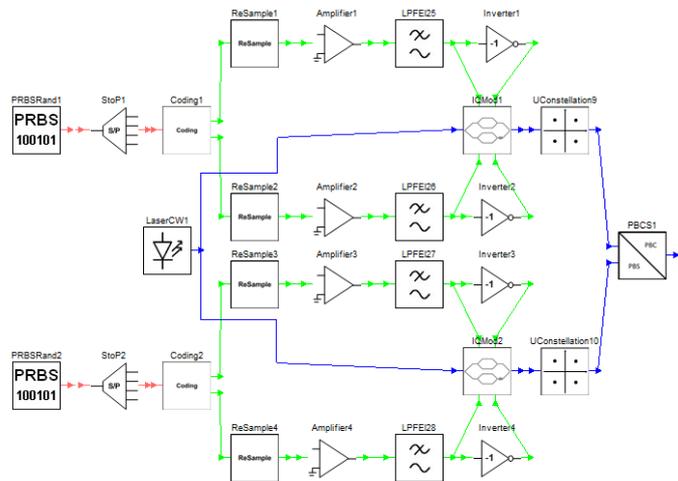


Figura 4.13: Modelo do transmissor ótico com multiplexagem na polarização implementado no OSIP.

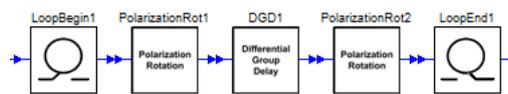


Figura 4.14: Modelo em OSIP do canal ótico para simular os efeitos da PMD.

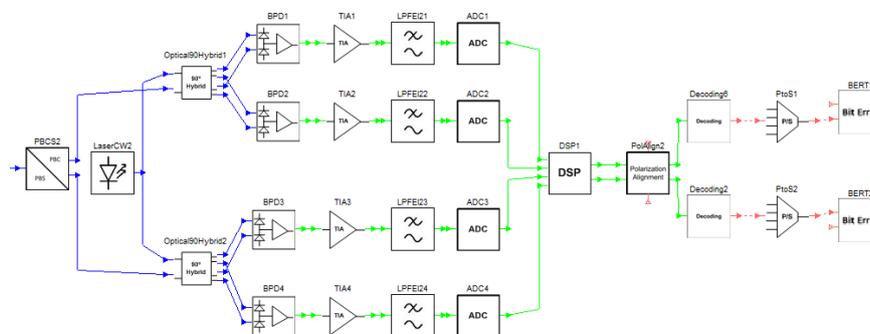


Figura 4.15: Modelo do recetor ótico com diversidade de polarização implementado no OSIP.

O sinal binário é gerado a um ritmo de 100Gbit/s, com um comprimento de 2^{17} bits, e é modulado com o formato QPSK. Deve notar-se que neste caso como se tem o sinal multiplexado nas duas componentes de polarização, sistema DP-QPSK, o ritmo de cada PRBS é de 50Gbit/s. Estes sinais são modulados em circuitos independentes mas com o mesmo laser e combinados posteriormente, utilizando para o efeito um PBC. Para simular os efeitos da PMD é utilizado o modelo de primeira ordem, sobre o qual são realizados vários ciclos para tornar as simulações mais realistas. A rotação da polarização é realizada aleatoriamente e o valor da DGD é 1ps, valor típico para 100km de fibra SSMF ($0.1ps\sqrt{km}^{-1}$ [17]). Considera-se que os componentes são ideais, logo

não introduzem DGD, e como tal o atraso das componentes de polarização é apenas resultado da propagação. Os restantes parâmetros de configuração do transmissor e recetor encontram-se na Tabela 4.3.

Transmissor	Recetor
- Filtro de Bessel a 25GHz	- Filtro de Bessel a 25GHz
- DAC de 8bits	- ADC de 8bits
- Ganho de 13.98dB no amplificador	- 2 amostras/símbolo
- Laser a 193.5THz	- Laser a 193.5THz
- V_{π} de 5V	

Tabela 4.3: Parâmetros de configuração do transmissor e recetor para testar o algoritmo de compensação da PMD.

O sinal ao ser adquirido no recetor é dividido nas suas componentes de polarização, as quais são desmoduladas em circuitos distintos. Os sinais resultantes são misturados com um oscilador local, resultando a sua representação em termos das componentes em fase e quadratura, e amostrados nas ADCs com uma frequência que é o dobro do ritmo de símbolos. Os filtros elétricos são ajustados para o ritmo de símbolos por polarização e os sinais, após serem amostrados, são processados por um DSP. Adicionalmente é utilizando um componente para permitir alinhar as polarizações à saída do DSP e desta forma verificar se o algoritmo está a recuperar corretamente a informação. Para este cenário é simulada a PMD de primeira ordem. Os filtros aplicados têm três coeficientes e um parâmetro de convergência de 10^{-3} . Na verdade realizando os cálculos conforme a equação (2.45) bastava utilizar um coeficiente por filtro. Os resultados obtidos encontram-se na Figura 4.16 e na Figura 4.17.

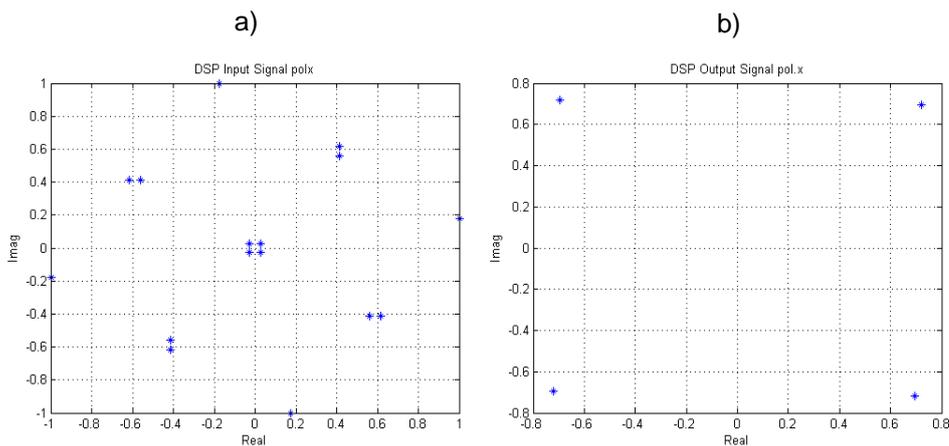


Figura 4.16: Constelações obtidas para a polarização x com uma DGD de 1ps e PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.

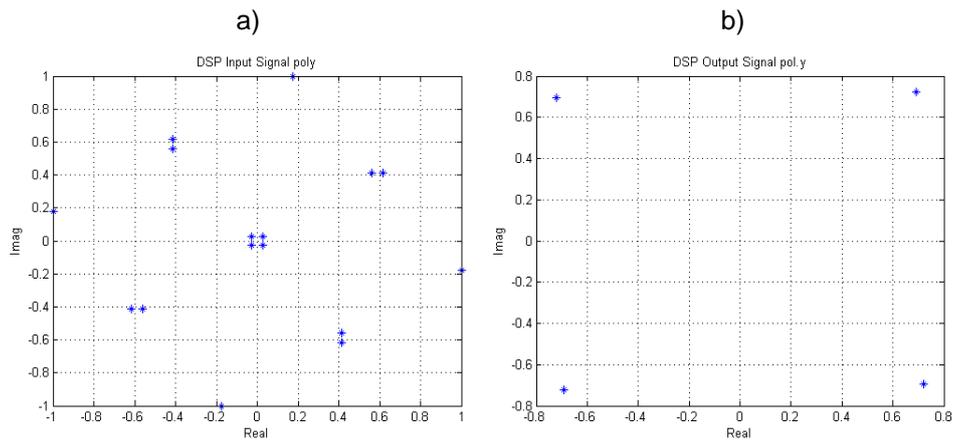


Figura 4.17: Constelações obtidas para a polarização y com uma DGD de 1ps e PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.

Na Figura 4.16-a) e na Figura 4.17-a) é possível verificar que o sinal registado após simular o troço com PMD apresenta 16 pontos da constelação, os quais são resultados das combinações com a matriz de Jones. As constelações na Figura 4.16-b) e na Figura 4.17-b) são obtidas após algumas amostras para garantir que o algoritmo já atingiu a convergência aquando da visualização da constelação. Nas sequências binárias decodificadas não se verificou nenhum erro. Assim o algoritmo é capaz de compensar a PMD de primeira ordem para um comprimento de fibra de 100km. Este caso é ainda assim bastante redutor. Para tornar o cenário de simulação mais realista realizaram-se 100 ciclos sobre no modelo de simulação da PMD. Os resultados obtidos encontram-se na Figura 4.18 e na Figura 4.19.

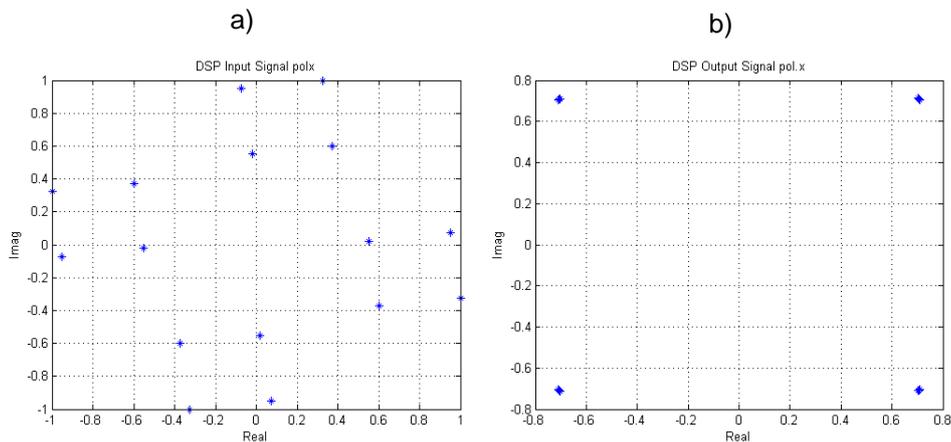


Figura 4.18: Constelações obtidas para a polarização x com uma DGD de 1ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.

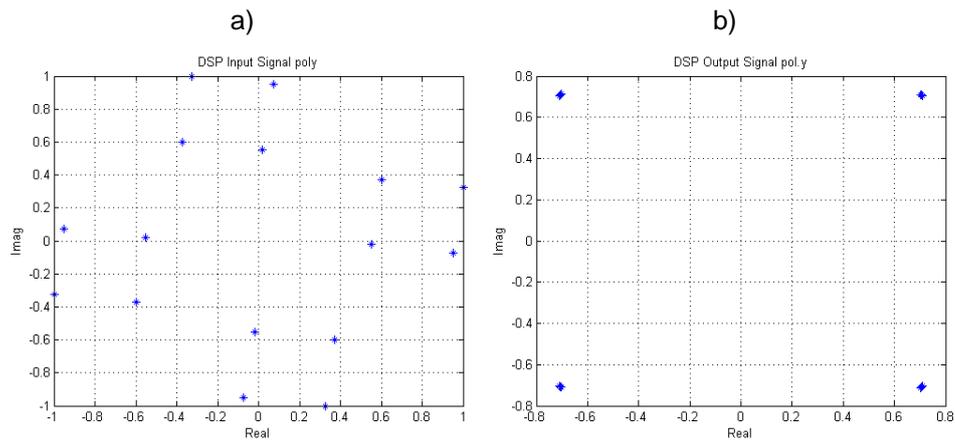


Figura 4.19: Constelações obtidas para a polarização y com uma DGD de 1ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.

Mesmo realizando 100 ciclos o algoritmo é capaz de compensar a PMD, apresentando um tempo de convergência de aproximadamente 1us. Desta forma o algoritmo é capaz de acompanhar as flutuações da PMD que tipicamente ocorrem na ordem dos milissegundos. Para testar melhor a robustez do algoritmo realiza-se uma simulação com um DGD mais elevado, 3.2ps, o que corresponde a aproximadamente 1000km de fibra. Manteve-se os 100 ciclos sobre a PMD de primeira ordem e o número de coeficientes do filtro. Os resultados obtidos encontram-se na Figura 4.20 e na Figura 4.21.

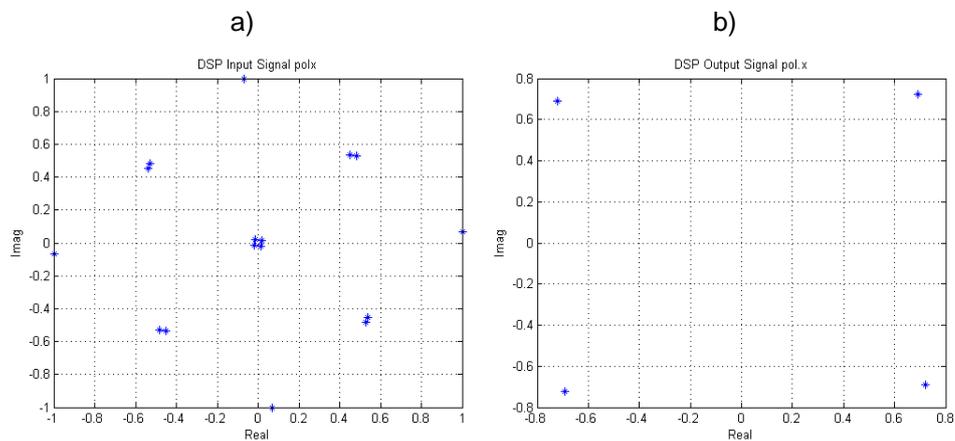


Figura 4.20: Constelações obtidas para a polarização x com uma DGD de 3.2ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.

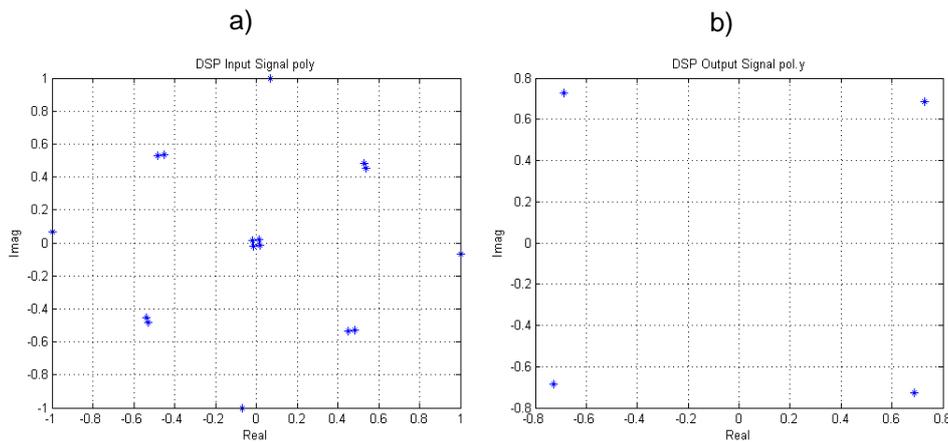


Figura 4.21: Constelações obtidas para a polarização y com uma DGD de 3.2ps e 100 ciclos de PMD de primeira ordem: a) – constelação à entrada da DSP; b) – constelação após a compensação no DSP.

Os resultados vão de encontro com as expectativas. O algoritmo consegue equalizar os efeitos da PMD proporcionando desta forma uma maior robustez ao recetor. Deve notar-se que o tempo de convergência vai depender da ordem da PMD mas também de outros efeitos presentes no sinal de que são exemplo a dispersão cromática e o ruído. Para uma maior eficiência a dispersão deve ser compensada previamente, tendo assim este filtro de lidar apenas com valores residuais de dispersão e com o ruído.

4.2.2.1 Desempenho do algoritmo CMA

O desempenho do algoritmo CMA com o ruído é um fator muito importante e que pode definir a viabilidade da sua implementação. Para estudar o seu comportamento mantiveram-se os modelos da Figura 4.13 e da Figura 4.15. Para além disso considerou-se um troço de fibra SSMF com 500km de comprimento, ao qual corresponde aproximadamente 2.236ps de DGD, e PMD de primeira ordem. Tal como no tópico anterior, considera-se que apenas a propagação na fibra introduz atraso entre as componentes de polarização. Cada PRBS gera a informação binária a 50Gbit/s, sendo essa informação codificada e modulada no formato QPSK com um laser cujo comprimento de onda é de 1549.32nm, resultando o DP-QPSK a 100Gbit/s. O sinal adquirido no recetor é amostrado e convertido para o domínio digital utilizando duas amostras por símbolo, sendo o algoritmo de igualação da PMD aplicado neste domínio. De seguida são decididos quais os símbolos recebidos e a sequência binária é decodificada. É também utilizado um componente que permite alinhar as polarizações na entrada dos circuitos de decodificação. Este torna-se necessário, visto que, o algoritmo de compensação da PMD não garante para qual das suas saídas converge cada uma das polarizações. As restantes configurações são apresentadas na Tabela 4.3. Realizando os cálculos segundo a equação (2.45), para compensar a PMD basta que os filtros sejam de um coeficiente. Para analisar o desempenho supôs-se filtros com um, dois e cinco coeficientes e variou-se a potência do ruído no recetor, utilizando o componente DSNR. Os resultados obtidos encontram-se na Figura 4.22.

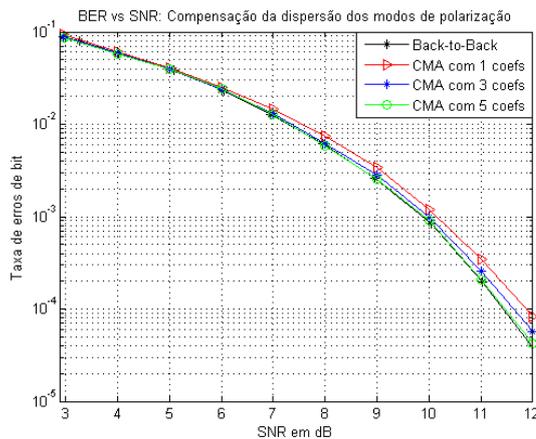


Figura 4.22: Desempenho do algoritmo CMA com o ruído para diferentes números de coeficientes dos filtros.

A Figura 4.22 mostra que utilizando apenas um coeficiente é possível compensar os efeitos da PMD. Deve notar-se ainda que é introduzida uma penalidade que, na gama representada pode atingir o valor máximo de 0.5dB. Esta penalidade deve-se à resolução da inversão da resposta impulsional do canal e é semelhante à observada em [66]. Como se pode verificar na Figura 4.22 o aumento do número de coeficientes permite aumentar a resolução dos filtros e como tal melhorar a aproximação da função de transferência do canal de transmissão, resultando a redução da penalidade. Para o caso de 5 coeficientes não é registada nenhuma penalidade estando os resultados sobrepostos aos obtidos sem canal de transmissão. De qualquer forma, o baixo valor da penalidade e a sua capacidade de manter o desempenho com o ruído, isto é, não entrar em instabilização, faz com que o algoritmo seja uma boa opção para implementações práticas da equalização da PMD.

4.2.3 Estimação e correção de fase

Os algoritmos de estimação e correção de fase foram apresentados na secção 2.4.3 e correspondem ao algoritmo de *Viterbi* e *Viterbi* (V&V) com e sem linha de realimentação. Tendo em vista a implementação dos algoritmos no ambiente de simulação os esquemas utilizados são apresentados na Figura 4.23.

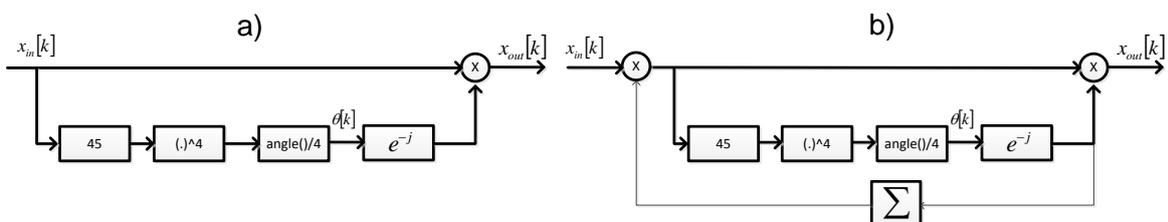


Figura 4.23: Modelos de implementação dos algoritmos de correção de fase: a) - V&V; b) - V&V com realimentação.

Os modelos para os dois algoritmos são idênticos, sendo que, a única diferença reside na linha de realimentação. Esta corresponde à fase cumulativa das amostras anteriores e permite seguir de uma forma mais robusta as variações bruscas de fase. Comparando estes modelos com os apresentados na Figura 2.16 e na Figura 2.17 pode notar-se uma diferença, agora o sinal é rodado de 45° antes de ser elevado à quarta ordem. Esta rotação permite eliminar a incerteza de π que se obtém na medida direta, medindo-se apenas o erro de fase. Tendo em conta que o algoritmo apresenta o melhor desempenho para o formato QPSK este vai ser adotado para as simulações a realizar. Algumas das vantagens e desvantagens de cada algoritmo já foram referidas na secção “2.4.3 Estimação e correção de fase” e pretende-se agora observá-las no ambiente de simulação. Os modelos do transmissor e do recetor adotados para as simulações são apresentados na Figura 4.24 e na Figura 4.25.

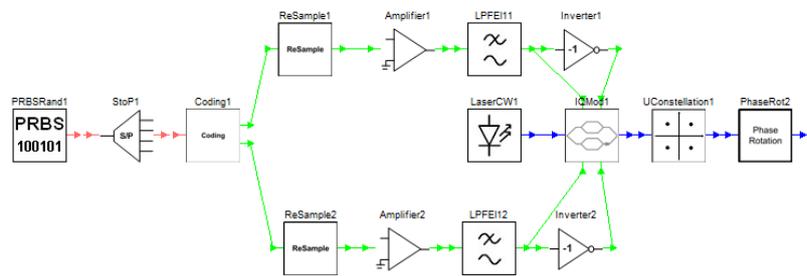


Figura 4.24: Modelo no OSIP do transmissor para teste dos algoritmos de correção de fase.

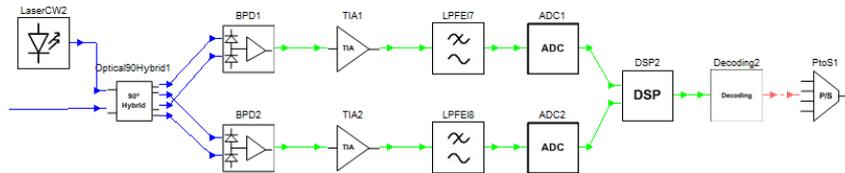


Figura 4.25: Modelo no OSIP do recetor para teste dos algoritmos de correção de fase.

