

Faculdade de Engenharia da Universidade do Porto



FEUP

Igualização Turbo em Sistemas de Comunicações Óticas

Rúben Tiago de Oliveira Pereira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major Telecomunicações

Orientador na FEUP: Prof. Dr. Henrique Salgado
Orientador: Prof. Dr. Miguel Rodrigues
Co-orientador: Mestre Hugo Reboredo

Julho de 2012

© Rúben Tiago de Oliveira Pereira, 2012

Resumo

Ao longo do tempo, e atualmente cada vez mais, existe uma enorme procura de novas técnicas para compensar os efeitos limitativos nas comunicações óticas. Esta procura deve-se essencialmente à necessidade de velocidades de transmissão mais elevadas e ao aumento da largura de banda.

Os códigos turbo foram um grande passo para a evolução das telecomunicações, pois, com o uso de códigos corretores de erros associado a técnicas de descodificação iterativas, foi possível obter um desempenho muito próximo do limite de Shannon. Tal como o nome indica, e pensando um pouco nos automóveis, turbo é baseado numa realimentação, onde existem sucessivas trocas de probabilidades, com o intuito de alcançar o melhor desempenho tendo em conta um número máximo de iterações onde existirá uma saturação.

Neste trabalho vai-se um pouco mais além e propõe-se o estudo da igualização turbo, que tem como princípio a codificação e descodificação turbo. A igualização turbo foi proposta em 1995 e demonstrou ser capaz de suavizar os efeitos da interferência intersimbólica. O igualizador turbo é composto por um igualizador e um descodificador, que tal como na descodificação turbo, trocam informação. O objetivo do igualizador é a compensação da interferência intersimbólica proveniente da fibra.

Abstract

Over the time and nowadays, there has been growing the demand for new techniques in order to compensate for the limiting effects of the optic communications. This demand is due to the need of a higher transmission speed and an increased bandwidth.

The turbo codes have been a huge step for the telecommunication evolution, associated to an iterative decoder technique, they made it possible to achieve a performance similar to the Shannon limit. As the name suggests and thinking a little bit about automobiles, turbo is based on the principle of iteratively exchanging information between the building blocks of the decoder, to achieve a better performance while knowing that exists a maximum number of iterations and a saturation point for it.

In this work, we tried to go a little further and introduced turbo equalization which is based on the turbo coding and decoding principle. The turbo equalization has been suggested in 1995 and it was shown to be capable of smoothing the effect of intersymbolic interference. The turbo equalization is composed by an equalizer and a decoder, which as on the turbo decoding, exchanges information. The equalizer main goal is to compensate the intersymbolic interference introduced by the fiber.

Agradecimentos

Deixo aqui os meus agradecimentos aos meus pais, avós, tio e irmã, que me apoiaram sempre ao longo do curso e agora nesta fase final.

Agradeço ao Hugo que me foi apoiando ao longo do trabalho, e também, ao Prof. Dr. Henrique Salgado e ao Prof. Dr. Miguel Rodrigues por terem proporcionado este trabalho, que de facto me fez adquirir bastantes conhecimentos, e pela disponibilidade prestada. Um outro agradecimento vai para o Prof. Dr. Sílvio Abrantes pela ajuda prestada.

Por fim um muito obrigado à minha namorada, Carina, pelo seu constante apoio, dedicação e amor que tornaram este trabalho possível.

Índice

Resumo	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	ix
Lista de figuras	xi
Lista de tabelas	xiv
Abreviaturas e Símbolos	xv
Capítulo 1	1
Introdução.....	1
1.1. Enquadramento	1
1.2. Contribuições	2
1.3. Organização do documento.....	2
Capítulo 2	3
Comunicações óticas	3
2.1. Sistema de Comunicações Óticas.....	3
2.2. Principais Limitações	3
2.2.1. Fibra: Aspetos principais e atenuação	4
2.2.2. Dispersão na fibra	6
2.2.3. PMD	11
2.2.4. Função transferência para fibra-ótica monomodo	12
2.3. Rádio sobre Fibra.....	12
2.4. Resumo	13
Capítulo 3	15
Modulação	15
3.1. Introdução	15
3.2. BPSK	19
3.3. Resumo	21

Capítulo 4	23
Códigos Corretores de Erros	23
4.1 Introdução	23
4.2 Código de Blocos	24
4.3 Códigos Convolucionais.....	28
4.4 Resumo.....	35
Capítulo 5	37
Algoritmos de Descodificação	37
5.1 Introdução	37
5.2 Viterbi.....	37
5.3 BCJR	42
5.4 Max-Log-MAP	48
5.5 Log-MAP.....	49
5.6 Entrelaçador e desentrelaçador	49
5.7 Resumo.....	50
Capítulo 6	51
Igualização Turbo	51
6.1. Introdução	51
6.2. Codificador externo.....	53
6.3. Codificador interno	54
6.4. Igualizador e decodificador	54
6.5. Resumo.....	56
Capítulo 7	57
Trabalho realizado.....	57
7.1. Introdução	57
7.2. Modulação BPSK	58
7.2.1. BPSK sobre um canal AWGN	58
7.2.2. BPSK sobre fibra	59
7.3. Algoritmos de decodificação	60
7.3.1. Algoritmo de Viterbi	60
7.3.1. Algoritmo BCJR.....	62
7.4. O processamento turbo	64
7.5. A igualização turbo	67
7.6. Resumo e conclusões	70
7.7. Trabalho futuro.....	72
Referências	73