Pretende-se testar os algoritmos para rotações de fase constantes em todos os símbolos e em condições ideais. Para esse fim, introduziu-se um componente que provoca essas rotações de fase, Figura 4.24. A compensação é realizada diretamente no ambiente de simulação utilizando para o efeito o componente DSP. Desta forma utilizando também o componente BERT é possível detetar erros na correção da fase do sinal. Estes podem levar à rotação de fase da constelação de um múltiplo inteiro da relação de fase entre os símbolos. O sinal binário para teste é gerado a 100Gbit/s com um comprimento de 2^{15} bits e é modulado com o formato QPSK. Depois de aplicada a rotação de fase, o sinal é adquirido no recetor e misturado com o oscilador local. Do respetivo batimento, resulta a representação do sinal nas suas componentes em fase e quadratura, as quais são amostradas, a duas amostras por símbolo, e processadas digitalmente. Posteriormente, a sequência binária é decodificada. A Tabela 4.4 apresenta as restantes configurações do transmissor e do recetor.

Transmissor	Recetor
- Filtro de Bessel a 50GHz	- Filtro de Bessel a 50GHz
- DAC de 8bits	- ADC de 8bits
- Ganho de 13.98dB no amplificador	- 2 amostras/símbolo
- Laser a 193.5THz	- Laser a 193.5THz
- V_{π} de 5V	

Tabela 4.4: Parâmetros de configuração do transmissor e do recetor para o teste dos algoritmos de correção de fase.

Os filtros elétricos utilizados no transmissor e no recetor são ajustados para o ritmo de símbolos do sinal. Os testes foram realizados para rotações de fase de +/- 30°, de 45° e de 60°. Os resultados obtidos são apresentados na Figura 4.26. Nesta as constelações à esquerda representam a entrada da DSP e à direita a saída. Visto que neste caso o desempenho dos dois algoritmos é idêntico apresenta-se apenas uma figura que representa os dois casos. É possível verificar que em todos os casos a fase da constelação é detetada e recuperada. Contudo no último ocorrem erros de deteção para o caso do algoritmo V&V. Este resultado deve-se ao facto do ângulo de rotação exceder o limiar de 45°, ocorrendo saltos de fase. Uma vez que a constelação mantém o seu posicionamento não seria possível detetar esta rotação sem que fosse utilizado o medidor de erros. Como seria de esperar o circuito com realimentação corrige corretamente o sinal. Outro cenário possível, neste caso para testar a abordagem com realimentação, é provocar rotações de fase variantes no tempo. Para tal considerou-se que a fase variava instantaneamente entre amostras. O algoritmo V&V não é considerado neste cenário visto que é de esperar que apresente um mau comportamento, a partir do momento que a rotação de fase excede os 45° o algoritmo começa a cometer erros elevados. As variações de fase introduzidas apresentam um intervalo de variação de 0 a π com transições instantâneas entre amostras de $\pi/6$. Os resultados obtidos são apresentados na Figura 4.27.

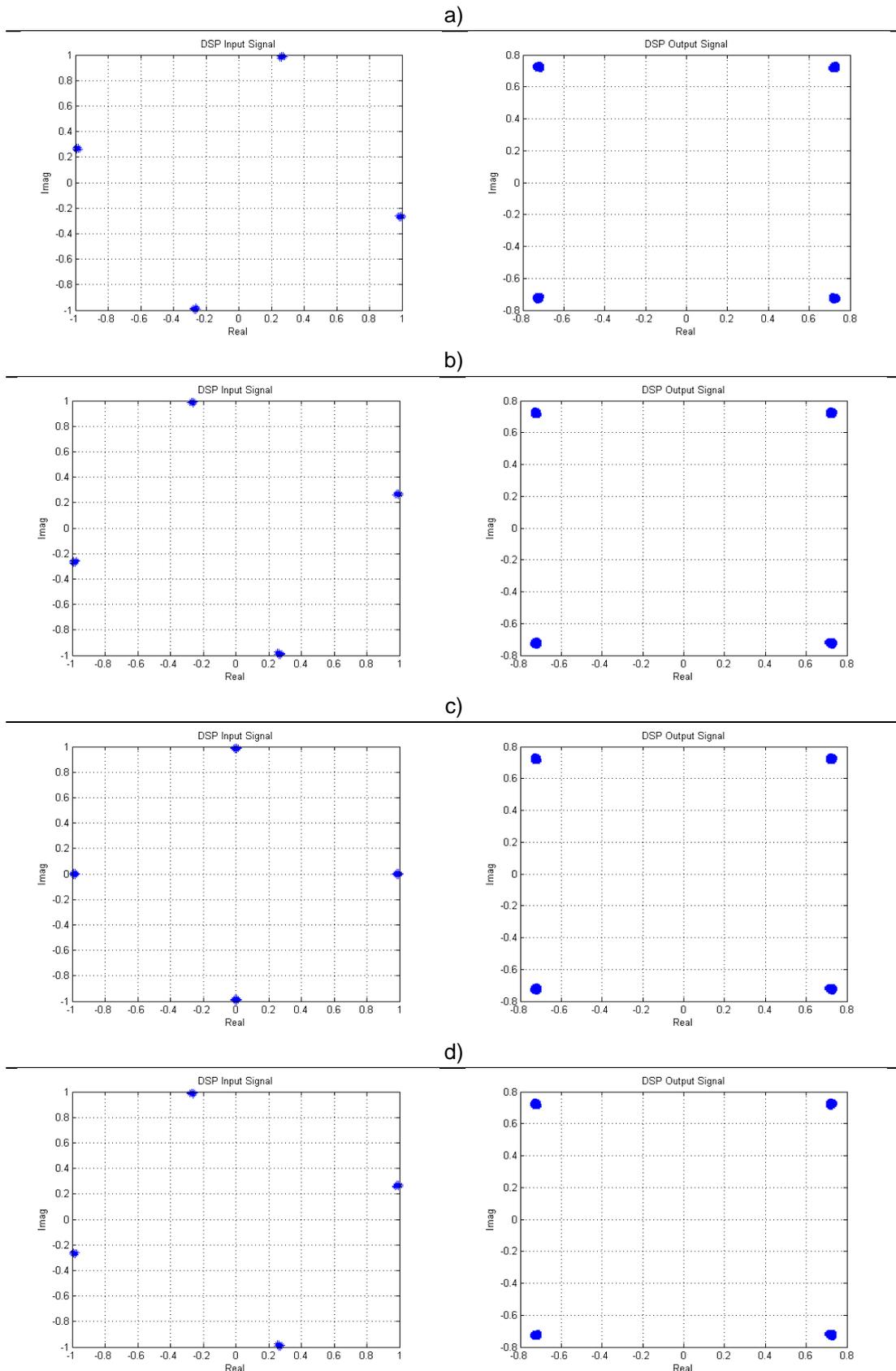


Figura 4.26: Resultados obtidos para os dois circuitos com rotações de fase: a) -30° ; b) -30° ; c) -45° ; d) -60° .

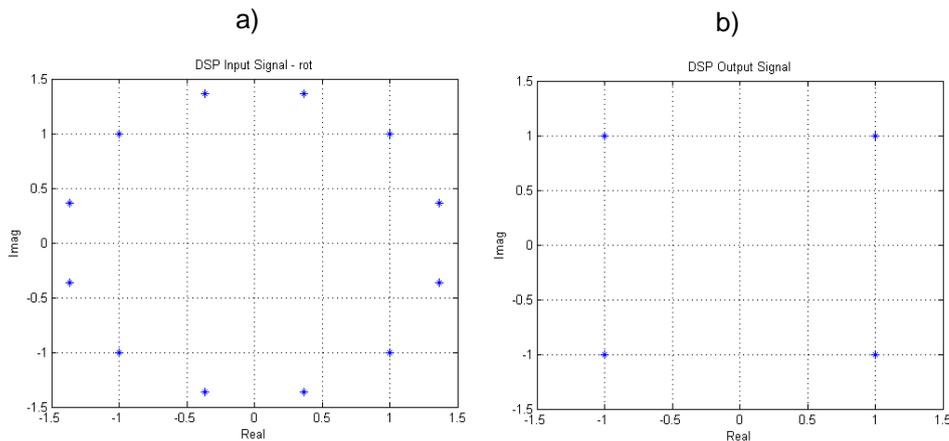


Figura 4.27: Resultados obtidos com o algoritmo com realimentação para uma variação temporal da rotação de fase: a) – sobreposição temporal do sinal recebido na DSP; b) – sinal compensado.

Na Figura 4.27-a) é apresentada a constelação que corresponde à sobreposição temporal de todos os sinais resultantes das rotações de fase realizadas. Verifica-se que o algoritmo é capaz de estimar corretamente a fase e acompanhar as suas variações recuperando corretamente a informação do sinal. Deve notar-se que se a rotação de fase da primeira amostra for nula, na segunda é $\frac{\pi}{6}$ e na terceira $\frac{\pi}{3}$ o que já seria problemático para o algoritmo V&V. Este estima para cada amostra o erro absoluto, sempre que a rotação de fase excede os 45° o quadrante de referência da constelação é alterado resultando erros na correção da fase e como consequência saltos de fase. A abordagem com realimentação continua a apresentar um bom desempenho não sendo observados erros. Na verdade, desde que as transições instantâneas não excedam os 45° o algoritmo conseguirá sempre acompanhar as rotações do sinal sem que provoque nenhum erro no sinal compensado.

4.2.3.1 Desempenho dos algoritmos de correção de fase

Conceptualmente, o algoritmo de correção de fase deverá conseguir acompanhar as variações de fase do sinal independentemente do cenário em que está inserido. Na verdade a presença de ruído do canal e de fase dos lasers pode influenciar o seu desempenho levando-o a cometer erros. O impacto destas fontes de erro depende do grau de paralelização dos circuitos e da frequência de amostragem no recetor. Uma boa tolerância ao ruído aditivo branco gaussiano exige um grau de paralelização elevado, enquanto que, para o ruído de fase exige um baixo grau de paralelização. Assim, estas fontes de erro apresentam exigências conflituosas, pelo que a melhor solução resulta de um compromisso entre estes dois fatores. Na verdade, devido a limitações de *hardware* as soluções adotadas nem sempre resultam deste compromisso. Para testar o desempenho dos algoritmos aos efeitos do ruído, consideraram-se aqui vários cenários. No primeiro caso de estudo define-se o processamento de 16 amostras em simultâneo.

Na verdade corresponde a ter 16 circuitos idênticos a processar em paralelo. Os modelos dos algoritmos implementados encontram-se na Figura 4.28.

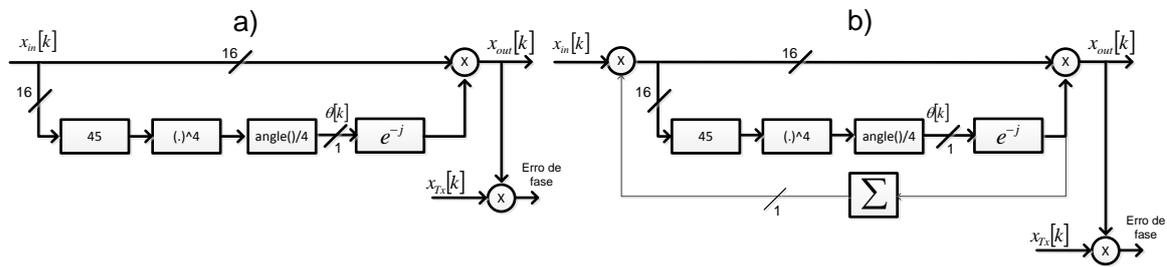


Figura 4.28: Modelo dos algoritmos de fase para testar o seu desempenho ao ruído do canal e de fase dos lasers: a) – V&V; b) – V&V com realimentação.

Após a correção de fase o sinal resultante é comparado com o sinal do transmissor sendo assim possível detetar o erro de fase cometido por cada algoritmo. Deve notar-se que nos dois algoritmos o valor da fase que compensa o sinal corresponde à média das fases detetadas nas amostras em paralelo e que no caso do algoritmo da Figura 4.28-b) apenas este valor é realimentado. O primeiro teste a realizar é com ruído branco gaussiano. Pretende-se variar a potência do ruído de forma a medir as relações entre a BER e a SNR, comparando depois os resultados com as relações teóricas. Para esse fim a sequência de dados de teste é gerada a 100Gbit/s, com um comprimento de 2^{19} bits, e modulada com o formato QPSK. Os modelos do transmissor e do recetor são iguais aos da Figura 4.24 e da Figura 4.25 respetivamente e as restantes configurações à Tabela 4.4. O sinal ótico é adquirido no recetor e convertido para o domínio elétrico onde é amostrado nas ADCs e processado digitalmente, sendo decodificadas as sequências binárias. A DSP implementa os dois algoritmos de estimação e correção de fase, permitindo assim comparar o seu desempenho. Os resultados obtidos encontram-se na Figura 4.29.

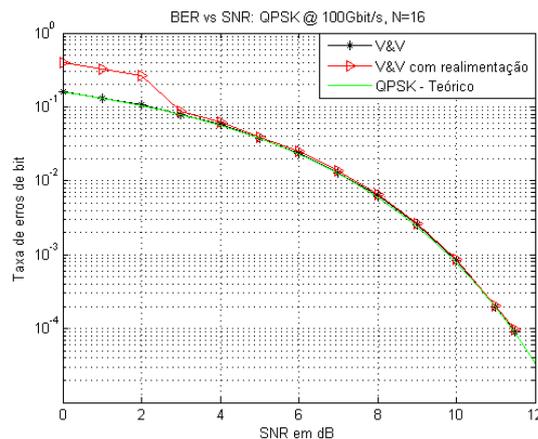


Figura 4.29: Relação entre a BER e a SNR teórica e para os algoritmos V&V e V&V com realimentação e processamento paralelo de 16 amostras.

Os dois circuitos apresentam desempenhos semelhantes e seguem os resultados teóricos. O algoritmo com realimentação para um valor de SNR inferior a 2.5dB começa a instabilizar-se provocando saltos de fase, aumentando consequentemente a

probabilidade de erro. Este comportamento é justificado pelo número reduzido de amostras processadas em paralelo, como tal o ruído tem uma grande influência na linha de realimentação. No segundo teste pretende-se variar a largura de linha do laser e medir a BER e o erro de fase cometido, para uma SNR constante de 10dB. As condições de simulação são as mesmas do caso anterior. Os resultados obtidos encontram-se na Figura 4.30.

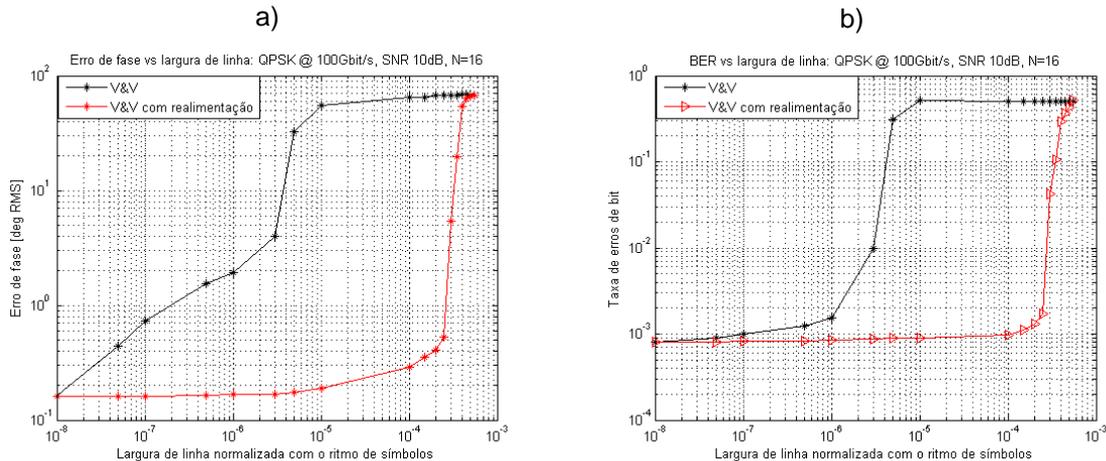


Figura 4.30: Resultados obtidos para os dois algoritmos com uma SNR de 10dB, processamento paralelo de 16 amostras e variando a largura de linha do laser: a) - erro de fase; b) - BER.

Na Figura 4.30 a largura de linha é normalizada ao ritmo de símbolos do sinal. Facilmente se verifica que o desempenho do circuito V&V com realimentação é bastante superior ao V&V. O circuito com realimentação apresenta um bom desempenho para uma probabilidade de erro inferior a 10^{-3} , aproximadamente, para uma SNR de 10dB, até próximo dos 5MHz de largura de linha em cada um dos lasers (transmissor e recetor), enquanto que o V&V apresenta uma tolerância de aproximadamente 5kHz. Este valor é muito reduzido visto que os lasers têm tipicamente larguras que vão dos 100kHz aos 10MHz. Já a largura de linha tolerada pelo algoritmo V&V com realimentação é bastante boa. Em termos de erro de fase para uma largura de linha de aproximadamente 12MHz o algoritmo com realimentação apresenta um erro médio inferior a um grau, enquanto que para o mesmo valor o V&V apresenta um erro de fase próximo dos 50° . Os resultados para o algoritmo V&V com realimentação seguem a mesma característica e apresentam resultados de BER semelhantes aos observados em [67]. Esta diferença no desempenho dos algoritmos já seria de esperar visto que o algoritmo original de *Viterbi* mede a fase absoluta em cada instante, logo a fase cumulativa resultante do ruído de fase provoca um elevado número de erros. Deve notar-se que o número de amostras em paralelo utilizado para este cenário, para a frequência de amostragem considerada, não é implementável na prática, uma vez que exigia um DSP com uma velocidade de processamento de aproximadamente 3GHz. Considerando agora um cenário implementável (DSP a aproximadamente 190MHz), procede-se ao mesmo estudo para 256 amostras em paralelo. O esquema de implementação dos algoritmos encontra-se na Figura 4.31.

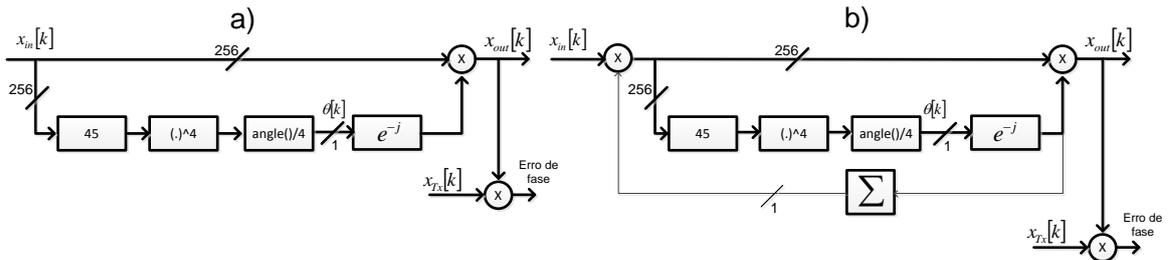


Figura 4.31: Modelo de implementação dos algoritmos V&V e V&V com realimentação considerando o processamento de 256 amostras em paralelo.

Os esquemas são idênticos aos da Figura 4.28, alterando-se apenas o número de amostras. O transmissor e o recetor também são os mesmos assim como as suas configurações. Os resultados obtidos para o comportamento com o ruído do canal encontram-se na Figura 4.32.

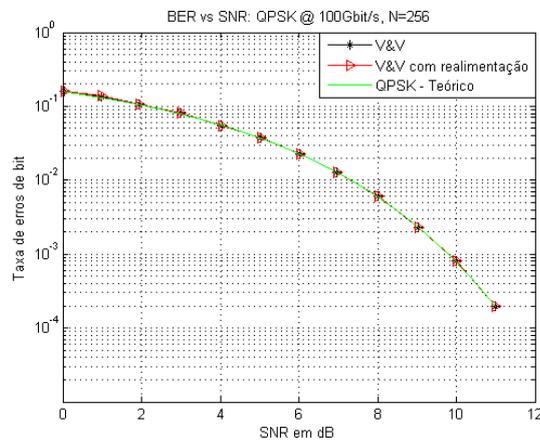


Figura 4.32: Relação entre a BER e a SNR teórica e para os algoritmos V&V e V&V com realimentação e processamento paralelo de 256 amostras.

Neste cenário os algoritmos continuam a ter um bom desempenho com o ruído branco gaussiano, seguindo os resultados teóricos. Como referido no exemplo anterior o aumento do número de amostras torna os algoritmos mais robustos ao ruído branco gaussiano. Comparando a Figura 4.29 com a Figura 4.32, pode-se verificar que os resultados estão agora mais próximos dos esperados teoricamente e que o algoritmo com realimentação já não se instabiliza. De igual forma registaram-se os resultados para a variação da largura de linha do laser, considerando uma SNR constante de 10dB, os quais são apresentados na Figura 4.33.

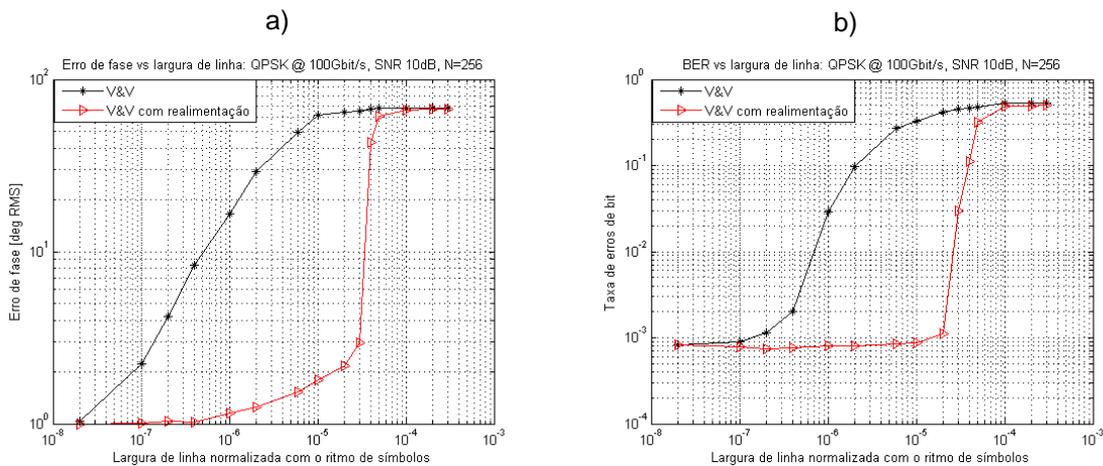


Figura 4.33: Resultados obtidos para os dois algoritmos com uma SNR de 10dB, processamento paralelo de 256 amostras e variando a largura de linha do laser: a) - erro de fase; b) - BER.

Verifica-se que a tolerância do algoritmo V&V com realimentação se reduz bastante para uma BER de 10^{-3} . Esta passa a ser aproximadamente 1MHz por laser, enquanto que o V&V continua a ser aproximadamente 5kHz. Verifica-se que o desempenho do V&V não sofre uma grande variação da máxima tolerância para uma BER de 10^{-3} , contrariamente ao algoritmo com realimentação. Ainda assim, para valores de largura de linha superiores a 5kHz o algoritmo V&V apresenta um crescimento da BER bastante mais rápido do que o registado para o caso anterior. O erro de fase dos algoritmos também é agora superior, apresentando o algoritmo com realimentação um erro inferior a aproximadamente 3° até uma largura de linha por laser de 1.5MHz. Já o V&V para a mesma largura de linha apresenta um erro de fase de aproximadamente 65° . Constata-se assim que uma maior tolerância a ruído branco gaussiano implica uma redução da eficiência do algoritmo com o ruído de fase e vice-versa. O estudo da tolerância dos algoritmos ao ruído de fase com o grau de paralelização é assim um fator importante de estudo. Considerando apenas o ruído de fase a Figura 4.34 apresenta a máxima largura de linha tolerada pelos dois circuitos em função do grau de paralelismo para um erro médio de fase de um grau.

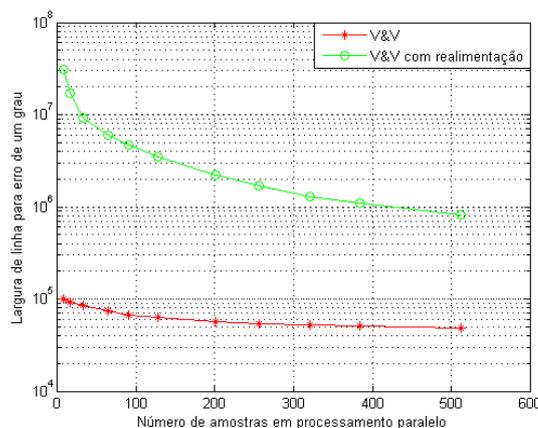


Figura 4.34: Largura de linha tolerada pelos algoritmos de estimação de fase para diferentes valores de paralelismo.

Comparando estes resultados com os anteriores verifica-se que o ruído branco gaussiano influencia bastante os erros cometidos. Atendendo ao exemplo com 16 amostras em processamento paralelo verifica-se um decréscimo de aproximadamente 7MHz na largura de linha tolerada por cada laser, no caso com realimentação, e 86kHz no caso do V&V. Estes valores são bastante significativos. A Figura 4.34 também mostra claramente a influência do grau de paralelismo do circuito no acompanhamento do ruído de fase. Verifica-se que no caso do algoritmo com realimentação aumentando 32 vezes o grau de paralelismo a máxima largura de linha tolerada vem reduzida de aproximadamente 21 vezes, já o algoritmo V&V original vê a sua tolerância reduzida de aproximadamente duas vezes. O algoritmo V&V apresenta assim uma capacidade de se adaptar a cenários de maior paralelismo, ainda assim o desempenho do algoritmo com realimentação é bastante superior. Dados os resultados o algoritmo V&V com linha de realimentação apresenta-se como uma boa solução prática. Contudo deve notar-se que este pode apresentar problemas de implementação devido ao atraso provocado na linha de realimentação. Caso este seja elevado o desempenho do compensador reduz-se.

4.3 WDM

A multiplexagem de informação por divisão do comprimento de onda assume um papel fundamental nos sistemas óticos atuais. A possibilidade de modular canais com diferentes características possibilita o aumento da largura de banda efetiva assim como agrupar tráfego de diferentes clientes. Nesta secção, são estudados dois casos de sistemas WDM, no primeiro considera-se dois canais modulados com o formato QPSK e com apenas uma componente de polarização. No segundo caso, os canais são modulados nas duas componentes de polarização com o formato QPSK. Pretende-se estudar os sistemas sem canal de transmissão, logo em condições ideais, e com canal de transmissão, sendo neste caso necessário aplicar algoritmos de igualização dos efeitos lineares que ocorrem na propagação no canal, CD e PMD.

4.3.1 Dois canais SP-QPSK a 50Gbit/s

Para estudar este sistema a informação binária para cada um dos canais é gerada a 50Gbit/s e modulada com o formato QPSK cujo comprimento da sequência é ajustado de acordo com a probabilidade de erro esperada (teoricamente) para o caso ideal. Considera-se que os canais estão espaçados de 100GHz. A frequência dos lasers é de 193.5THz e de 193.6THz, sendo estes configurados com uma largura de linha de 100kHz. Os modelos do transmissor, do canal e do recetor encontram-se na Figura 4.35, na Figura 4.36 e na Figura 4.37 respetivamente.

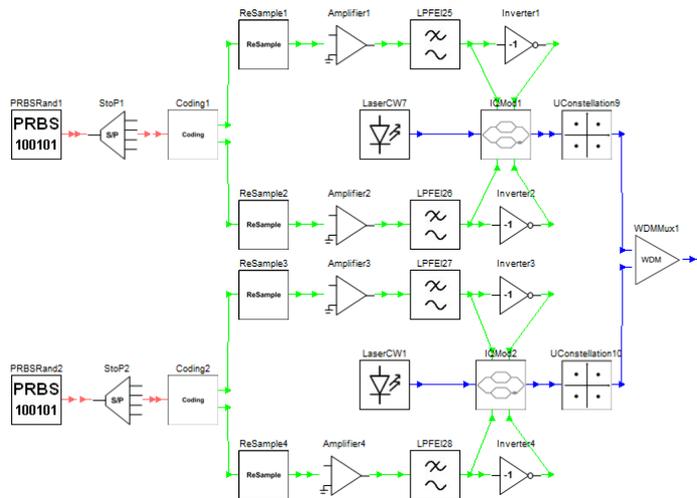


Figura 4.35: Modelo do transmissor WDM com dois canais SP-QPSK implementado no OSIP.

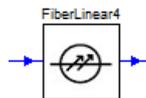


Figura 4.36: Exemplo ilustrativo do canal de transmissão considerado para a simulação do sistema WDM 2xSP-QPSK.

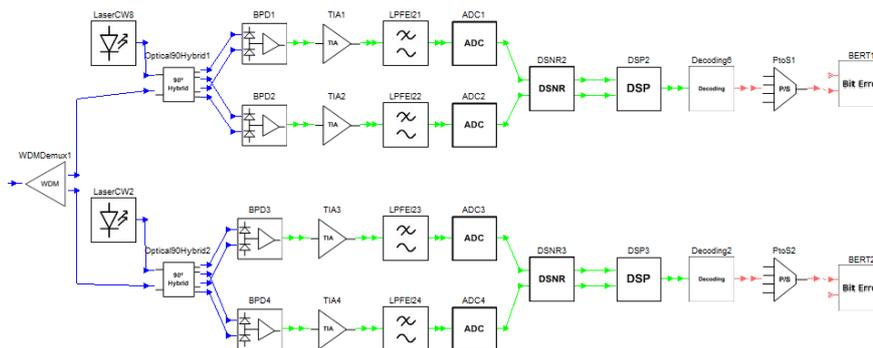


Figura 4.37: Modelo do recetor para dois canais SP-QPSK implementado no OSIP.

Os canais são modulados individualmente e posteriormente são multiplexados no comprimento de onda utilizando para o efeito o componente *WDM_Mux* (ver Figura 4.35). De seguida o sinal é transmitido sendo utilizado um troço de fibra SSMF com 500km para simular o canal de transmissão. Deve notar-se que o modelo do canal apresentado na Figura 4.36 apenas considera o efeito da dispersão cromática. No caso de estudo o sinal é apenas modulado numa componente de polarização, no entanto a PMD continua presente, dado que resulta de um comportamento intrínseco das fibras óticas. Revela-se como uma distribuição de potência entre as duas polarizações e uma rotação de fase. Apesar disso não se considerou no âmbito desta simulação. O recetor (ver Figura 4.37) adquire o sinal e divide-o nos dois canais óticos utilizando o componente *WDM_Demux* sendo posteriormente cada um deles misturado com um laser local e convertido nas suas componentes em fase e quadratura. Estas são depois amostradas nas ADCs a uma frequência de amostragem que é o dobro do ritmo de símbolos do canal. Já no domínio

digital estes são processados e as sequências binárias decodificadas. As configurações adicionais dos componentes encontram-se na Tabela 4.5.

Transmissor	Recetor
- Filtro de Bessel a 50GHz	- Filtro de Bessel a 25GHz
- Filtro Gaussiano largura de banda de 100GHz	- Filtro Gaussiano largura de banda de 100GHz
- DAC de 8bits	- ADC de 8bits
- Ganho de 13.98dB no amplificador	- 2 amostras/símbolo
- V_{π} de 5V	

Tabela 4.5: Configurações do transmissor e do recetor para a simulação do sistema WDM 2xSP-QPSK.

Para efeitos de simulação bastaria que os filtros de Bessel do transmissor apresentassem uma largura de banda de 25GHz. Sendo o espaçamento entre canais de 100GHz os filtros utilizados visam a preservação do máximo de informação espectral do sinal. Já os do recetor são ajustados para o ritmo de símbolos para que da operação de amostragem não resulte *aliasing*. Para filtrar os canais óticos aquando da multiplexagem e demultiplexagem utilizam-se filtros gaussianos, no *WDM_MUX* e no *WDM_Demux*, com uma largura de banda igual ao espaçamento entre canais (100GHz). No DSP introduzem-se os algoritmos de compensação da dispersão cromática nos domínios da frequência e do tempo e o V&V com linha de realimentação e processamento de 256 amostras em paralelo. O espectro dos canais após a multiplexagem encontra-se na Figura 4.38.

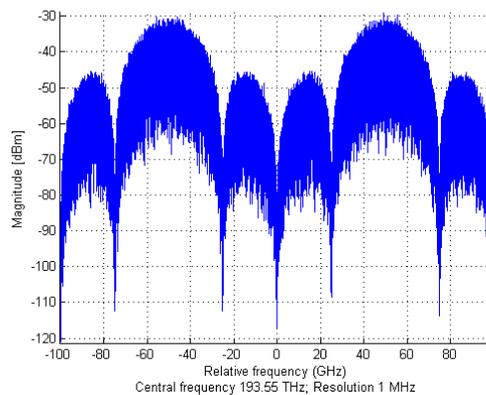


Figura 4.38: Ilustração do espectro do sinal após a multiplexagem dos dois canais SP-QPSK.

O espectro é observado adicionando o componente OSA, o qual é configurado com uma resolução de 1MHz. Pode-se verificar na Figura 4.38 que os canais estão corretamente centrados nas frequências especificadas, que apresentam uma largura de banda de 100GHz e que não se sobrepõem. Para as simulações a realizar consideram-se agora dois cenários: com e sem canal de transmissão. Variando a potência do ruído no recetor obtém-se as relações de BER com a SNR que são apresentadas na Figura 4.39, para os dois canais.

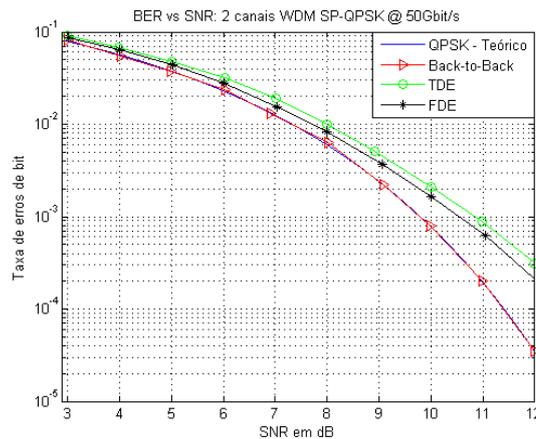


Figura 4.39: Relações entre a BER e a SNR para o sistema WDM 2xSP-QPSK.

Pela Figura 4.39 verifica-se que, na ausência do canal de transmissão o sistema segue a curva teórica. Adicionando o canal de transmissão regista-se uma penalidade que neste cenário sofre um ligeiro aumento para valores baixos de SNR, para valores de SNR superiores não são registadas agravantes nas penalidades. Nas secções “Desempenho dos algoritmos TDE e FDE” e “Desempenho dos algoritmos de correção de fase” verificou-se que para baixos valores de SNR a penalidade introduzida era bastante pequena. Agora, contrariamente às secções anteriores, os algoritmos são utilizados em conjunto, verificando-se que a dispersão residual influencia o desempenho do algoritmo de estimação de fase, sendo introduzidos erros e como tal uma penalidade ligeiramente superior. Para valores de SNR mais elevados o comportamento é semelhante ao observado para os algoritmos de equalização da dispersão para duas amostras por símbolo, sendo que, desta forma o algoritmo de correção de fase opera de forma transparente. Os resultados registados sugerem que o algoritmo de compensação da dispersão no domínio da frequência apresenta melhor desempenho que no domínio do tempo. Na verdade, deve notar-se que, tal como se referiu no tema 4.2.1.3, os algoritmos TDE e FDE não podem ser comparados diretamente, uma vez que apresentam um número diferente de coeficientes, ainda assim o algoritmo FDE deverá apresentar melhor desempenho que o TDE. O uso do algoritmo de compensação da PMD permitiria reduzir esta penalidade para baixos valores de SNR uma vez que permite compensar a dispersão residual.

4.3.2 Dois canais PDM-QPSK a 100Gbit/s

Para este cenário a informação binária é gerada a 100Gbit/s sendo utilizado para o efeito dois PRBS independentes a 50Gbit/s por canal que são modulados com o formato QPSK. O comprimento da sequência é ajustado de acordo com a probabilidade de erro esperada teoricamente para o caso ideal. Considera-se que os canais estão espaçados de 100GHz e portanto, os lasers têm frequências de 193.5THz e 193.6THz, sendo estes configurados com uma largura de linha de 100kHz. Os modelos do transmissor, do canal e do recetor encontram-se na Figura 4.40, na Figura 4.41 e na Figura 4.42

respetivamente. Deve notar-se que estas figuras apenas representam os modelos para sistemas PDM-QPSK, desta forma é necessário utilizar dois transmissores idênticos ao da Figura 4.40, assim como dois recetores idênticos ao da Figura 4.42.

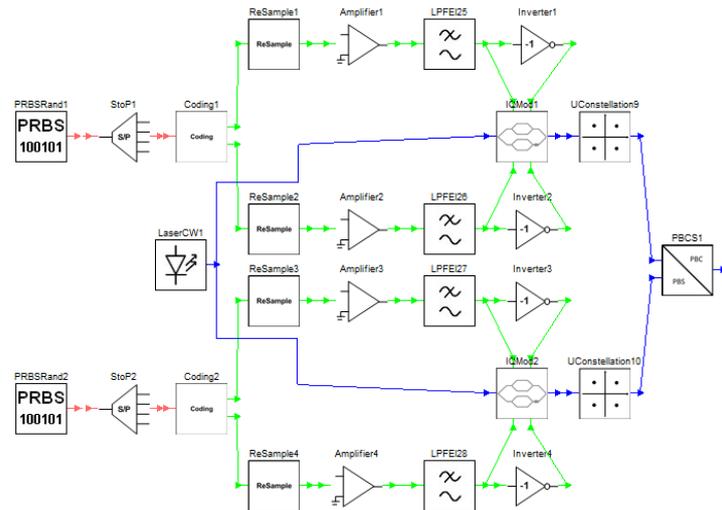


Figura 4.40: Modelo do transmissor implementado no OSIP para um canal PDM-QPSK.

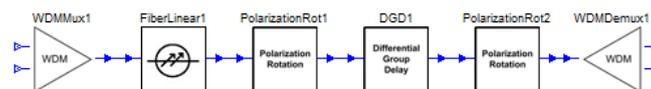


Figura 4.41: Exemplo ilustrativo da multiplexagem e desmultiplexagem dos canais WDM e do canal de transmissão.

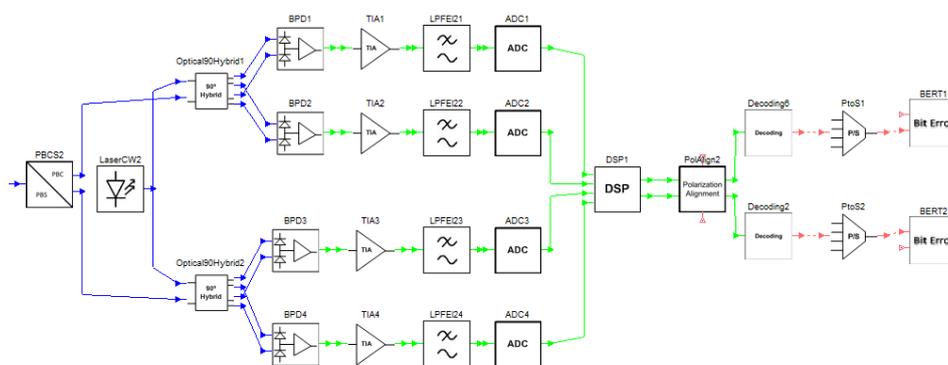


Figura 4.42: Modelo do recetor ótico implementado no OSIP para um canal PDM-QPSK.

Após serem moduladas as duas componentes de polarização são combinadas recorrendo a um PBC. Os dois canais são gerados de forma idêntica e são multiplexados no comprimento de onda, utilizando para o efeito o componente WDM_Mux (ver Figura 4.40 e Figura 4.41). Deve notar-se que os componentes de multiplexagem e desmultiplexagem dos canais estão representados juntamente com o canal de transmissão. Para simular o meio de propagação considera-se um troço de fibra SSMF com 500km e PMD de primeira ordem com um parâmetro de DGD de 2.236ps, correspondente ao comprimento de fibra considerado. As rotações da polarização são geradas aleatoriamente. No recetor os dois canais são separados, sendo utilizado para o

efeito o componente *WDM_Demux* assim como as componentes de polarização de cada canal (recorrendo a um PBS) e os sinais resultantes misturados com o oscilador local. Deste processo resulta a representação do sinal nas suas componentes de fase e quadratura, as quais são amostrados nas ADCs com uma frequência de amostragem que é o dobro do ritmo de símbolos. Os sinais digitalizados são posteriormente processados no DSP, onde são aplicados os algoritmos de compensação da dispersão cromática nos domínios da frequência e do tempo, o CMA e o V&V com linha de realimentação. O CMA é dimensionado com 3 e 5 coeficientes por filtro e o algoritmo de estimação de fase com processamento de 256 amostras em paralelo. Concluída a equalização dos efeitos lineares a sequência de dados é decodificada (ver Figura 4.42). Adicionalmente utiliza-se um componente para alinhar as polarizações visto que o algoritmo de equalização da PMD não dá garantias para qual das suas saídas converge cada polarização. As restantes configurações dos componentes encontram-se na Tabela 4.6.

Transmissor	Recetor
- Filtro de Bessel a 50GHz	- Filtro de Bessel a 25GHz
- Filtro Gaussiano largura de banda de 100GHz	- Filtro Gaussiano largura de banda de 100GHz
- DAC de 8bits	- ADC de 8bits
- Ganho de 13.98dB no amplificador	- 2 amostras/símbolo
- V_{π} de 5V	

Tabela 4.6: Configurações do transmissor e do recetor para a simulação do sistema WDM com dois canais DP-QPSK.

Deve notar-se que os filtros de Bessel do transmissor poderiam ser de 25GHz, apenas se adotou filtros com uma largura de banda de 50GHz para preservar o máximo da informação espectral possível de acordo com a separação entre canais considerada. Os filtros do recetor apresentam uma largura de banda de 25GHz de modo evitar *aliasing* na operação de amostragem. Para filtrar os canais óticos aquando da multiplexagem e demultiplexagem utilizam-se filtros gaussianos, no *WDM_MUX* e no *WDM_Demux*, com uma largura de banda igual ao espaçamento entre canais (100GHz). O espectro dos canais após a multiplexagem encontra-se na Figura 4.43.

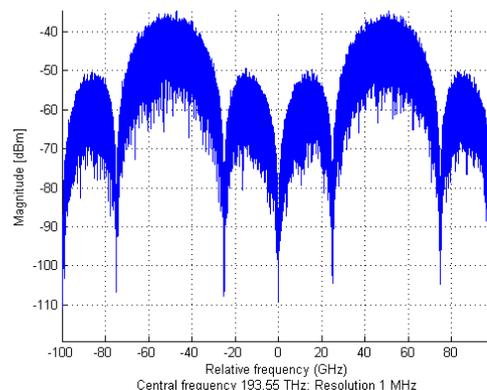


Figura 4.43: Ilustração do espectro do sinal após a multiplexagem dos dois canais PDM-QPSK.

O espectro do sinal é observado adicionando o componente OSA, o qual é configurado com uma resolução de 1MHz. Pode-se verificar na Figura 4.43 que os canais estão corretamente centrados nas frequências especificadas não sendo verificadas sobreposições espectrais e apresentando cada um deles uma largura de banda de 100GHz. Para testar o desempenho do sistema consideram-se dois cenários. No primeiro pretende-se simular o sistema sem canal de transmissão. O objetivo é registar as relações entre a BER e a SNR, variando a potência do ruído no recetor. No segundo caso o procedimento é o mesmo, passando apenas a considerar-se o canal de transmissão. Os resultados obtidos para os dois canais encontram-se na Figura 4.44.

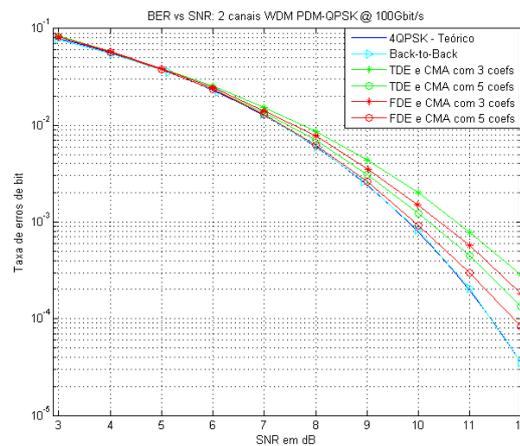


Figura 4.44: Relações entre a BER e a SNR para o sistema WDM 2xPDM-QPSK.

Pela Figura 4.44 verifica-se que na ausência do canal de transmissão o sistema segue as relações teóricas. Adicionando o canal de transmissão registam-se penalidades para valores de SNR superiores a 6dB. Estas penalidades são inferiores às registadas para a dispersão cromática (*Desempenho dos algoritmos TDE e FDE*). Apesar do CMA introduzir alguma penalidade, como foi verificado na secção “Desempenho do algoritmo CMA”, a penalidade total registada é inferior à soma de todas as penalidades. A compensação de parte da dispersão residual pelo algoritmo CMA permite eliminar os erros cometidos pelo algoritmo de estimação de fase para baixos valores de SNR (contrariamente ao que se verificou na secção “Dois canais SP-QPSK a 50Gbit/s”) e reduzir substancialmente o seu valor para as outras gamas de SNR. Pode-se verificar na Figura 4.44 que para o algoritmo CMA com 3 coeficientes a penalidade ainda é substancialmente elevada. Contudo, aumentando o comprimento do filtro para 5 coeficientes a dispersão residual é praticamente compensada na totalidade, sendo registada uma penalidade para valores elevados de SNR que rondam, para uma BER de 10^{-4} , os 0.5dB e 1dB para os algoritmos FDE e TDE respetivamente. Desta forma a penalidade introduzida pelo algoritmo de compensação da dispersão cromática pode ser eliminada na totalidade pelo algoritmo CMA, como é referido em [43, 68]. Contudo deve-se notar que este passa a ter um tempo de convergência superior. Na verdade quando o algoritmo CMA atinge o ponto de convergência este representa a função inversa do canal de transmissão, permitindo assim a compensação da dispersão cromática. Deve notar-se

ainda assim, que a quantidade de dispersão que é possível compensar com este filtro depende do número de coeficientes utilizados, logo da resolução da aproximação ao canal de transmissão.

4.4 PDM-QPSK a 100Gbit/s

Neste tema pretende-se simular um sistema com diversidade de polarização PDM-QPSK, com e sem fibra. O estudo da penalidade introduzida pelos algoritmos de compensação digitais é bastante importante definindo as suas condições e limites de implementação. A informação a ser transmitida é gerada a 100Gbit/s sendo para o efeito utilizado um PRBS a 50Gbit/s por polarização. O comprimento da sequência binária é ajustado de acordo com a probabilidade de erro que é esperada teoricamente. O modelo do transmissor, do canal e do recetor encontram-se na Figura 4.45, na Figura 4.46 e na Figura 4.47 respetivamente.

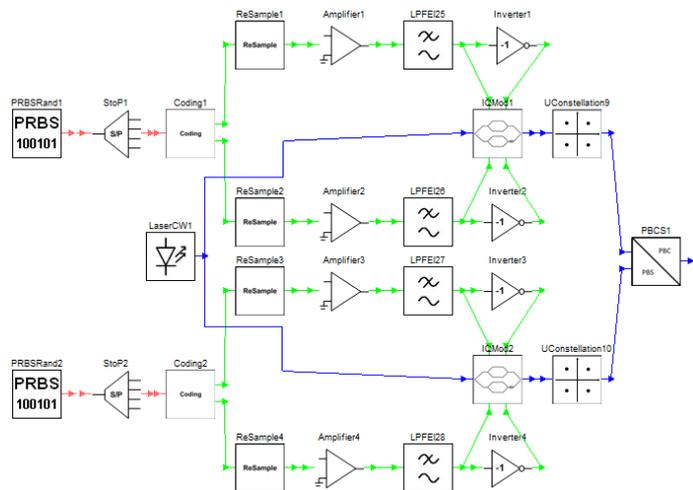


Figura 4.45: Modelo do transmissor com multiplexagem de polarização implementado no OSIP para o sistema PDM-QPSK.

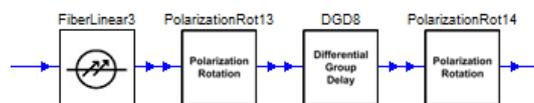


Figura 4.46: Modelo do canal de transmissão para simular a CD e a PMD.

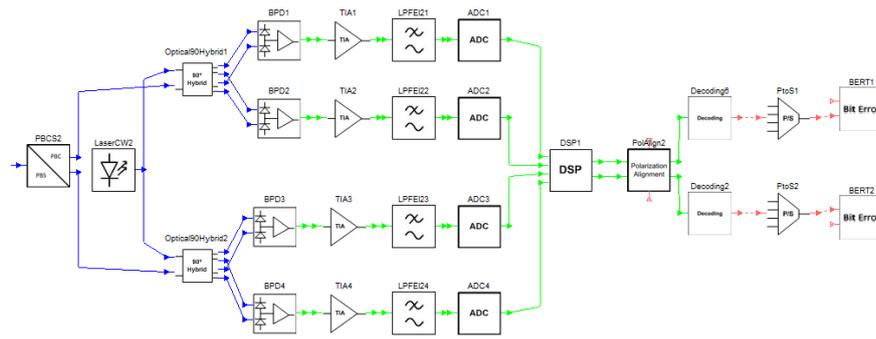


Figura 4.47: Modelo do recetor com diversidade de polarização implementado no OSIP para o sistema DP-QPSK.

Configurando o componente de codificação é definido em cada polarização o formato de modulação QPSK, os restantes componentes do transmissor e do recetor também são configurados segundo as especificações desejadas, Tabela 4.7. Após a modulação do sinal em cada uma das polarizações, as duas componentes são combinados utilizando um PBC e enviados pelo canal de transmissão. No recetor as duas polarizações são divididas recorrendo para o efeito a um PBS e convertidas para o domínio digital, sendo posteriormente processadas digitalmente para compensar os efeitos lineares que ocorrem na propagação e as sequências binárias são decodificadas. No DSP implementam-se os algoritmos de compensação da dispersão cromática nos domínios da frequência e do tempo, o CMA com 3 e 5 coeficientes por filtro e o V&V com linha de realimentação e processamento paralelo de 256 amostras. Adicionalmente é utilizado um componente para alinhar as polarizações visto que ao ser aplicado o algoritmo CMA as polarizações podem vir rodadas.

Transmissor	Recetor
- Filtro de Bessel a 25GHz	- Filtro de Bessel a 25GHz
- DAC de 8bits	- ADC de 8bits
- Ganho de 13.98dB no amplificador	- 2 amostras/símbolo
- Laser a 193.5THz	- Laser a 193.5THz
- V_{π} de 5V	

Tabela 4.7: Configurações dos componentes do transmissor e do recetor para o sistema PDM-QPSK.

Os lasers do transmissor e do recetor são configurados para uma largura de linha de 100kHz. Considera-se o canal de transmissão como sendo o conjunto da dispersão cromática e da PMD sendo utilizado para o efeito o modelo de primeira ordem representado na Figura 4.46. Para a simulação considera-se um troço de fibra SSMF com 500km ao qual corresponde uma DGD de aproximadamente 2.236ps. Adicionalmente para controlar a potência do ruído no recetor foi adicionado o componente DSNR. O primeiro objetivo é averiguar, passo-a-passo, o efeito de cada algoritmo no sinal. Para isso, dos algoritmos implementados no DSP que foram referidos anteriormente, considerou-se a compensação da CD no domínio da frequência, o CMA com 3

coeficientes e o V&V com realimentação. Para um SNR de 15dB obtiveram-se os resultados apresentados na Figura 4.48.

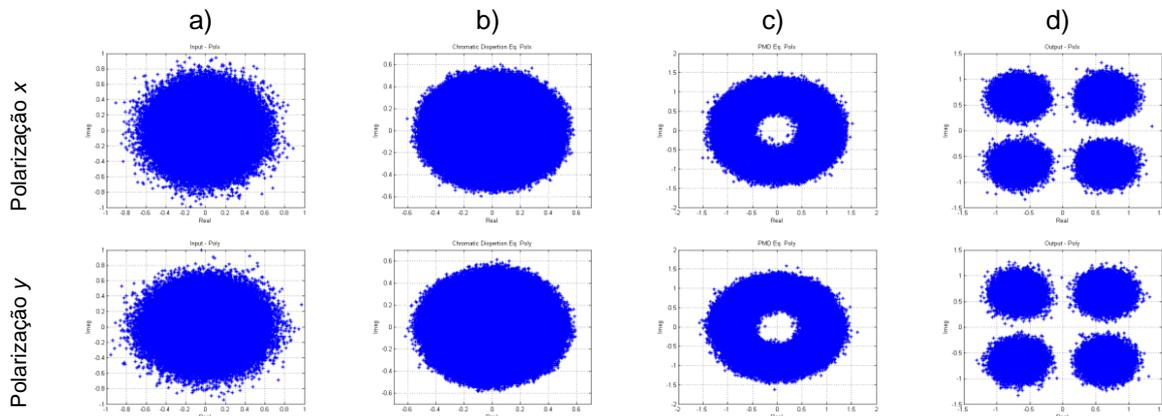


Figura 4.48: Equalização do sinal passo-a-passo: a) – sinal de entrada do equalizador; b) – sinal após a compensação da CD; c) – sinal após a compensação da PMD; d) – sinal após a correção de fase.

Da compensação da CD ainda se verifica uma interferência elevada, que resulta do ruído e dos efeitos da PMD. Deve notar-se que a PMD origina 16 pontos da constelação os quais estão ainda sujeitos a ruído branco gaussiano e a ruído de fase. Aplicando o algoritmo de compensação da PMD já é possível verificar uma constelação circular centrada na circunferência de módulo unitário. O algoritmo V&V permite recuperar a fase do sinal, obtendo-se a constelação final do sinal recuperado nas duas polarizações. Verifica-se que o sinal é corretamente recuperado. Um aspeto importante é analisar o impacto do uso destes algoritmos no sistema. Considerando as condições iniciais, todos os algoritmos implementados no DSP, e variando a potência do ruído, as curvas de BER obtidas para as duas polarizações encontram-se na Figura 4.49.

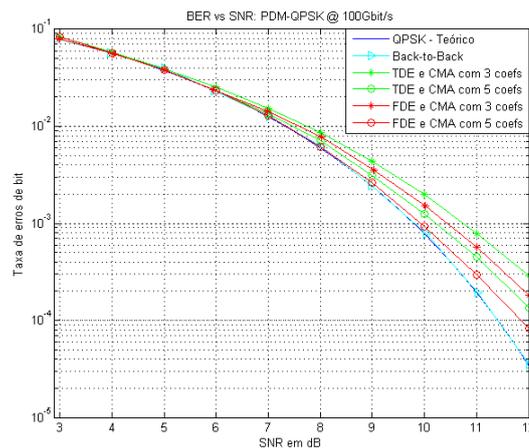


Figura 4.49: Resultados da BER obtidos para o sistema PDM-QPSK.

Os resultados permitem verificar que o sistema funciona corretamente quando o transmissor está ligado diretamente ao recetor sendo as curvas de BER iguais às calculadas teoricamente. Quando se introduz o canal de transmissão os algoritmos de

compensação introduzem penalidades, como se verificou na secção “4.2 Implementação de algoritmos DSP”. Apesar de agora se reunirem todos os algoritmos a penalidade não sofre agravantes. À semelhança do cenário anterior a penalidade vem ligeiramente reduzida devido ao facto do CMA compensar em parte a dispersão cromática residual. Esta pode ser mesmo eliminada na sua totalidade aumentando o número de coeficientes dos filtros do CMA, como pode ser verificado na Figura 4.49. Como custo, o tempo de convergência do algoritmo aumenta, assim como a sua complexidade. As relações registadas são idênticas às verificadas na secção “Dois canais PDM-QPSK a 100Gbit/s”, o que indica que não ocorre interferência entre canais e como tal um agravamento na penalidade. A penalidade introduzida toma o valor máximo de 1dB e 0.5dB na gama representada, para o algoritmo FDE e cada um dos filtros CMA considerados, já no caso do TDE registaram-se as penalidades de 1.2dB e 0.8dB. O uso de outros algoritmos, dos quais são exemplo os FECs, permite tornar os sistemas mais robustos ao ruído e assim reduzir o impacto desta penalidade, facilitando a implementação deste tipo de igualadores. Outra solução para reduzir esta penalidade, passa por aumentar o número de coeficientes dos filtros do CMA.

5 Conclusões e Trabalho Futuro

Ao longo deste trabalho foram apresentadas as novas funcionalidades da versão atual do OSIP. O algoritmo de controlo com base no método de Monte-Carlo permite dividir as simulações mais longas em vários blocos de menor dimensão, de modo a que as simulações se processem de forma mais eficiente e com menores recursos computacionais. Considerando também a otimização dos recursos, os algoritmos de suporte a esta funcionalidade devem ser o mais simples e robustos possíveis de forma a não influenciar o desempenho do sistema. O mesmo deve ser considerado no projeto dos algoritmos de suporte à interface gráfica. Do ponto de vista do utilizador é fundamental a compreensão do modo de operação do simulador. A facilidade de compreensão da sua arquitetura permite realizar atualizações frequentes assim como também o desenvolvimento de novos componentes. Como tal, o simulador apresenta uma arquitetura muito simples baseada numa hierarquia de diretorias. Os componentes adicionados permitem que o simulador esteja preparado para a simulação dos sistemas mais atuais, tornando-se assim numa excelente ferramenta para fins de investigação. No capítulo 3 foram apresentadas as novas funcionalidades do simulador assim como a sua arquitetura. A sua organização simplificada possibilita uma fácil comunicação entre processos e o desenvolvimento de novas funcionalidades orientadas a novos componentes, métodos de simulação ou mesmo algoritmos de controlo. Outro aspeto abordado foi a atualização do simulador para sistemas coerentes. Os componentes adicionados permitem a simulação de forma eficiente dos modelos do transmissor e do recetor desejados, bem como simular o processamento digital de sinal. Por outro lado, o uso de identificadores permite desde já uma maior flexibilidade nas configurações dos modelos. No futuro poderá permitir a implementação de novos sistemas, por exemplo com recurso a multiplexagem modal. No capítulo 4 são apresentadas algumas simulações que permitem validar o desempenho do simulador. São testados os componentes do transmissor e do recetor de modo a comparar o seu comportamento com os resultados esperados teoricamente. Foram também implementados e testados alguns algoritmos de compensação da CD e da PMD bem como de estimação e correção de fase.

Verificou-se a dependência da resolução dos filtros de equalização da CD na qualidade da compensação e como tal na quantidade de dispersão residual registada. Quer nas implementações no domínio do tempo quer na frequência, aumentando a taxa de amostragem é possível igualar os efeitos da CD sem que daí resulte qualquer penalidade para o sistema. Claro que este número também está limitado pelos tempos de computação e pelos recursos dos componentes físicos utilizados para o efeito. Na compensação da PMD testou-se a condição do número mínimo de coeficientes necessários para representar a função de transferência inversa do canal. Este número é válido e possível de utilizar, contudo origina uma penalidade. Esta penalidade pode ser reduzida aumentando o número de coeficientes dos filtros permitindo assim uma aproximação mais fiel do canal de transmissão. Foram também testados os algoritmos de

estimação e correção de fase sendo verificado que o seu desempenho está fortemente dependente do ruído. Uma boa tolerância a ruído aditivo branco gaussiano implica um elevado número de amostras em processamento paralelo, enquanto que, para acompanhar as variações impostas pelo ruído de fase é necessário um número reduzido de amostras. A escolha do número de amostras terá de ser assim, sempre que possível, um compromisso para otimizar o desempenho destes algoritmos. O estudo de modelos de implementação real é um aspeto que possibilita a investigação do seu desempenho e como tal a sua viabilidade de implementação. Por outro lado, ao possibilitarem a compensação dos efeitos resultantes do canal de transmissão permitem testar os algoritmos em conjunto, são exemplo o PDM-QPSK e sistemas WDM com e sem multiplexagem de polarização. Os algoritmos revelam um bom funcionamento conjunto. Foi possível verificar que o algoritmo CMA é capaz de compensar a dispersão cromática residual, reduzindo a penalidade do sistema.

A nova versão do simulador de comunicações óticas apresenta-se como uma ferramenta poderosa e preparada para os sistemas futuros. A otimização dos métodos de simulação continua a ser um fator preponderante, traduzindo-se na eficiência da simulação e na necessidade de recursos. Também, a possibilidade de implementar algoritmos de compensação no domínio digital permite estudar os sistemas de uma forma mais realista. Do ponto de vista da implementação, estes permitem a redução da complexidade da parte ótica dos sistemas e o desenvolvimento de sistemas mais eficientes. O estudo destes algoritmos e a implementação de novas funcionalidades e modelos irão contribuir para o ensino e investigação dos presentes e próximos sistemas óticos coerentes.

5.1 Tópicos de Trabalho Futuro

Como sugestões de trabalho futuro a ser desenvolvido na sequência deste trabalho são propostos os tópicos:

- Processamento paralelo por computação gráfica, GPU (*Graphics Processing Unit*): O processamento paralelo permite aumentar o desempenho do simulador, reduzindo os tempos de computação. Torna-se especialmente importante para simular a propagação não-linear em fibra ótica.

- Implementação de sistemas multimodo: Apesar de ainda não se fazer uso deste tipo de sistemas em implementações reais, o simulador ao utilizar identificadores encontra-se preparado para a sua implementação.

- Compensação de efeitos não-lineares: Estes continuam a ser uma das grandes limitações nos sistemas óticos. Implementado o DSP é possível desenvolver e testar em ambiente de simulação novos algoritmos. Dada a importância dos efeitos não-lineares, tais algoritmos são de grande interesse de estudo.

- Introdução de sequências de treino: Estas permitem compensar o canal de transmissão através da inversão da sua função de transferência, método estudado em [36].

- Alteração do método de simulação para permitir o processamento em série: Pretende-se que apresente um funcionamento idêntico ao do *Simulink*[®]. Desta forma

seria possível estudar os sistemas de forma mais realista assim como realizar qualquer tipo de sistema no ambiente de simulação, incluindo realimentações.

- Implementação em FPGA de alguns algoritmos de compensação digital: Estudado o seu desempenho em ambiente de simulação torna-se interessante a sua implementação em *hardware* e a comparação dos resultados. Por outro lado permite o seu teste em sistemas reais.

Referências

- [1] W. J. Tomlinson, "Evolution of Passive Optical Component Technologies for Fiber-Optic Communication Systems," *Lightwave Technology, Journal of*, vol. 26, pp. 1046-1063, 2008.
- [2] G. P. Agrawal, *Fiber-Optic Communication Systems*: Wiley-Interscience, 2002.
- [3] A. Gnauck, *et al.*, "High-Capacity Optical Transmission Systems," *J. Lightwave Technol.*, vol. 26, pp. 1032-1045, 2008.
- [4] R. C. Alferness, *et al.*, "The evolution of optical systems: Optics everywhere," *Bell Labs Technical Journal*, vol. 5, pp. 188-202, 2000.
- [5] T. Okoshi, "Heterodyne and Coherent Optical Fiber Communications: Recent Progress," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 30, pp. 1138-1149, 1982.
- [6] G. Keiser, *Optical Fiber Communications*, third ed.: McGraw-Hill, 2000.
- [7] M. Seimetz, *High-Order Modulation for Optical Fiber Transmission*: Springer, 2009.
- [8] G. Li, "Recent advances in coherent optical communication," *Adv. Opt. Photon.*, vol. 1, pp. 279-307, 2009.
- [9] <http://www.sciencedirect.com/>. (2012, 09-05-2012).
- [10] P. J. Winzer and R. J. Essiambre, "Advanced Optical Modulation Formats," *Proceedings of the IEEE*, vol. 94, pp. 952-985, 2006.
- [11] VPIsystems. (2012). <http://www.vpiphotonics.com/>.
- [12] D. Gurkan, *et al.*, "Remote Laboratories for Optical Circuits," *Education, IEEE Transactions on*, vol. 51, pp. 53-60, 2008.
- [13] A. J. Sakena, *et al.*, "Optical amplifier number and placement in the superPON architecture," in *Advanced Communication Technology, 2004. The 6th International Conference on*, 2004, pp. 186-191.
- [14] OSIP, "<http://www.av.it.pt/osip/>," 2012.
- [15] B. Pereira, *et al.*, "Simulador de Sistemas de Comunicações Óticas: OSIP 2012," *DETUA*, 2012.
- [16] B. Pereira, *et al.*, "Performance of digital equalization techniques for coherent optical receivers," *SEONS*, 2012.
- [17] E. Ip, *et al.*, "Coherent detection in optical fiber systems," *Opt. Express*, vol. 16, pp. 753-791, 2008.
- [18] T. L. Ivan P. Kaminow, Alan E. Willner, *Optical Fiber Telecommunications V B, Systems and Networks*: Elsevier, 2008.
- [19] Y. Yamamoto and T. Kimura, "Coherent optical fiber transmission systems," *Quantum Electronics, IEEE Journal of*, vol. 17, pp. 919-935, 1981.
- [20] R. A. Linke and A. H. Gnauck, "High-capacity coherent lightwave systems," *Lightwave Technology, Journal of*, vol. 6, pp. 1750-1769, 1988.
- [21] T. Pfau, *et al.*, "Phase-Noise-Tolerant Two-Stage Carrier Recovery Concept for Higher Order QAM Formats," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 16, pp. 1210-1216, 2010.

- [22] G. Charlet, "Coherent detection associated with digital signal processing for fiber optics communication," *Comptes Rendus Physique*, vol. 9, pp. 1012-1030, 2008.
- [23] M. Seimetz, "Performance of coherent optical square-16-QAM-systems based on IQ-transmitters and homodyne receivers with digital phase estimation," in *Optical Fiber Communication Conference, 2006 and the 2006 National Fiber Optic Engineers Conference. OFC 2006*, 2006, p. 10 pp.
- [24] G.-W. Lu, *et al.*, "40-Gbaud 16-QAM transmitter using tandem IQ modulators with binary driving electronic signals," *Opt. Express*, vol. 18, pp. 23062-23069, 2010.
- [25] M. Seimetz, "Multi-format transmitters for coherent optical M-PSK and M-QAM transmission," in *Transparent Optical Networks, 2005, Proceedings of 2005 7th International Conference*, 2005, pp. 225-229 Vol. 2.
- [26] P. Watts, *et al.*, "An FPGA-Based Optical Transmitter Design Using Real-Time DSP for Advanced Signal Formats and Electronic Predistortion," *J. Lightwave Technol.*, vol. 25, pp. 3089-3099, 2007.
- [27] R. I. Killey, *et al.*, "Electronic dispersion compensation by signal predistortion using digital Processing and a dual-drive Mach-Zehnder Modulator," *Photonics Technology Letters, IEEE*, vol. 17, pp. 714-716, 2005.
- [28] P. J. Winzer and R. J. Essiambre, "Advanced Modulation Formats for High-Capacity Optical Transport Networks," *Lightwave Technology, Journal of*, vol. 24, pp. 4711-4728, 2006.
- [29] S. J. Savory, "Digital filters for coherent optical receivers," *Opt. Express*, vol. 16, pp. 804-817, 2008.
- [30] M. Kuschnerov, *et al.*, "DSP for Coherent Single-Carrier Receivers," *Lightwave Technology, Journal of*, vol. 27, pp. 3614-3622, 2009.
- [31] F. Buchali and B. Henning, "Adaptive PMD compensation by electrical and optical techniques," *Lightwave Technology, Journal of*, vol. 22, pp. 1116-1126, 2004.
- [32] T. Kudou, *et al.*, "Theoretical basis of polarization mode dispersion equalization up to the second order," *Lightwave Technology, Journal of*, vol. 18, pp. 614-617, 2000.
- [33] H. Bulow, *et al.*, "Measurement of the maximum speed of PMD fluctuation in installed field fiber," in *Optical Fiber Communication Conference, 1999, and the International Conference on Integrated Optics and Optical Fiber Communication. OFC/IOOC '99. Technical Digest*, 1999, pp. 83-85 vol.2.
- [34] H. Bulow and G. Veith, "Temporal dynamics of error-rate degradation induced by polarisation mode dispersion fluctuation of a field fiber link," in *Integrated Optics and Optical Fibre Communications, 11th International Conference on, and 23rd European Conference on Optical Communications (Conf. Publ. No.: 448)*, 1997, pp. 115-118 vol.1.
- [35] T. Merker, *et al.*, "Comparison of PMD-compensation techniques at 10 Gbit/s using an optical first-order compensator and electrical transversal filter," *Optics Communications*, vol. 182, pp. 135-141, 2000.
- [36] B. Spinnler, "Equalizer Design and Complexity for Digital Coherent Receivers," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 16, pp. 1180-1192, 2010.
- [37] F. P. Guiomar, *et al.*, "Digital Postcompensation Using Volterra Series Transfer Function," *Photonics Technology Letters, IEEE*, vol. 23, pp. 1412-1414, 2011.

- [38] E. Mateo, *et al.*, "Nonlinearity compensation using very-low complexity backward propagation in dispersion managed links," 2012, p. OTh3C.4.
- [39] R. Freund, *et al.*, "Next generation optical networks based on higher-order modulation formats, coherent receivers and electronic distortion equalization," in *ICTON Mediterranean Winter Conference, 2009. ICTON-MW 2009. 3rd*, 2009, pp. 1-6.
- [40] S. J. Savory, "Digital Coherent Optical Receivers: Algorithms and Subsystems," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 16, pp. 1164-1179, 2010.
- [41] M. Seimetz and C.-M. Weinert, "Options, Feasibility, and Availability of 2×4 90° Hybrids for Coherent Optical Systems," *J. Lightwave Technol.*, vol. 24, p. 1317, 2006.
- [42] R. Kudo, *et al.*, "Coherent Optical Single Carrier Transmission Using Overlap Frequency Domain Equalization for Long-Haul Optical Systems," *Lightwave Technology, Journal of*, vol. 27, pp. 3721-3728, 2009.
- [43] T. Xu, *et al.*, "Chromatic dispersion compensation in coherent transmission system using digital filters," *Opt. Express*, vol. 18, pp. 16243-16257, 2010.
- [44] F. Yangyang, *et al.*, "The Comparison of CMA and LMS Equalization Algorithms in Optical Coherent Receivers," in *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, 2010, pp. 1-4.
- [45] C. S. Fludger, *et al.*, "Coherent Equalization and POLMUX-RZ-DQPSK for Robust 100-GE Transmission," *J. Lightwave Technol.*, vol. 26, pp. 64-72, 2008.
- [46] I. Fatadin and S. J. Savory, "DSP techniques for 16-QAM coherent optical systems," in *Photonics Society Summer Topical Meeting Series, 2010 IEEE*, 2010, pp. 22-23.
- [47] K. Kikuchi, "Polarization-demultiplexing algorithm in the digital coherent receiver," in *IEEE/LEOS Summer Topical Meetings, 2008 Digest of the*, 2008, pp. 101-102.
- [48] S. Narayan, *et al.*, "Transform domain LMS algorithm," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 31, pp. 609-615, 1983.
- [49] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm: analysis and simulations," *Signal Processing, IEEE Transactions on*, vol. 45, pp. 631-639, 1997.
- [50] C. M. Panazio and J. M. T. Romano, "On the convergence of a new joint DFE & decoding procedure for blind decision directed LMS equalization," in *Communications, 2002. ICC 2002. IEEE International Conference on*, 2002, pp. 129-133.
- [51] R. Noe, "Phase noise-tolerant synchronous QPSK/BPSK baseband-type intradyne receiver concept with feedforward carrier recovery," *Lightwave Technology, Journal of*, vol. 23, pp. 802-808, 2005.
- [52] T. Pfau, *et al.*, "Hardware-Efficient Coherent Digital Receiver Concept With Feedforward Carrier Recovery for M -QAM Constellations," *J. Lightwave Technol.*, vol. 27, pp. 989-999, 2009.
- [53] N. Stojanovic, *et al.*, "Robust Carrier Recovery in Polarization Division Multiplexed Receivers," 2012, p. OTu2G.3.
- [54] H. Sun and K.-T. Wu, "Techniques in Carrier Recovery for Optical Coherent Systems," 2012, p. OTh4C.3.

- [55] Q. Zhuge, *et al.*, "Linewidth-Tolerant Low Complexity Pilot-Aided Carrier Phase Recovery for M-QAM using Superscalar Parallelization," 2012, p. OTu2G.2.
- [56] S. Dris, *et al.*, "Phase Entropy-Based Frequency Offset Estimation for Coherent Optical QAM Systems," 2012, p. OTu2G.4.
- [57] C. Xie, *et al.*, "Adaptive Carrier Phase Estimation in Coherent Systems," 2012, p. OTu2G.5.
- [58] K. Kikuchi, "Phase-diversity homodyne detection of multilevel optical modulation with digital carrier phase estimation," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 12, pp. 563-570, 2006.
- [59] M. Seimetz, "Laser Linewidth Limitations for Optical Systems with High-Order Modulation Employing Feed Forward Digital Carrier Phase Estimation," in *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*, 2008, pp. 1-3.
- [60] G. De Jonghe and M. Moeneclaey, "Optimal averaging filter length of the Viterbi and Viterbi carrier synchronizer for a given frequency offset," in *Global Telecommunications Conference, 1994. GLOBECOM '94. Communications: The Global Bridge., IEEE*, 1994, pp. 1363-1368 vol.3.
- [61] R. A. Shafik, *et al.*, "On the Extended Relationships Among EVM, BER and SNR as Performance Metrics," in *Electrical and Computer Engineering, 2006. ICECE '06. International Conference on*, 2006, pp. 408-411.
- [62] J. C. Geyer, *et al.*, "Efficient frequency domain chromatic dispersion compensation in a coherent Polmux QPSK-receiver," in *Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC)*, 2010, pp. 1-3.
- [63] T. Xu, "Digital Dispersion Equalization and Carrier Phase Estimation in 112-Gbit/s Coherent Optical Fiber Transmission System," Royal Institute of Technology, Stockholm, Sweden, 2011.
- [64] G. Katz, *et al.*, "Electrical Dispersion Compensation Equalizers in Optical Direct and Coherent-Detection Systems," *Communications, IEEE Transactions on*, vol. 54, pp. 2045-2050, 2006.
- [65] G. Katz, *et al.*, "Electrical dispersion compensation equalizers in optical long-haul coherent-detection system," in *Transparent Optical Networks, 2005, Proceedings of 2005 7th International Conference*, 2005, pp. 80-83 Vol. 2.
- [66] B. Mariani, *et al.*, "Blind constant modulus equalization via convex optimization," *Signal Processing, IEEE Transactions on*, vol. 51, pp. 805-818, 2003.
- [67] I. Fatadin, *et al.*, "Laser Linewidth Tolerance for 16-QAM Coherent Optical Systems Using QPSK Partitioning," *Photonics Technology Letters, IEEE*, vol. 22, pp. 631-633, 2010.
- [68] T. F. Portela, *et al.*, "Analysis of signal processing techniques for optical 112 Gb/s DP-QPSK receivers with experimental data," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 10, pp. 155-164, 2011.
- [69] R. A. Soriano, *et al.*, "Chromatic Dispersion Estimation in Digital Coherent Receivers," *Lightwave Technology, Journal of*, vol. 29, pp. 1627-1637, 2011.

Apêndice A – Cálculo teórico da SER e BER

O apêndice encontra-se dividido em duas partes, na primeira deduz-se as relações entre a SER e a SNR para os formatos BPSK, QPSK, 16QAM, 32QAM e 64QAM, na segunda a aproximação para as relações entre a BER e a SNR. O estudo realizado tem como foco a transição dos símbolos entre zonas de decisão e como tal no posicionamento dos pontos na constelação. Para efeitos de cálculo considera-se que o ruído introduzido é gaussiano de média nula e variância $\sigma^2 = \frac{N_0}{2}$. Assim sendo, a função densidade de probabilidade do ruído pode escrever-se:

$$f(n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-n^2}{2\sigma^2}\right)}. \quad (\text{A.1})$$

Por outro lado, considera-se que os símbolos estão espaçados de d e que o ruído é introduzido segundo os dois versores das constelações n_1 e n_2 . Dos formatos de modulação considerados o mais simples é o BPSK, um exemplo da sua constelação é apresentado na Figura A.1.

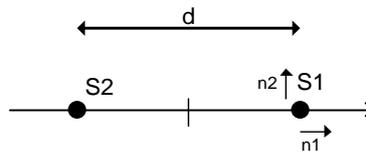


Figura A.1: Exemplo ilustrativo da constelação do formato de modulação BPSK.

É possível verificar na Figura A.1 que para este caso de estudo é particularmente importante o ruído introduzido segundo o versor n_1 , visto que a constelação não tem nenhuma zona de decisão em n_2 . Assim a probabilidade de erro pode escrever-se:

$$P_e = P(s_1)P(s_2/s_1) + P(s_2)P(s_1/s_2). \quad (\text{A.2})$$

Assumindo que os símbolos são igualmente prováveis, $P(s_1) = P(s_2) = \frac{1}{2}$, e que o ruído segundo os versores n_1 e n_2 é independente pode escrever-se:

$$P_e = \frac{1}{2} [P(s_2/s_1) + P(s_1/s_2)] = P(s_2/s_1). \quad (\text{A.3})$$

Considerando a função densidade de probabilidade, equação (A.1), a probabilidade de erro de símbolo toma o valor:

$$P_e = P(s_2 / s_1) = \int_{-\infty}^{\frac{d}{2}} f(n_1) dn_1 = \frac{1}{\sqrt{2\pi\sigma^2}} \sqrt{2\sigma^2} \int_{-\infty}^{\frac{d}{2\sqrt{2\sigma^2}}} e^{-t^2} dt = \frac{1}{2} \operatorname{erfc}\left(\frac{d}{2\sqrt{2\sigma^2}}\right). \quad (\text{A.4})$$

A equação (A.4) não é muito útil na prática uma vez que não traduz a probabilidade de erro em função da relação entre a potência do sinal e do ruído. Para isso, é necessário calcular a energia média por símbolo, $E_s = \frac{1}{M} \sum_{i=1}^M \|s_i\|^2$. Esta toma o valor

$E_s = \frac{1}{2} \left[\frac{d^2}{2} + \frac{d^2}{2} \right] = \frac{d^2}{2}$, resultando $d = 2\sqrt{E_s}$. Substituindo a relação obtida na equação (A.4) obtém-se:

$$P_e = \frac{1}{2} \operatorname{erfc}\left(\frac{d}{2\sqrt{2\sigma^2}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_s}{2\sigma^2}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_s}{N_0}}\right) = \frac{1}{2} \operatorname{erfc}(\sqrt{SNR}). \quad (\text{A.5})$$

O próximo formato de modulação a estudar é o QPSK, um exemplo da sua constelação encontra-se na Figura A.2.

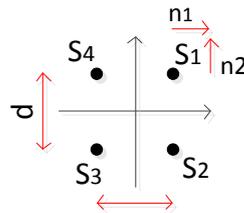


Figura A.2: Exemplo ilustrativo da constelação do formato de modulação QPSK.

Tal como no formato anterior, os símbolos estão igualmente espaçados de d e o ruído é adicionado a cada eixo da constelação, sendo estes representados pelos versores n_1 e n_2 . Para este caso os dois eixos da constelação são igualmente importantes. Deve notar-se que neste caso torna-se muito complexo o cálculo da probabilidade de erro diretamente, visto que, cada símbolo pode permanecer em quatro zonas de decisão. Para simplificar o cálculo opta-se por calcular a probabilidade dos símbolos serem recebidos corretamente e partindo desta calcular então a probabilidade de erro:

$$P_e = 1 - P_c. \quad (\text{A.6})$$

A probabilidade de receber os símbolos corretamente é dada pela equação (A.7), em que $P(S_i)$ corresponde à probabilidade do símbolo i e $P(C/S_i)$ a probabilidade de receber corretamente o símbolo i .

$$P_c = \sum_{i=1}^4 P(S_i)P(C/S_i) \quad (\text{A.7})$$

Assumindo que os símbolos são igualmente prováveis, $P(S_i) = \frac{1}{4}$, a probabilidade dos símbolos serem recebidos corretamente é dada por:

$$\begin{aligned} P_c &= \frac{1}{4} \sum_{i=1}^4 P(C/S_i) = P(C/S_i) = P\left(n_1 > -\frac{d}{2}\right)P\left(n_2 > -\frac{d}{2}\right) \\ &= \int_{-\frac{d}{2}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n_1^2}{2\sigma^2}} dn_1 \int_{-\frac{d}{2}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n_2^2}{2\sigma^2}} dn_2 \\ &= \left(1 - \frac{1}{2} \operatorname{erfc}\left(\frac{d}{2\sqrt{2\sigma^2}}\right)\right)^2. \end{aligned} \quad (\text{A.8})$$

Considerando a relação em (A.6), a probabilidade de erro pode escrever-se:

$$\begin{aligned} P_e &= 1 - \left[1 - \frac{1}{2} \operatorname{erfc}\left(\frac{d}{2\sqrt{2\sigma^2}}\right)\right]^2 \\ &= \operatorname{erfc}\left(\frac{d}{2\sqrt{2\sigma^2}}\right) - \frac{1}{4} \operatorname{erfc}\left(\frac{d}{2\sqrt{2\sigma^2}}\right)^2. \end{aligned} \quad (\text{A.9})$$

O resultado obtido em (A.9) não tem em consideração a SNR, para isso tem de se calcular a energia de símbolo (A.10). Do resultado apresentado em (A.10) obtém-se a relação $d = \sqrt{2E_s}$. Substituindo em (A.9), a probabilidade de erro para o formato QPSK pode escrever-se como na equação (A.11).

$$E_s = \sum_{i=1}^4 E_i = \frac{d^2}{2} \quad (\text{A.10})$$

$$P_e = \operatorname{erfc}\left(\sqrt{\frac{\operatorname{SNR}}{2}}\right) - \frac{1}{4} \operatorname{erfc}\left(\sqrt{\frac{\operatorname{SNR}}{2}}\right)^2 \quad (\text{A.11})$$

O terceiro caso de estudo é o formato de modulação 16QAM, um exemplo da sua constelação encontra-se na Figura A.3.

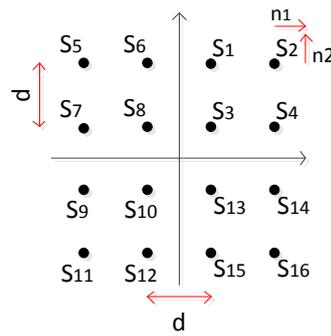


Figura A.3: Exemplo ilustrativo da constelação do formato de modulação 16QAM.

Tal como no caso anterior procede-se ao cálculo da probabilidade dos símbolos serem recebidos corretamente e depois recorrendo à equação (A.6) calcula-se a probabilidade de erro. Visto que a função densidade de probabilidade pode representar-se por (A.1), a probabilidade dos símbolos serem recebidos corretamente pode escrever-se:

$$P_c = 1 - \frac{3}{2} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma}\right) + \frac{9}{16} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma}\right)^2, \quad (\text{A.12})$$

atendendo a (A.6) obtém-se:

$$P_e = \frac{3}{2} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma}\right) - \frac{9}{16} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma}\right)^2. \quad (\text{A.13})$$

Calculando a energia média de símbolo (A.14), obtém-se a relação $d = \sqrt{\frac{2}{5} E_s}$.

Substituindo esta relação na equação (A.13) a probabilidade de erro para o formato de modulação 16QAM pode-se escrever como na equação (A.15). No caso deste formato foram omitidos alguns cálculos intermédios visto que são idênticos aos casos anteriores. O mesmo será considerado nos casos seguintes.

$$E_s = \sum_{i=1}^{16} E_i = \frac{5d^2}{2} \quad (\text{A.14})$$

$$P_e = \frac{3}{2} \operatorname{erfc}\left(\sqrt{\frac{\text{SNR}}{10}}\right) - \frac{9}{16} \operatorname{erfc}\left(\sqrt{\frac{\text{SNR}}{10}}\right)^2 \quad (\text{A.15})$$

Um exemplo da constelação do formato de modulação 32QAM é apresentado na Figura A.4.

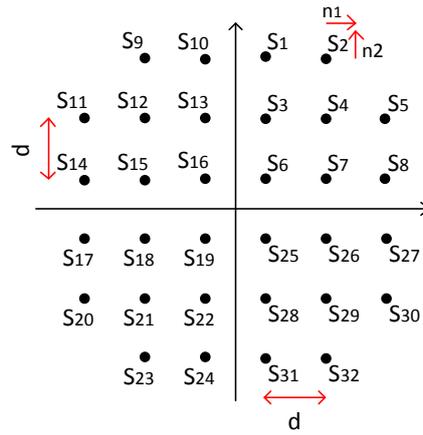


Figura A.4: Exemplo ilustrativo da constelação do formato de modulação 32QAM.

Recorrendo ao mesmo método do caso anterior é possível calcular a probabilidade dos símbolos serem recebidos corretamente:

$$P_c = 1 - \frac{25}{16} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right) + \frac{11}{16} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right)^2. \quad (\text{A.16})$$

Recorrendo à equação (A.6) obtém-se a equação da probabilidade de erro:

$$P_e = \frac{25}{16} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right) - \frac{11}{16} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right)^2. \quad (\text{A.17})$$

Para escrever a probabilidade de erro em função da SNR é necessário calcular a energia média de símbolo, $E_s = \frac{11}{2} d^2$, obtendo-se a relação $d = \sqrt{\frac{2}{11} E_s}$. Substituindo esta relação na equação (A.17) obtém-se a equação final da probabilidade de erro em função da SNR:

$$P_e = \frac{25}{16} \operatorname{erfc}\left(\sqrt{\frac{SNR}{22}}\right) - \frac{11}{16} \operatorname{erfc}\left(\sqrt{\frac{SNR}{22}}\right)^2. \quad (\text{A.18})$$

Por fim, o último formato de modulação a ser estudado é o 64QAM. Um exemplo da sua constelação é apresentado na Figura A.5.

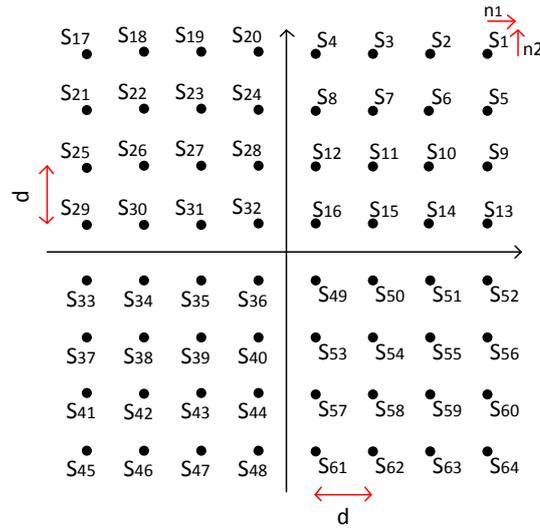


Figura A.5: Exemplo ilustrativo da constelação do formato de modulação 64QAM.

Os pontos da constelação estão igualmente espaçados de d . Tal como nos casos anteriores é possível obter a probabilidade dos símbolos serem recebidos corretamente:

$$P_c = 1 - \frac{7}{4} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right) + \frac{49}{64} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right)^2. \quad (\text{A.19})$$

Considerando (A.6) obtém-se a equação da probabilidade de erro:

$$P_e = \frac{7}{4} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right) - \frac{49}{64} \operatorname{erfc}\left(\frac{d}{2\sqrt{2}\sigma^2}\right)^2. \quad (\text{A.20})$$

A energia média de símbolo é dada por $E_s = \frac{21}{2}d^2$, obtendo-se a relação

$d = \sqrt{\frac{2}{21}E_s}$. Substituindo em (A.20) a equação final da probabilidade de erro em função da SNR pode escrever-se:

$$P_e = \frac{7}{4} \operatorname{erfc}\left(\sqrt{\frac{\text{SNR}}{42}}\right) - \frac{49}{64} \operatorname{erfc}\left(\sqrt{\frac{\text{SNR}}{42}}\right)^2. \quad (\text{A.21})$$

Concluído o estudo das funções de probabilidade de erro de símbolo para os vários formatos de modulação, na Figura A.6 apresenta-se as curvas que descrevem cada uma delas.

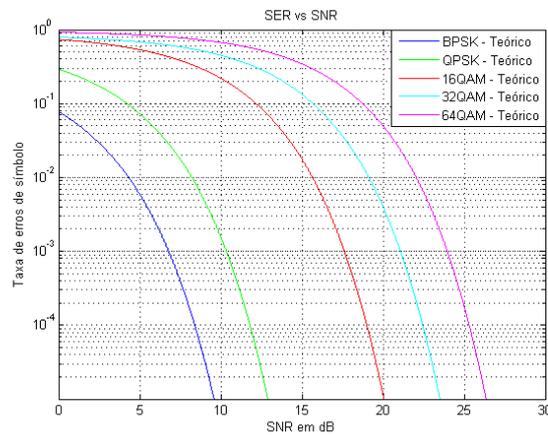


Figura A.6: Curvas ilustrativas da probabilidade de erro de símbolo em função da SNR para os formatos de modulação BPSK, QPSK, 16QAM, 32QAM e 64QAM.

Partindo das equações calculadas anteriormente é possível calcular as curvas de BER em função da SNR recorrendo a algumas aproximações. Considerando que os símbolos são codificados com código de *Gray* e que a SNR é relativamente elevada, pode considerar-se que apenas ocorre um erro de bit por símbolo. Esta aproximação diz-nos que a probabilidade de erro de bit será dada por:

$$P_{e,bit} = \frac{P_{e,simb}}{\log_2(M)}. \quad (A.22)$$

Na equação (A.22) M é o número de símbolos da constelação. Considerando esta aproximação obtém-se as curvas representadas na Figura A.7.

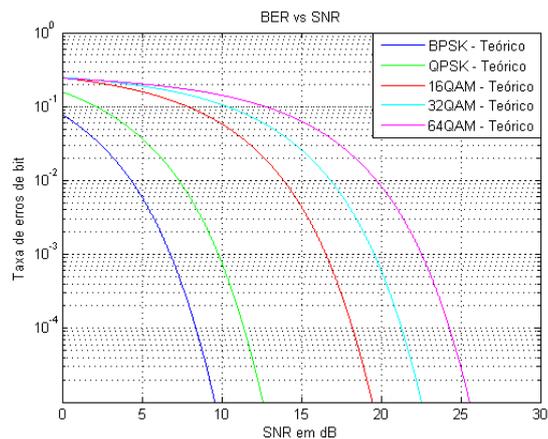


Figura A.7: Curvas ilustrativas da probabilidade de erro de bit em função da SNR para BPSK, QPSK, 16QAM, 32QAM e 64QAM.

Deve notar-se que para baixos valores de SNR pode ser necessário ajustar as relações obtidas. Será de esperar que para estes valores a aproximação não seja válida. Em todo o caso esta é uma boa referência para os estudos realizados.

Apêndice B – Descrição dos componentes adicionados

Neste apêndice são descritos todos os componentes adicionados na nova versão do simulador. No total são 31 componentes, sendo estes identificados ao longo do texto a negrito.

ADC

O objetivo da ADC é servir de transdutor entre o domínio analógico e digital. O funcionamento deste componente é definido de acordo com os seus parâmetros de configuração, Figura B.1.

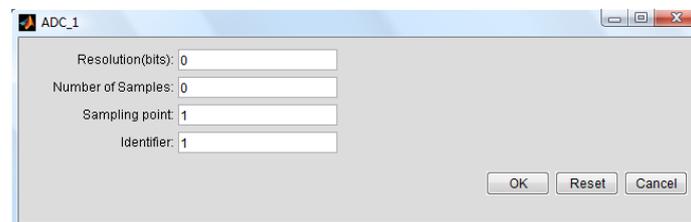


Figura B.1: Parâmetros de configuração da ADC no OSIP.

Na Figura B.1 o campo “*Resolution*” representa a resolução em *bits* da ADC, “*Number of Samples*” o número de amostras por símbolo, “*Sampling point*” o ponto de amostragem e “*Identifier*” o identificador do componente. Deve notar-se que o número de amostras a retirar por símbolo define a frequência de amostragem da ADC, enquanto que, o ponto de amostragem o ajuste do instante temporal de amostragem. O identificador permite armazenar as configurações do componente para que outros lhe possam aceder e configurarem-se automaticamente. O diagrama de funcionamento do componente é apresentado na Figura B.2.

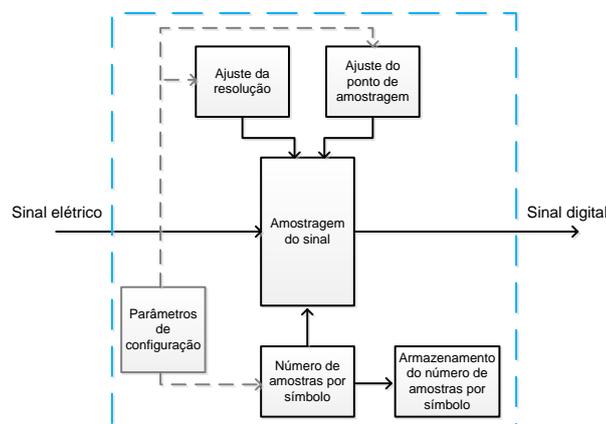


Figura B.2: Esquema de funcionamento da ADC no OSIP.

Após realizadas todas as configurações o circuito de amostragem é ajustado e o sinal digitalizado, sendo o sinal de saída normalizado entre -1 e 1. Deve notar-se que a periodicidade da amostragem é definida de acordo com o número de amostras do transmissor. Caso a modulação seja em intensidade esta, em amostras, é dada pela relação entre a frequência de amostragem global do simulador e o ritmo de *bits* caso contrário, é dada pela relação entre o número de amostras por símbolo no transmissor e na ADC, o primeiro é definido no componente “ReSample” e é carregado pela ADC utilizando o identificador. O ícone do componente no OSIP encontra-se na Figura B.3.



Figura B.3: Ícone da ADC no OSIP.

Amplifier

Este componente representa um amplificador de tensão ideal. A sua saída é dada pelo produto entre a entrada e o ganho definido nas configurações do componente, Figura B.4.

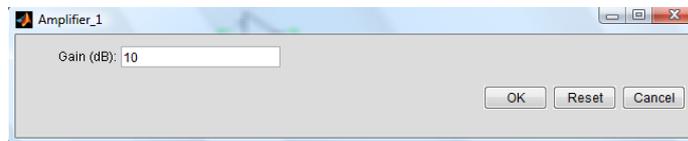


Figura B.4: Parâmetros de configuração do componente Amplifier no OSIP.

Este dispositivo pertence ao grupo dos componentes elétricos, o seu ícone no OSIP é apresentado na Figura B.5.

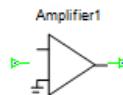


Figura B.5: Ícone do Amplifier no OSIP.

Coding

O componente de codificação permite mapear a informação binária, que deve estar paralelizada, na constelação pretendida e gerar os sinais analógicos para a sua modulação. Este apenas implementa o mapeamento com código de gray contudo, é possível alterar facilmente o tipo de código de mapeamento. O processo de codificação é realizado tendo em conta tabelas de correspondência logo basta alterar essas tabelas

para alterar o código. Os parâmetros de configuração do componente são apresentados na Figura B.6.

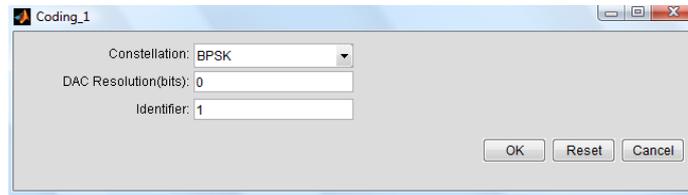


Figura B.6: Parâmetros de configuração do componente Coding no OSIP.

Para configurar o componente basta definir o formato de modulação desejado (BPSK, QPSK, 16QAM, 32QAM ou 64QAM), a resolução do conversor digital para analógico e o identificador do componente. O funcionamento do algoritmo é apresentado na Figura B.7.

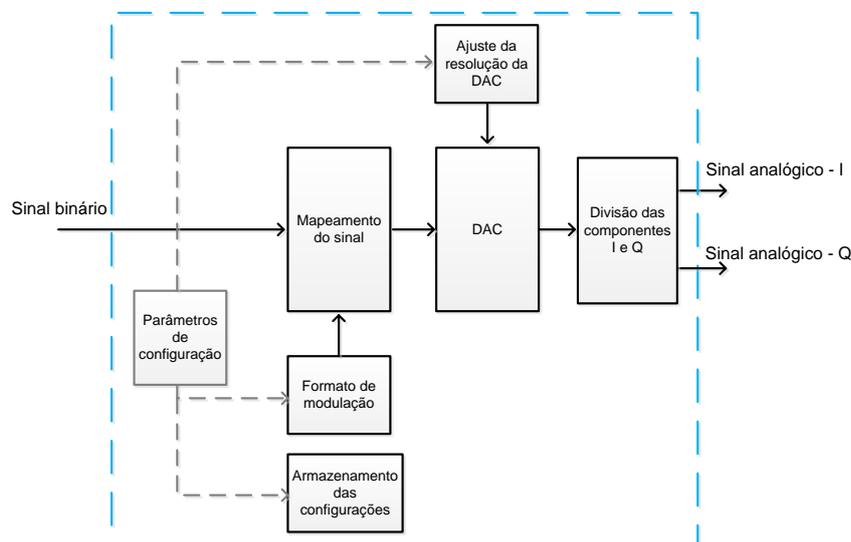


Figura B.7: Diagrama de funcionamento do componente Coding.

O algoritmo pode decompor-se em duas partes, na primeira o sinal é mapeado e são gerados os sinais com a máxima resolução e na segunda a resolução dos sinais à saída do componente é limitado à resolução da DAC. Estes sinais são normalizados entre 0 e 1, logo pode ser necessário recorrer a um amplificador para que a modulação seja realizada corretamente. O ganho do amplificador introduzido depende da tensão v_{π} do modulador. No algoritmo implementado considera-se que o mapeamento dos sinais é realizado de acordo com a Figura B.8. Deve notar-se que o formato da constelação do BPSK deve-se à simplificação do algoritmo implementado. Assim os formatos de ordem inferior podem ser identificados nos formatos de ordem mais elevada, facilitando a modulação e a versatilidade do algoritmo implementado. Por outro lado, o uso de um identificador permite armazenar os pontos ideais da constelação em variáveis temporárias de modo a que outros componentes durante a simulação lhe possam aceder. É exemplo o componente *UConstellation*.

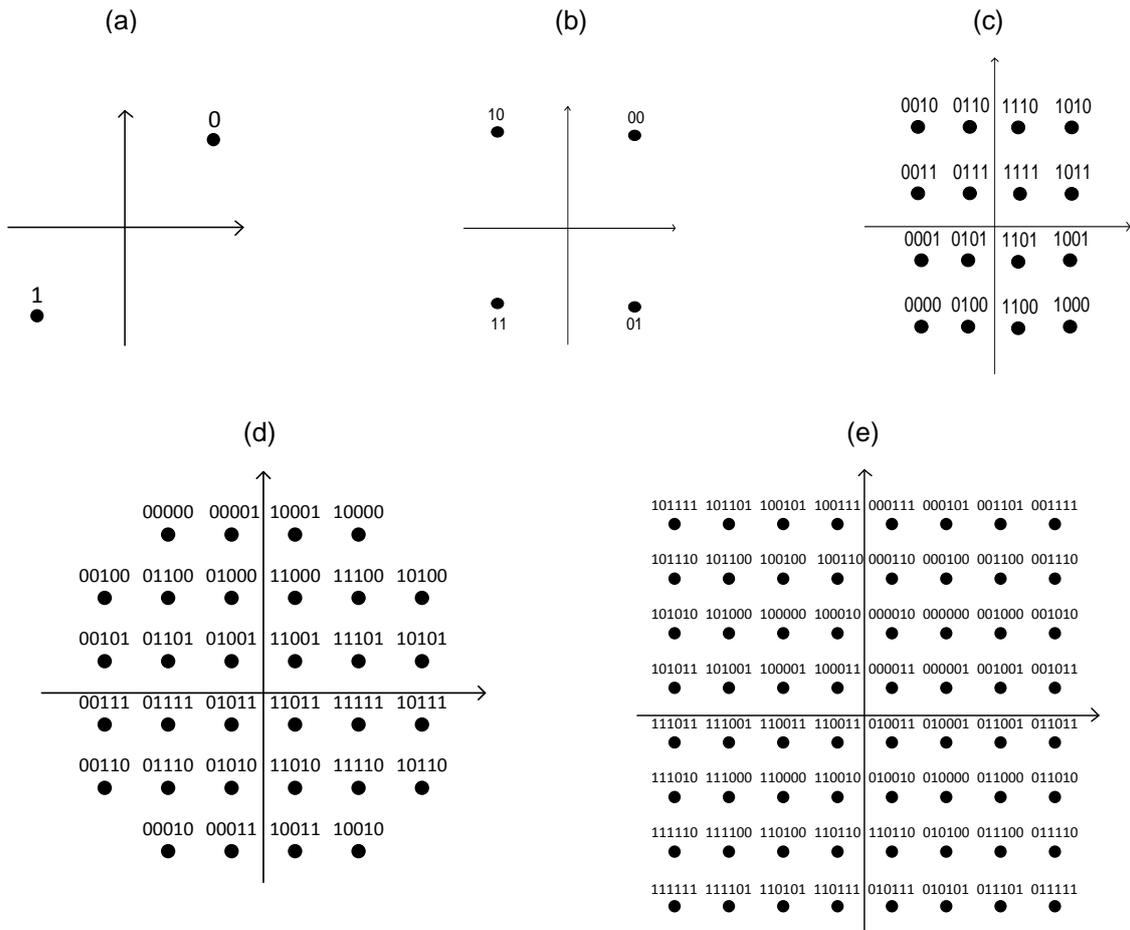


Figura B.8: Mapeamento da seqüência binária na constelação: (a) BPSK; (b) QPSK; (c) 16QAM; (d) 32QAM; (e) 64QAM.

O ícone do componente no OSIP é apresentado na Figura B.9. Como se pode verificar este tem uma entrada, para o sinal digital, e duas saídas, que correspondem às componentes em fase e quadratura do sinal modulante.

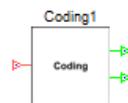


Figura B.9: Ícone do componente Coding no OSIP.

Decoding

Este componente realiza a operação inversa do Coding. Assim deverá receber o sinal complexo modulado e gerar a seqüência binária correspondente. Este é apenas configurado pelo identificador, Figura B.10, o qual permite o acesso aos parâmetros de configuração do componente de codificação e como tal ajusta automaticamente os circuitos de descodificação.



Figura B.10: Parâmetros de configuração do componente Decoding no OSIP.

Deve notar-se que a decodificação do sinal é realizada de acordo com a Figura B.8 e que a sequência binária é gerada em paralelo. O esquema do funcionamento do algoritmo implementado é apresentado na Figura B.11.

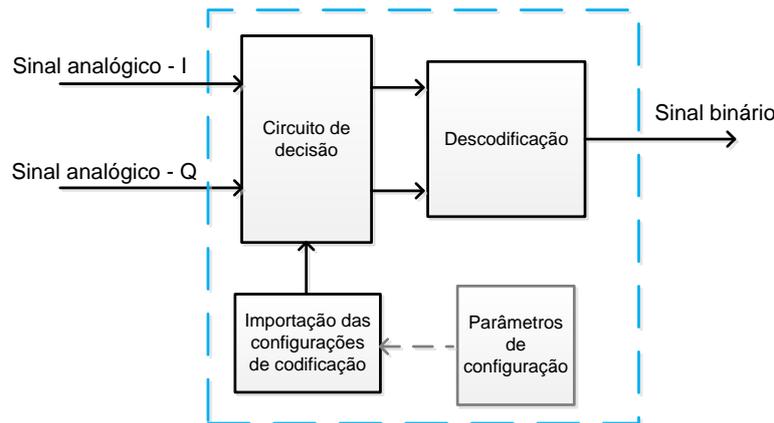


Figura B.11: Diagrama de funcionamento do componente Decoding.

O componente para além de possibilitar a receção do sinal nas suas componentes de fase e quadratura, permite também a sua receção na forma complexa ($I+JQ$), sendo o modo de funcionamento configurado acedendo às definições das suas portas. Se este apresentar duas entradas está configurado para receber as componentes de forma independente, se apresentar apenas uma entrada permite receber diretamente um sinal complexo. O ícone do componente no simulador encontra-se na Figura B.12.

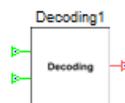


Figura B.12: Ícone do componente Decoding no OSIP.

Downsampling

Este componente permite reduzir o número de amostras por símbolo através da operação de decimação. Assim à saída do componente apenas é gerada uma amostra por símbolo permitindo a aplicação do sinal a um circuito de decodificação. A forma como são geradas as amostras depende da configuração do componente, os seus parâmetros de configuração são apresentados na Figura B.13.

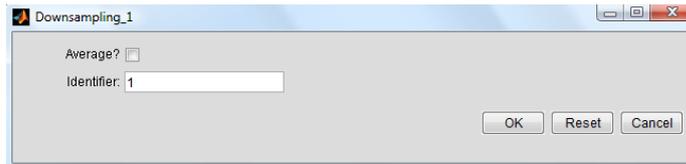


Figura B.13: Parâmetros de configuração do componente Downsampling.

O identificador permite importar o número de amostras por símbolo na ADC, definindo a janela de cada símbolo em amostras. Caso a opção “Average?” não esteja selecionada a saída corresponde à primeira amostra de cada símbolo caso contrário, a saída corresponde à média de todas as amostras do símbolo. O ícone do componente no simulador é apresentado na Figura B.14.

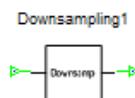


Figura B.14: Ícone do componente Downsampling no OSIP.

ReSample

Este componente permite facilitar o funcionamento do simulador para formatos de ordem mais elevada. Cabe-lhe a responsabilidade de realizar a amostragem do sinal no transmissor de acordo com a frequência de amostragem do simulador. Os parâmetros de configuração do componente são apresentados na Figura B.15.



Figura B.15: Parâmetros de configuração do componente ReSample.

Definindo um identificador o número de amostras por símbolo retiradas são armazenadas internamente para que outros componentes, com o mesmo identificador, lhe possam aceder. O campo “Sampling factor” permite definir o número de amostras por símbolo desejadas e deverá respeitar a relação entre a frequência de amostragem e o ritmo de símbolos. O ícone do componente no simulador encontra-se na Figura B.16.

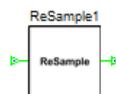


Figura B.16: Ícone do componente ReSample no OSIP.

DSNR

O objetivo deste componente é adicionar ruído aos sinais no domínio elétrico. A potência do ruído adicionado depende da relação entre a SNR à entrada e saída do componente e da potência do sinal de dados, estes que são os parâmetros de configuração do componente, Figura B.17. O valor da SNR adicionada ao sinal é apresentada na linha de comandos do Matlab®. Caso se pretenda realizar uma simulação com vários blocos apenas no primeiro é apresentado o valor da SNR do sinal, proporcionando um aumento da eficiência da simulação.

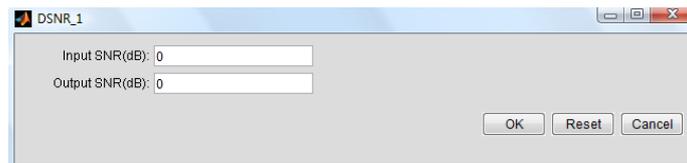


Figura B.17: Parâmetros de configuração do componente DSNR.

De acordo com a configuração das suas portas, este pode funcionar de dois modos distintos. Estes diferenciam-se pela aquisição do sinal sob a forma complexa ou nas componentes I e Q de forma independente. Assim são admissíveis dois cenários: uma ou duas entradas e saídas. A configuração das portas do componente pode ser realizada selecionando nas suas opções “*port definitions*”. O ícone do componente no simulador é apresentado na Figura B.18.



Figura B.18: Ícone do componente DSNR no OSIP.

DSP

A implementação deste componente permite aplicar algoritmos de compensação digital aos sinais no recetor. Após desenvolver o código Matlab® com os respetivos algoritmos o utilizador deve especificar o seu caminho no componente e atribuir um identificador, Figura B.19.

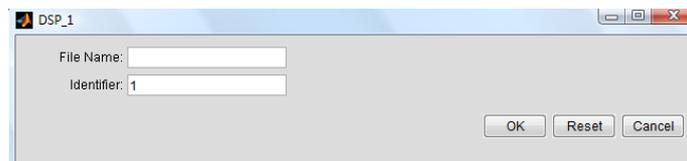


Figura B.19: Parâmetros de configuração do componente DSP.

Caso o ficheiro especificado se encontre na área de trabalho basta indicar o seu nome, caso contrário é necessário indicar o caminho completo para o ficheiro. O

identificador vai permitir a passagem de parâmetros entre o componente e o código implementado. Na verdade, este deve seguir o modelo apresentado na Figura B.20.

(a)	(b)
<pre>function testeDSP_single global DSP Input1 = DSP(id).IX; Input2 = DSP(id).QX; f = DSP(1).f; DSP(id).Xout = Input1 + 1j*Input2; return</pre>	<pre>function testeDSP_dual global DSP InputX1 = DSP(id).IX; InputX2 = DSP(id).QX; InputY1 = DSP(id).IY; InputY2 = DSP(id).QY; f = DSP(id).f; DSP(id).Xout = InputX1 + 1j*InputX2; DSP(id).Yout = InputY1 + 1j*InputY2; return</pre>

Figura B.20: Modelo de implementação dos algoritmos para o DSP: (a) – para uma polarização; (b) – para duas polarizações; id - identificador.

Na Figura B.20 são apresentados dois modelos, o primeiro para sistemas com uma única polarização e o segundo para diversidade de polarização. Como se pode verificar o identificador permite a passagem dos sinais de entrada e a respetiva devolução de forma eficiente. Adicionalmente o utilizador pode aceder à escala de frequências de simulação. Estes dois métodos implicam diferentes configurações do componente. Para alterar o modo de funcionamento basta alterar o número de portas do componente acedendo às suas opções e selecionando as definições das portas. Deve notar-se que o sinal na saída do DSP é complexo (I+jQ), contrariamente às suas entradas. O ícone do componente no simulador encontra-se na Figura B.21.

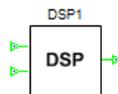


Figura B.21: Ícone do componente DSP no OSIP.

O ícone apresenta o caso em que o DSP recebe um sinal nas suas componentes de fase e quadratura e retorna o sinal complexo compensado. Caso o DSP tenha de lidar com diversidade de polarização são necessárias quatro entradas e duas saídas.

Inverter

Este componente permite aplicar ao sinal um ganho de -1, invertendo-o. Torna-se particularmente importante quando se usam formatos de modulação de ordem mais elevada, visto que os MZMs implementados necessitam de duas tensões de controlo, que podem ser simétricas. O ícone do componente no simulador encontra-se na Figura B.22.

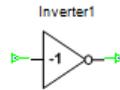


Figura B.22: Ícone do componente Inverter no OSIP.

PtoS

Este componente permite converter a informação binária paralela para uma sequência série, de acordo com o número de bits definido. A Figura B.23 apresenta um exemplo para um sinal binário que é recebido em sequências paralelas de 2 *bits*.

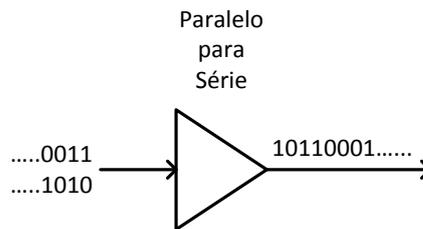


Figura B.23: Exemplo da conversão para série de um sinal binário em sequências paralelas de 2 bits.

Este componente torna-se particularmente importante para utilizar após o componente de decodificação, assim a informação binária decodificada pode ser convertida para uma sequência série e posteriormente analisada. Para configurar o componente basta definir o seu identificador, Figura B.24. Através do seu valor são importadas as configurações do componente “StoP”, configurando-se automaticamente.



Figura B.24: Parâmetros de configuração do componente PtoS.

A Figura B.25 ilustra o ícone do componente no simulador. Deve notar-se que este apenas tem uma entrada e uma saída, assim a informação paralela é passada por uma única ligação.

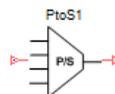


Figura B.25: Ícone do componente PtoS no OSIP.

StoP

Este componente permite converter uma sequência binária série numa de n bits paralelos. Um exemplo da sua operação para 2 bits é apresentado na Figura B.26.

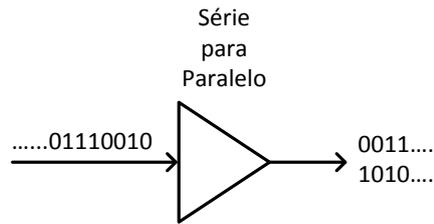


Figura B.26: Exemplo da paralelização de um sinal binário em sequências de 2 bits.

O seu funcionamento é muito simples e torna-se particularmente importante ao ser utilizado antes do componente de codificação, de modo a paralelizar a informação. Para configurar o componente basta definir dois parâmetros, Figura B.27. O primeiro define o número de bits a paralelizar e o segundo o identificador do componente. O seu valor vai permitir armazenar o número de bits em paralelo de modo a poder ser utilizado em outros componentes.



Figura B.27: Parâmetros de configuração do componente StoP.

O ícone do componente no simulador encontra-se na Figura B.28. Deve notar-se que o componente apenas tem uma saída, como tal a informação paralelizada é passada por uma única ligação.

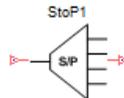


Figura B.28: Ícone do componente StoP no OSIP.

TIA

O objetivo deste componente é implementar um amplificador de trans-impedância ideal. Na verdade, este apenas realiza a normalização do sinal e como tal o sinal na sua saída estará compreendido entre -1 e 1. A Figura B.29 mostra o ícone do componente no OSIP.

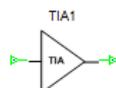


Figura B.29: Ícone do componente TIA no OSIP.

PolAlign

Este componente permite alinhar as polarizações do sinal. A sua implementação torna-se fundamental visto que da compensação da dispersão dos modos de polarização existe a incerteza da porta de saída para que converge cada uma das polarizações. Desta forma, este componente permite detetar em que porta está cada polarização e alinhá-las nas saídas corretas. O esquema de funcionamento do algoritmo implementado encontra-se na Figura B.30.

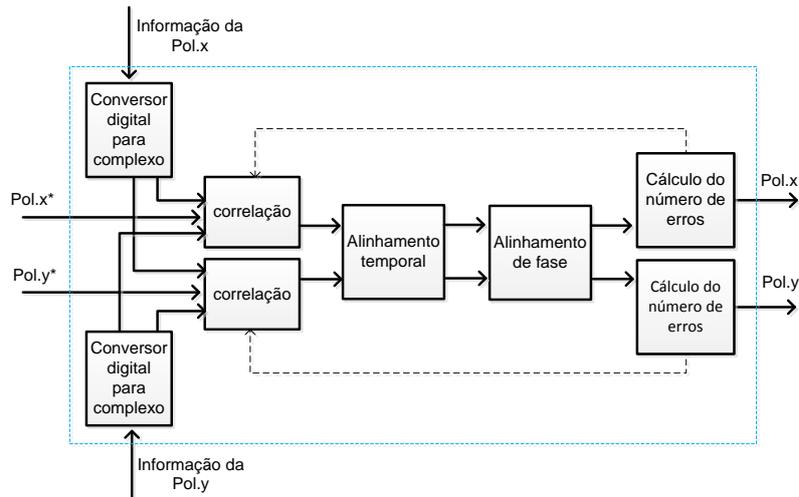


Figura B.30: Diagrama de funcionamento do componente PolAlign.

O componente recebe os sinais binários transmitidos em cada polarização e converte-os para símbolos complexos. De seguida estes são correlacionados com os sinais de dados de cada polarização, sendo alinhados temporalmente e corrigida a fase inicial. A deteção da polarização é realizada de acordo com o mínimo de erros em cada uma das saídas, sendo para o efeito utilizado um sinal de realimentação de acordo com o número de erros. De acordo com este sinal é possível alterar os sinais de comparação, sendo rodadas as polarizações. O ícone do componente no OSIP encontra-se na Figura B.31.

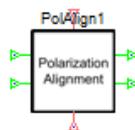


Figura B.31: Ícone do componente PolAlign no OSIP.

IQ_Mod

Este componente permite modular o sinal ótico com a informação binária desejada. Na verdade a informação binária deve ser primeiro codificada de modo a gerar os sinais

de controlo ao modulador, sendo utilizado para o efeito o componente “Coding”. O esquema de implementação do componente encontra-se na Figura B.32.

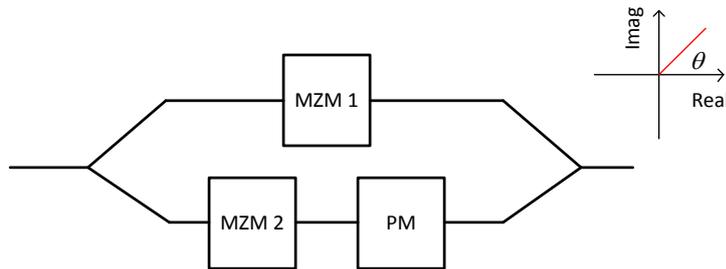


Figura B.32: Modelo de implementação do modulador IQ.

De uma forma geral, o feixe ótico é dividido em dois sinais de igual potência sendo cada um deles propagado num braço diferente. Cada um dos sinais resultantes é modulado por um MZM e posteriormente combinado. Deve notar-se que um dos braços é rodado de uma fase arbitrária, normalmente -90° , para permitir a modulação nas componentes de fase e de quadratura. Desta forma, a função de transferência do componente é a apresentada na equação (2.7). As tensões de controlo dos MZMs, assim como as suas tensões de polarização, e do PM são os parâmetros de configuração do componente, Figura B.33. Adicionalmente, pode definir-se também as perdas por inserção e o valor da sua tensão v_π .

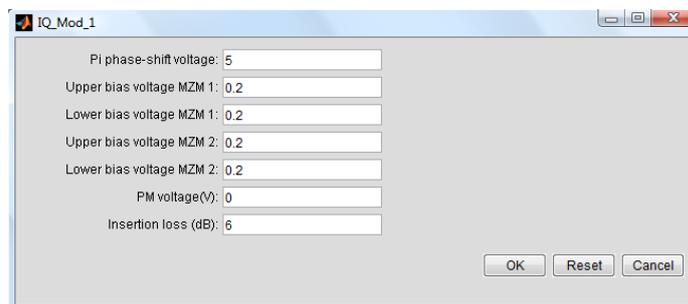


Figura B.33: Parâmetros de configuração do componente IQ_Mod.

O ícone do componente no OSIP encontra-se na Figura B.34. Este apresenta 4 entradas de controlo, que são destinadas aos MZMs, uma entrada para o feixe ótico e uma saída para o sinal ótico modulado.

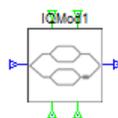


Figura B.34: Ícone do componente IQ_Mod no OSIP.

BPD

Este componente permite converter o sinal ótico resultante do 2x4 90° híbrido em componentes elétricas. Desta forma é possível representar as componentes de fase e quadratura dos sinais no domínio elétrico. A sua função de transferência é apresentada na equação (B.1).

$$I_{out} = R(E_{in1}E_{in1}^* - E_{in2}E_{in2}^*) \quad (B.1)$$

Na equação (B.1) E_{in1} e E_{in2} representam os campos óticos resultantes do recetor híbrido e R a responsividade em [A/W]. Deve notar-se que a saída do componente é em corrente e como tal pode ser necessário a sua conversão para níveis de tensão. O parâmetro de configuração do componente encontra-se na Figura B.35.

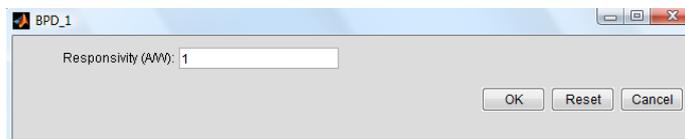


Figura B.35: Parâmetro de configuração do componente BPD.

Para configurar o componente basta definir qual o valor da sua responsividade. O ícone do componente no simulador é apresentado na Figura B.36.

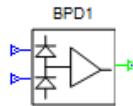


Figura B.36: Ícone do componente BPD no OSIP.

DGD

O objetivo deste componente é introduzir um atraso entre as componentes de polarização. Este valor é definido nos parâmetros de configuração do componente e deve estar em μs , Figura B.37. A sua função de transferência é apresentada na equação (B.2) [69].



Figura B.37: Parâmetro de configuração do componente DGD.

$$E_{out} = \begin{bmatrix} e^{j2\pi f\tau/2} & 0 \\ 0 & e^{-j2\pi f\tau/2} \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix} \quad (B.2)$$

Na equação (B.2) f representa a frequência da portadora e τ o atraso entre as polarizações. Deve notar-se que para aplicar esta função de transferência é necessário primeiro dividir as polarizações e posteriormente voltar a combiná-las. A Figura B.38 representa o ícone do componente no simulador.

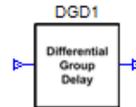


Figura B.38: Ícone do componente DGD no OSIP.

DOSNR

O objetivo deste componente é adicionar ruído aos sinais no domínio ótico. A potência do ruído adicionado depende da relação entre a OSNR à entrada e saída do componente e da potência do sinal de dados, estes que são os parâmetros de configuração do componente, Figura B.39. O valor da OSNR adicionada ao sinal é apresentada na linha de comandos do Matlab®. Caso se pretende realizar uma simulação com vários blocos apenas no primeiro é apresentado o valor da OSNR do sinal, proporcionando um aumento da eficiência da simulação.

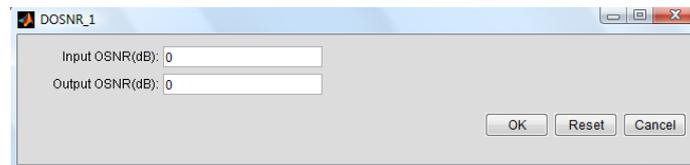


Figura B.39: Parâmetros de configuração do componente DOSNR.

O ícone do componente no simulador é apresentado na Figura B.40.

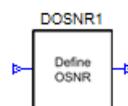


Figura B.40: Ícone do componente DOSNR no OSIP.

Optical90Hybrid

Este componente permite detetar as componentes de fase e quadratura do sinal, tornando-se assim num componente muito importante quando se pretende utilizar formatos de modulação de ordem mais elevada. O ícone do componente no simulador é apresentado na Figura B.41. Deve notar-se que este tem duas entradas e quatro saídas.

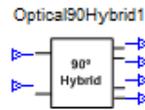


Figura B.41: Ícone do componente Optical90Hybrid no OSIP.

As duas entradas do componente correspondem ao sinal de dados e ao laser local e as quatro saídas o resultado do seu batimento. A função de transferência do componente é apresentada na equação (2.23). Neste caso consideram-se as fases $\psi_{11} - \psi_{12} = \psi_{21} - \psi_{22} = \psi_{31} - \psi_{32} = \psi_{41} - \psi_{42} = \psi$ nulas. Desta forma, selecionando as saídas corretas e utilizando fotodíodos balanceados é possível obter as componentes de fase e quadratura do sinal no domínio elétrico.

PBC_S

Este componente apresenta dois modos de operação, pode funcionar como um divisor de polarizações (PBS), Figura B.42–a), ou combinador de polarizações (PBC), Figura B.42–b). O modo de funcionamento do componente pode ser alterado acedendo às opções do componente e configurando as portas de entrada e saída. Caso este apresente duas entradas e uma saída funciona como PBC e no caso de ter uma entrada e duas saídas opera como PBS. Não é permitida mais nenhuma combinação entre entradas e saídas. Para o funcionamento do componente considera-se que a primeira porta corresponde sempre à polarização x, independentemente deste estar a funcionar como PBC ou PBS.

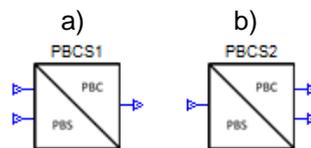


Figura B.42: Ícone do componente PBC_S no OSIP: (a) - PBC; (b) - PBS.

PolarizationRot

O objetivo deste componente é provocar rotações das componentes de polarização. Para esse fim deve ser configurado com os ângulos de rotação θ e φ , Figura B.43.

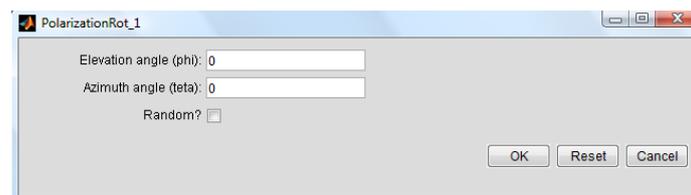


Figura B.43: Parâmetros de configuração do componente PolarizationRot.

Na Figura B.43 θ e φ representam os ângulos de rotação segundo a horizontal e a vertical respetivamente. Este componente torna-se assim bastante importante para simular efeitos como a PMD. Este em conjugação com o componente “DGD” permitem simular a PMD de primeira ordem. A sua função de transferência é apresentada na equação (B.3) [35, 69].

$$E_{out} = \begin{bmatrix} e^{j\varphi/2} & 0 \\ 0 & e^{-j\varphi/2} \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix} \quad (\text{B.3})$$

Deve notar-se que para aplicar a equação (B.3) é necessário dividir as duas componentes de polarização do sinal e posteriormente voltar a combiná-las. Por outro lado, o ângulo θ está uniformemente distribuído entre 0 e 2π , enquanto que, φ entre $-\pi/2$ e $\pi/2$ [35]. A Figura B.44 apresenta o ícone do componente no simulador.

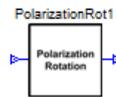


Figura B.44: Ícone do componente PolarizationRot no OSIP.

Polarizer

Este componente permite selecionar uma polarização do sinal. No fundo pode ser visto como um filtro ideal de polarizações. Por defeito a polarização na saída do componente é sempre a x. Assim, caso o utilizador pretenda preservar a polarização y é necessário proceder à rotação das polarizações, esta que pode ser realizada pelo componente “PolarizationRot”, antes de utilizar este componente. O ícone do componente é apresentado na Figura B.45.

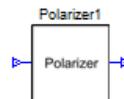


Figura B.45: Ícone do componente Polarizer no OSIP.

WDM_mux

O objetivo deste componente é combinar um conjunto de canais WDM. O número canais é definido pelo número de entradas do componente, que pode ser alterado nas suas opções. Os parâmetros de configuração do componente são apresentados na Figura B.46.

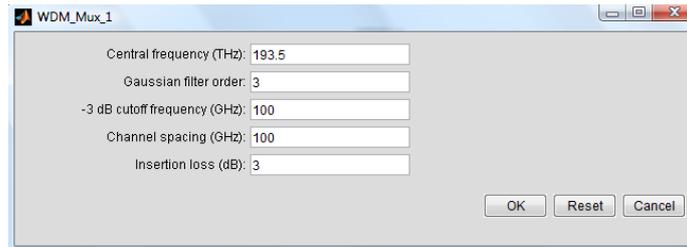


Figura B.46: Parâmetros de configuração do componente WDM_Mux.

Tendo em conta os parâmetros da Figura B.46 e o número de entradas do componente são criadas um conjunto de portadoras, correspondendo cada uma delas a um canal. São também criados filtros de característica gaussiana centrados em cada uma das portadoras. Definidas as condições de multiplexagem, as várias entradas são filtradas e multiplexadas. Deve notar-se que a primeira entrada do componente é sempre considerada como a que tem a frequência da portadora mais baixa. A Figura B.47 apresenta o diagrama com o funcionamento referido.

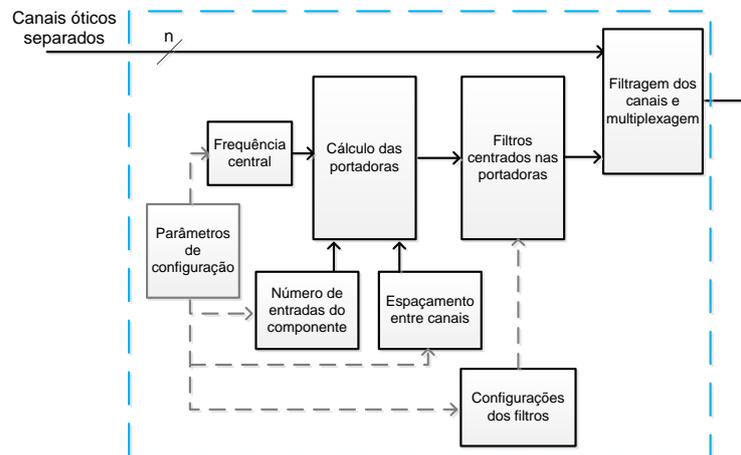


Figura B.47: Princípio de funcionamento do algoritmo do componente WDM_Mux.

Na Figura B.47 não apresenta as perdas. Na verdade o utilizador pode definir as perdas, sendo estas aplicadas após a multiplexagem dos canais. O ícone do componente no simulador encontra-se na Figura B.48.

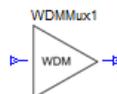


Figura B.48: Ícone do componente WDM_Mux no OSIP.

WDM_demux

Este componente é muito semelhante ao “*WDM_Mux*”, na verdade aplica a função inversa. Da mesma forma que este, o número de canais a desmultiplexar é definido pelo número de saídas do componente. Os seus parâmetros de configuração encontram-se na Figura B.49.

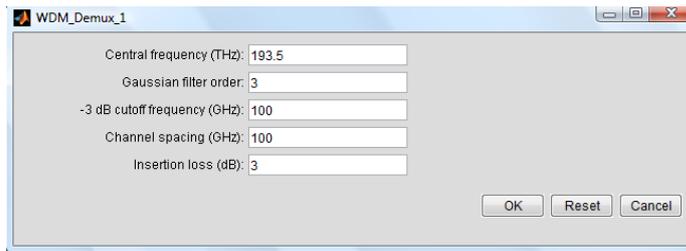


Figura B.49: Parâmetros de configuração do componente WDM_Demux.

Dado o número de saídas do componente, a frequência central e o espaçamento entre canais são definidas um conjunto de portadoras, correspondendo cada uma a um canal, sendo que a de menor frequência está associada à primeira saída do componente. Centrada em cada portadora é depois definido um filtro gaussiano cujos parâmetros são especificados pelo utilizador. Da filtragem de cada zona espectral resulta a separação dos canais e como tal a sua desmultiplexagem. Este processo é apresentado no diagrama da Figura B.50.

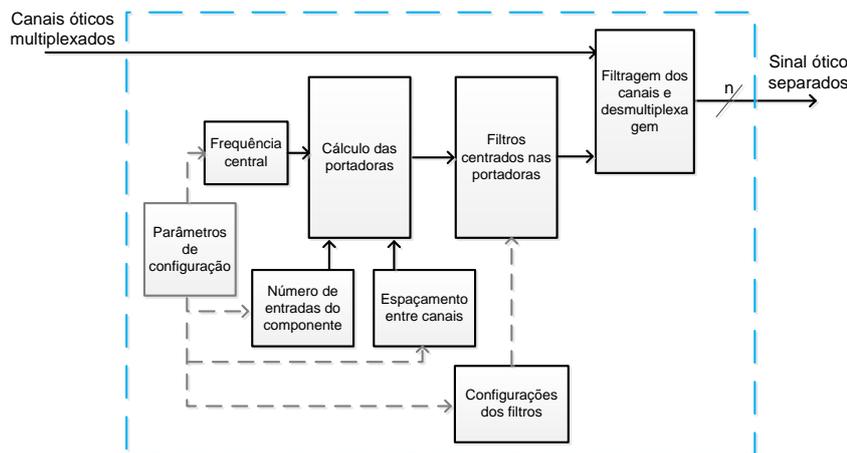


Figura B.50: Diagrama de funcionamento do algoritmo do componente WDM_Demux.

Deve notar-se que no esquema da Figura B.50 não são apresentadas as perdas do componente. O utilizador pode defini-las nos seus parâmetros de configuração (ver Figura B.49) e estas são aplicadas após a separação dos canais. O ícone do componente no simulador encontra-se na Figura B.51.

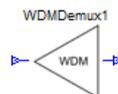


Figura B.51: Ícone do componente WDM_Demux no OSIP.

PDL

Este componente permite simular as perdas dependentes da polarização. Os seus parâmetros de configuração encontram-se na Figura B.52.

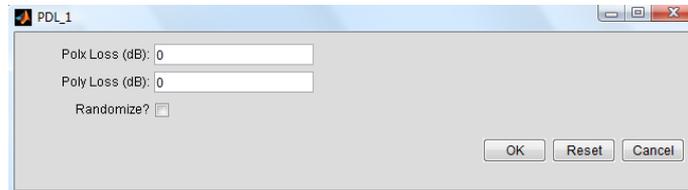


Figura B.52: Parâmetros de configuração do componente PDL.

Pela Figura B.52 pode-se verificar que o componente pode funcionar de duas formas distintas, isto é, o utilizador pode definir um valor de perdas fixo ou aleatório. Caso o valor seja aleatório, o valor máximo admissível para as perdas é definido pelos parâmetros de configuração. Deve notar-se que as perdas estão na unidade *decibel*. O ícone do componente no simulador encontra-se na Figura B.53.

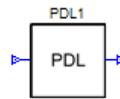


Figura B.53: Ícone do componente PDL no OSIP.

PhaseRot

Este componente permite realizar rotações de fase constantes no sinal. O seu uso é bastante útil para compensar rotações fixas do sinal, que sejam conhecidas, ou mesmo para testar algoritmos de estimação de fase, de que é exemplo o *Viterbi*. Para configurar o componente basta definir a rotação que se pretende realizar em radianos, Figura B.54.

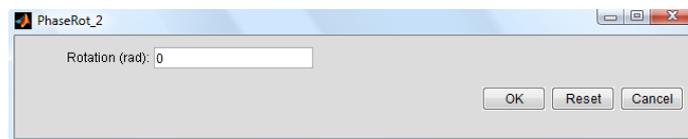


Figura B.54: Parâmetro de configuração do componente PhaseRot.

A Figura B.55 apresenta o ícone do componente no simulador.

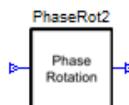


Figura B.55: Ícone do componente PhaseRot no OSIP.

BERT

O objetivo deste componente é analisar o sinal descodificado no recetor e o transmitido, detetando o número de erros cometidos. Este é apenas configurado com um identificador, Figura B.56, através do qual é possível armazenar os resultados na variável global *NUM*. Desta forma o número total de erros é armazenado no campo *NUM.BERTresults.Nerrors(identificador,1)* e a probabilidade de erro no *NUM.BERTresults.BER(identificador, nframes)*, onde *nframes* representa o número de blocos a simular. Estas servem de auxílio ao método de Monte-Carlo e permitem implementar o método de paragem da simulação à deteção do número de erros desejados.

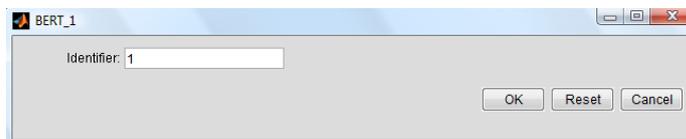


Figura B.56: Parâmetro de configuração do componente BERT.

O utilizador pode consultar os resultados obtidos após a simulação utilizando a linha de comandos do Matlab[®]. A probabilidade de erro é armazenada em cada iteração da simulação, representando uma tabela para cada identificador, e o número de erros é um valor cumulativo, como tal só existe um valor para cada identificador. O ícone do componente no simulador encontra-se na Figura B.57.



Figura B.57: Ícone do componente BERT no OSIP.

EVM

O objetivo deste componente é medir o EVM do sinal recebido. Para configurá-lo basta definir um identificador, Figura B.58. Com o uso desta variável é possível importar os pontos ideais da constelação que são definidos no componente de codificação. Desta forma o componente configura-se automaticamente e calcula o EVM de acordo com a equação (B.4) [61].

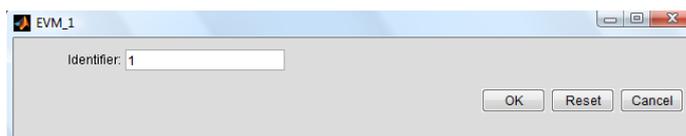


Figura B.58: Parâmetro de configuração do componente EVM.

$$EVM_{RMS} = \left[\frac{\frac{1}{T} \sum_{t=1}^T |I_t - I_{0,t}|^2 + |Q_t - Q_{0,t}|^2}{\frac{1}{N} \sum_{n=1}^N [(I_{0,n})^2 + (Q_{0,n})^2]} \right]^{\frac{1}{2}} \quad (\text{B.4})$$

Os resultados obtidos são armazenados na variável $NUM.EVM(\text{identificador}, nframes)$, onde $nframes$ representa o número de blocos a simular. Deve notar-se que os resultados são armazenados a cada iteração, sendo criada uma tabela para cada identificador. Estes podem ser consultados através da linha de comandos do Matlab[®]. O ícone do componente no simulador é apresentado na Figura B.59.

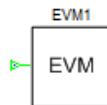


Figura B.59: Ícone do componente EVM no OSIP.

SavetoFile

Em alguns casos o utilizador pode estar interessado em armazenar os sinais para realizar análises posteriores. Um bom exemplo é armazenar os resultados após o circuito de amostragem para implementar posteriormente algoritmos de processamento digital. Este componente permite armazenar qualquer sinal do simulador, sendo apenas necessário definir o nome do ficheiro e o seu formato, Figura B.60.



Figura B.60: Parâmetros de configuração do componente SavetoFile.

Os ficheiros podem ser armazenados em dois formatos: *ascii* e *mat*. Caso seja apenas identificado o nome pretendido para o ficheiro este é armazenado na área de trabalho. Caso o utilizador pretenda guardar noutra local deve indicar o caminho completo para o ficheiro. De acordo com o formato do ficheiro o armazenamento dos sinais pode ter algumas particularidades. Caso o formato desejado seja *ascii* e o sinal complexo são criados dois ficheiros com o nome definido e com a distinção da parte real e imaginária: *nome_real* e *nome_imag*. Em ambos os formatos, se o sinal a armazenar for de dupla polarização são também criados dois ficheiros sendo distinguidas as polarizações: *nome_X* e *nome_Y*. O ícone do componente no simulador é apresentado na Figura B.61.



Figura B.61: Ícone do componente SavetoFile no OSIP.

UConstellation

Este componente permite ver a constelação dos sinais nos domínios ótico e elétrico. Os seus parâmetros de configuração são apresentados na Figura B.62.

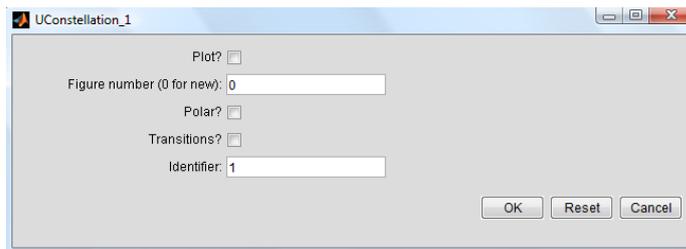


Figura B.62: Parâmetros de configuração do componente UConstellation.

Pode-se verificar na figura Figura B.62 que o componente permite ativar e desativar a funcionalidade de mostrar a constelação. Esta funcionalidade é muito importante pois permite ao utilizador fazer simulações muito longas sem que seja necessário remover componentes. Deve notar-se que para mostrar as figuras das constelações é ocupada uma parte considerável da memória computacional, podendo colocar em causa o desempenho do sistema. É também possível selecionar o número da figura a visualizar, sendo que 0 está reservado para a criação de uma figura nova, e apresentar as transições entre símbolos da constelação, ou mesmo mostrar a constelação do sinal em coordenadas polares. É também necessário definir um identificador. No caso de este representar um valor válido são visualizados os pontos ideais da constelação juntamente com o sinal recebido. Por defeito o componente apresenta duas entradas e uma saída, Figura B.63, isto é, está preparado para os sinais no domínio elétrico. A saída do componente corresponde ao sinal na sua representação complexa ($I+jQ$). Para funcionar no domínio ótico este deve ser configurado para ter uma única porta, a qual deve ser definida como ótica. Neste caso ele interpreta o sinal de entrada como sendo complexo ($I+jQ$).

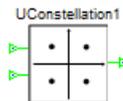


Figura B.63: Ícone do componente UConstellation no OSIP.

Transmitter

Este componente permite implementar um sistema de transmissão completo sem que seja necessário desenhar manualmente o seu esquema. O modelo do transmissor implementado encontra-se na Figura B.64.

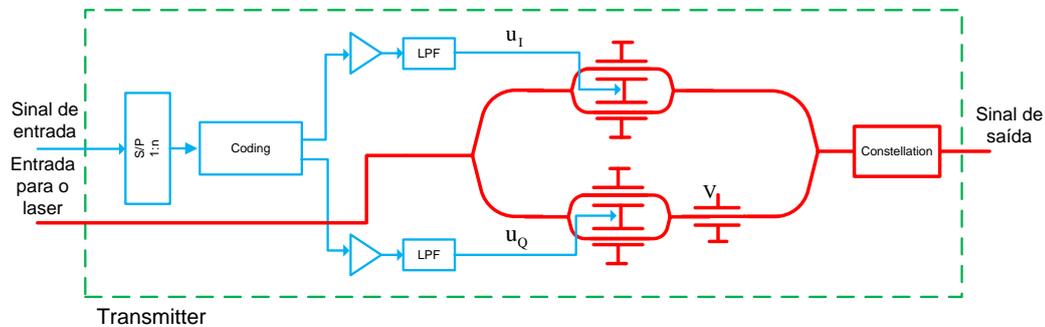


Figura B.64: Modelo do transmissor adotado para o componente Transmitter.

O componente apresenta duas entradas, uma para o sinal binário e outra para o laser local. A sequência binária pode ser modulada com cinco formatos de modulação: BPSK, QPSK, 16QAM, 32QAM ou 64QAM. É também possível ver a constelação do sinal na saída do modulador, assim como a resposta dos filtros elétricos. Este componente implementa assim um sistema de transmissão completo, sendo bastante útil por exemplo para simular vários canais WDM. A redução da complexidade do desenho do sistema traduz-se numa maior eficiência da simulação. Os parâmetros de configuração do componente encontram-se na Figura B.65.

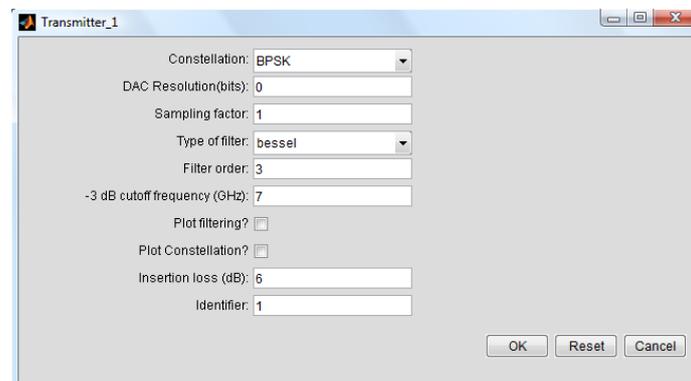


Figura B.65: Parâmetros de configuração do componente Transmitter.

Como se pode ver na Figura B.65 as configurações do componente são idênticas às dos componentes necessários para desenhar o sistema manualmente. O ícone do componente no simulador é apresentado na Figura B.66.

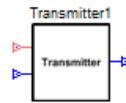


Figura B.66: Ícone do componente Transmitter no OSIP.

Receiver

O objetivo deste componente é implementar um recetor completo sem que seja necessário ocupar uma grande área de desenho. Dessa forma é possível aumentar a eficiência das simulações realizadas. O modelo adotado para a sua implementação é apresentado na Figura B.67.

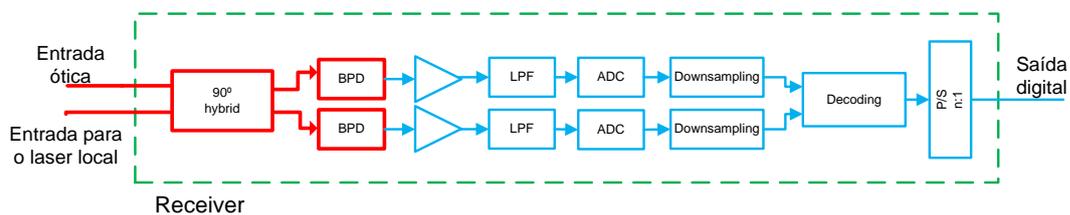


Figura B.67: Modelo do recetor adotado para o componente Receiver.

Este apresenta duas entradas, uma para o sinal modulado no domínio ótico e outra para o laser local. Estas são misturadas resultando a representação do sinal nas suas componentes de fase e de quadratura, que posteriormente são convertidas para o domínio elétrico e amostradas. Os sinais resultantes são depois adquiridos por um circuito de decisão e as sequências binárias descodificadas. Assim todas as funcionalidades de um recetor são implementadas por este componente. Deve notar-se que com este componente não é possível implementar algoritmos de compensação. Para implementar algoritmos é necessário desenhar os esquemas manualmente e introduzir um DSP. Os parâmetros de configuração do componente são apresentados na Figura B.68.

Figura B.68: Parâmetros de configuração do componente Receiver.

As configurações do componente são idênticas às necessárias caso o recetor fosse desenhado manualmente. O ícone do componente no simulador encontra-se na Figura B.69.

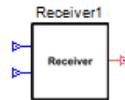


Figura B.69: Ícone do componente Receiver no OSIP.

Apêndice C – Criar uma interface gráfica utilizando o GUIDE

Neste apêndice pretende-se explicar o processo de criação de uma janela de interface com o utilizador em Matlab®. Para essa finalidade utiliza-se a ferramenta GUIDE. Esta permite criar interfaces de forma simples, fornecendo um grande número de recursos, como botões, menus, entre outros. Disponibilizadas estas funções o processo de criação da janela gráfica torna-se bastante simplificado. Para aceder à ferramenta basta escrever na linha de comandos do Matlab® *guide*, sendo apresentada a janela de configuração da Figura C.1.

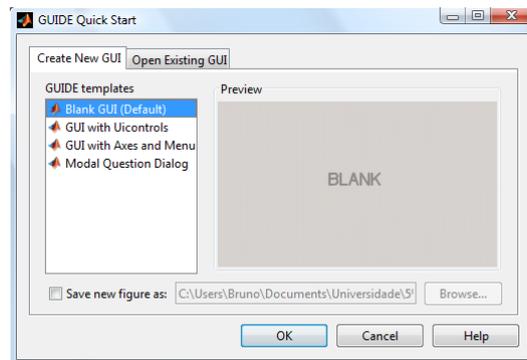


Figura C.1: Ilustração do primeiro passo de configuração da interface gráfica.

Para o caso em estudo, deve-se escolher a opção por defeito, *Blank GUI*, e selecionar *OK*. É agora apresentada a janela da Figura C.2, que permite desenhar a interface.

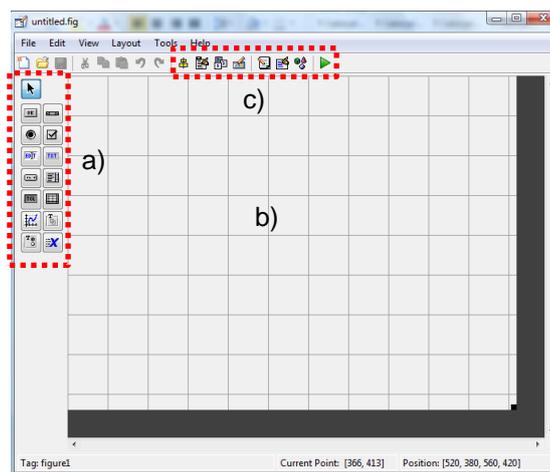


Figura C.2: Ilustração da janela de desenho da interface gráfica com o GUIDE: a) – barra de componentes; b) – área de desenho; c) – barra de ferramentas.

Nesta janela podem distinguir-se três zonas fundamentais: a barra de componentes (Figura C.2-a)), área de desenho (Figura C.2-b)) e a barra de ferramentas (Figura C.2-c)). Com o auxílio destas ferramentas facilmente se constrói a interface desejada. Quando

esta janela é apresentada, são criados internamente dois ficheiros: um referente à componente gráfica e outro para o controlo da componente gráfica (ficheiro de código). O ficheiro de código é inicializado após a gravação da componente gráfica e contém todos os componentes introduzidos, sendo depois necessário configurar as operações de cada um deles.

O primeiro passo é desenhar a janela, para isso deve ter-se em mente o que se pretende realizar e elaborar previamente um esboço. Sendo que o objetivo é apenas exemplificar o processo de criação de uma janela, o exemplo considerado é bastante simples, sendo o seu esboço apresentado na Figura C.3.

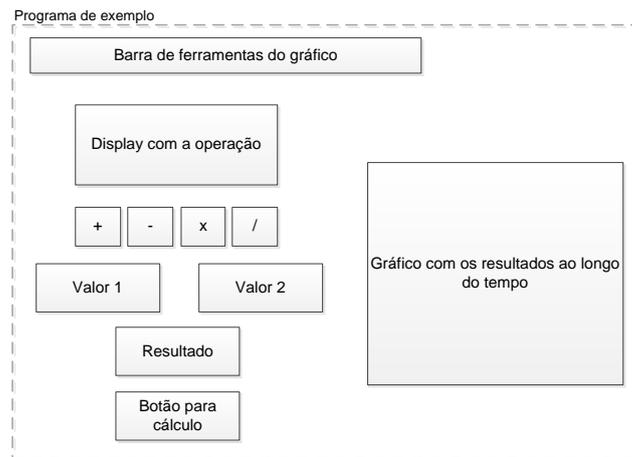


Figura C.3: Esboço da interface a criar.

Pretende-se assim criar uma calculadora que apresente a operação a realizar num display e os resultados no respetivo botão para os resultados, ao mesmo tempo que apresenta todos os resultados obtidos ao longo do tempo num gráfico. Utilizando as opções da barra de componentes, Figura C.2-a), desenha-se o esboço pretendido, sendo este apresentado na Figura C.4.

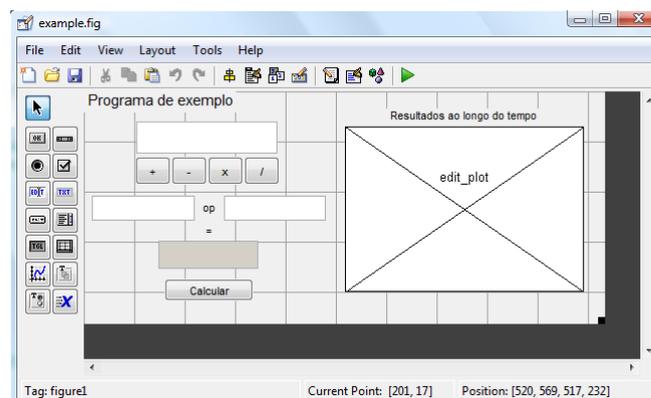


Figura C.4: Esboço da interface pretendida no editor do GUIDE.

Concluído o desenho, clicando duas vezes com o cursor sobre cada objeto introduzido é apresentada a sua janela de configuração, Figura C.5.

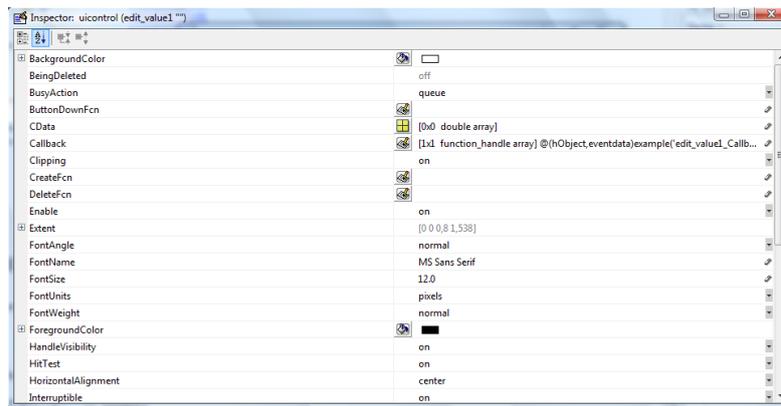


Figura C.5: Exemplo ilustrativo da janela de configuração de cada objeto introduzido.

Aqui deve definir-se as configurações fundamentais do objeto, de que são exemplo o nome, a cor, a textura e o tamanho. Concluído a configuração de cada componente é agora necessário criar a barra de ferramentas do gráfico. Para esse fim utiliza-se a opção *Toolbar editor*, que se encontra na barra de ferramentas (ver Figura C.2-c)). Nesta basta adicionar as opções desejadas, sendo que neste caso apenas se considerou as opções: *zoom in*, *zoom out*, *pan* e *data cursor*. Concluída a configuração gráfica, pode agora proceder-se à gravação da mesma. Assim que este processo termina é gerado automaticamente um ficheiro de código Matlab®, Figura C.6, que se destina à configuração de cada componente gráfico.

```

1 function varargout = example(varargin)
2 % EXAMPLE MATLAB code for example.fig
3 % EXAMPLE, by itself, creates a new EXAMPLE or raises the existing
4 % singleton*.
5 %
6 % H = EXAMPLE returns the handle to a new EXAMPLE or the handle to
7 % the existing singleton*.
8 %
9 % EXAMPLE('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in EXAMPLE.M with the given input arguments.
11 %
12 % EXAMPLE('Property','Value',...) creates a new EXAMPLE or raises the
13 % existing singleton*. Starting from the left, property value pairs are
14 % applied to the GUI before example_OpeningFcn gets called. An
15 % unrecognized property name or invalid value makes property application
16 % stop. All inputs are passed to example_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help example
24
25 % Last Modified by GUIDE v2.5 23-Feb-2012 14:22:55
26
27 % Begin initialization code - DO NOT EDIT
28 gui_Singleton = 1;
29 gui_State = struct('gui_Name',       mfilename, ...
30                  'gui_Singleton',   gui_Singleton, ...
31                  'gui_OpeningFcn', @example_OpeningFcn, ...
32                  'gui_OutputFcn',  @example_OutputFcn, ...
33                  'gui_LayoutFcn',  [], ...
34                  'gui_Callback',    []);
35 if nargin && ischar(varargin{1})

```

Figura C.6: Exemplo ilustrativo do cabeçalho do ficheiro de código gerado para a configuração da interface gráfica.

Para cada componente é criada uma função, que é executada quando se interage com o mesmo. O primeiro passo é ler os valores necessários para os cálculos. Para tal basta ler o valor de cada objeto e posteriormente armazená-lo na estrutura gráfica do

componente até que se pretenda realizar alguma operação, que é desencadeada pelo botão “calcular”. Posto isto, a estrutura gráfica de cada botão deve ser atualizada antes de finalizar as operações. O exemplo do código Matlab[®] para um dos valores é apresentado na Figura C.7.

```
function edit_value1_Callback(hObject, eventdata, handles)
% hObject    handle to edit_value1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_value1 as text
%        str2double(get(hObject,'String')) returns contents of edit_value1 as a
double

value1 = str2num(get(hObject,'String'));
handles.value1 = value1;
guidata(hObject,handles);
return
```

Figura C.7: Exemplo do código de configuração da leitura dos valores de cálculo.

Um algoritmo de controlo semelhante deve ser realizado para as operações de soma, subtração, multiplicação e divisão. Para estes, basta armazenar na estrutura gráfica o valor da operação, sendo considerado que 1 é para a soma, 2 para a subtração, 3 para a multiplicação e 4 para a divisão. Após definido esse valor, o objeto deve ser atualizado e o display atualizado com a operação a realizar. O código Matlab[®] para o caso do operador de soma é apresentado na Figura C.8.

```
function button_Soma_Callback(hObject, eventdata, handles)
% hObject    handle to button_Soma (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.option = 1;
guidata(hObject,handles);
set(handles.display_text,'String','Soma');

return
```

Figura C.8: Exemplo ilustrativo do código de configuração da operação de soma na calculadora.

O botão de cálculo é responsável por atualizar a escala temporal e dar início à operação desejada, os resultados devem depois ser apresentados no botão de resultado e no gráfico. Para facilitar a compreensão do algoritmo dividiu-se esta operação em duas etapas. A primeira é executada quando se pressiona o botão. É então atualizada a variável temporal e atualizado o objeto do botão. De seguida é chamada a função responsável por ler os valores de todos os elementos da janela, que foi designada de *dispatch*. A Figura C.9 apresenta o exemplo do código para as primeiras operações a realizar e a Figura C.10 a função responsável pela leitura, cálculo e atualização dos elementos da interface gráfica.

```

function button_OK_Callback(hObject, eventdata, handles)
% hObject    handle to button_OK (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.val_x = handles.val_x + 1;
guidata(hObject,handles);
dispatch(handles);

return

```

Figura C.9: Exemplo ilustrativo do código de configuração do botão de cálculo.

```

function dispatch(handles)

option = handles.option;
value1 = handles.value1;
value2 = handles.value2;
result = 0;
val_x = handles.val_x;

switch(option)
case 1
    result = value1 + value2;
    set(handles.display_text,'String','Soma');
case 2
    result = value1 - value2;
    set(handles.display_text,'String','Subtração');
case 3
    result = value1 * value2;
    set(handles.display_text,'String','Multiplicação');
case 4
    result = value1 / value2;
    set(handles.display_text,'String','Divisão');
end

set(handles.edit_value1,'String',value1);
set(handles.edit_value2,'String',value2);
set(handles.edit_result,'String',result);

plot(val_x,result,'*r');

return

```

Figura C.10: Exemplo ilustrativo da função responsável pela leitura, cálculo e atualização dos parâmetros da interface gráfica.

Pode verificar-se pela Figura C.10 que o algoritmo de controlo ao funcionamento da calculadora é bastante simples. Nesta abordagem era objetivo construir um algoritmo facilmente compreensível, como tal não se teve em conta a sua otimização. Este deve ser um fator a ter em conta em interfaces mais complexas.

Concluído o processo de construção da componente gráfica e da elaboração do código de controlo dos vários elementos, pode-se agora executar a aplicação. Para isso basta digitar o seu nome na linha de comandos do Matlab®. A Figura C.11 apresenta a janela final, após serem realizados alguns cálculos.

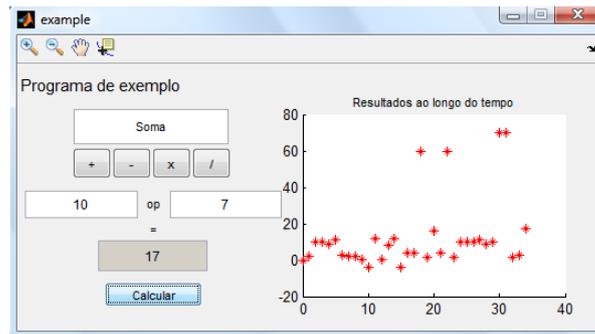


Figura C.11: Janela de interface com o utilizador resultante das configurações definidas.

Na janela final é possível identificar todos os componentes definidos, assim como a barra de ferramentas do gráfico. Pode-se verificar que, em poucos passos e com um grau reduzido de complexidade é possível construir de forma intuitiva janelas de interface com o utilizador.