Lista de figuras

Figura 1 - Diagrama de Blocos de um Sistema de Comunicações Óticas	3
Figura 2 - Lei de Snell [3]	4
Figura 3 - Possíveis modos a propagar na fibra	5
Figura 4 - Atenuação na fibra [3]	6
Figura 5 - Espalhamento espectral e interferência intersimbólica [3].....	7
Figura 6 - Dispersão na fibra monomodo [3]	10
Figura 7 - Propagação numa fibra multimodo [3]	10
Figura 8 - Índice de refração numa fibra <i>graded index</i> [3].....	11
Figura 9 - Sistema de rádio-fibra [14]	13
Figura 10 - Representação gráfica de alguns tipos de modulações [15]	16
Figura 11 - Circuitos geradores de modulações digitais [15].....	16
Figura 12 - BER em função da relação sinal-ruído para alguns tipos de modulação [16]	17
Figura 13 - Constelação de um sinal BPSK.....	18
Figura 14 - Constelações de vários sinais modulados [15]	18
Figura 15 - Espectro do Sinal PSK [15].....	19
Figura 16 - Constelação do sinal BPSK	20
Figura 17 - Desmodulador BPSK [18].....	20
Figura 18 - Tipos de Códigos Corretores de Erros	23
Figura 19 - BER para vários SNR e várias taxas de código [1]	24
Figura 20 - Código de blocos a partir da mensagem original [19]	24
Figura 21 - Circuito combinatório para codificação [19]	26
Figura 22 - Grafo de <i>Tanner</i> [20].....	26

Figura 23 - Diagrama de codificação e descodificação com síndrome[19]	27
Figura 24 - Codificador Convolutacional NASA[21]	28
Figura 25 - Diagrama de blocos de um codificador convolutacional[21]	29
Figura 26 - Distância livre[19]	30
Figura 27 - Distância livre para vários tipos de código com taxa 1/2 [21]	30
Figura 28 - Distância livre para vários tipos de código com taxa 1/3 [21]	31
Figura 29 - Codificador convolutacional (2,1,3) de taxa 1/2 [21]	31
Figura 30 - Treliça do codificador convolutacional (2,1,3) [21]	32
Figura 31 - Árvore do código (2,1,3) [21].....	32
Figura 32 - Diagrama de estados do código (2,1,3) [21]	33
Figura 33 - Ganho de codificação em dB para vários códigos com taxa de 1/2 e 1/3 [21]	34
Figura 34 - Codificador RSC [21]	34
Figura 35 - Treliça do codificador sistemático recursivo (2,1,3) [19]	35
Figura 36 - Sistema de codificação e descodificação com canal AWGN [22]	37
Figura 37 - Distância euclidiana [19]	38
Figura 38 - Canal simétrico de entrada binária e saída com 8 níveis [19]	40
Figura 39 - Treliça exemplo [19]	41
Figura 40 - Treliça no estado $l=4$ [19].....	41
Figura 41 - Diagrama de blocos de descodificação iterativa série [23]	43
Figura 42 - Desempenho da descodificação iterativa [23].....	43
Figura 43 - Codificação turbo em paralelo[22]	44
Figura 44 - Diagrama de blocos de descodificador turbo para concatenação em paralelo [22]	45
Figura 45 - Entrelaçamento de uma mensagem[23]	50
Figura 46 - Exemplo de um sistema com interferência intersimbólica [31]	52
Figura 47 - Constituição do bloco de transmissão [25].....	52
Figura 48 - Representação de um canal com ISI [25]	52
Figura 49 - Diagrama de blocos do recetor [25].....	53
Figura 50 - Codificador convolutacional de taxa 1/2 [25]	53
Figura 51 - Treliça do codificador convolutacional [25].....	54

Figura 52 - Igualizador MAP (<i>Maximum a posteriori</i>) [25]	54
Figura 53 - Decodificador MAP [25].....	55
Figura 54 - Igualizador turbo série [25].....	55
Figura 55 - Igualização iterativa [25].....	56
Figura 56 - Sistema com modulação BPSK sobre canal AWGN	58
Figura 57 - BER em função do SNR para um canal AWGN com modulação BPSK.....	59
Figura 58 - Modulação BPSK sobre fibra	59
Figura 59 - BER em função de E_b/N_0 para modulação BPSK em fibra	60
Figura 60 - Decodificação de Viterbi para canal AWGN.....	61
Figura 61 - Decodificação de Viterbi <i>hard-decision</i> após passagem por canal AWGN	61
Figura 62 - Circuito para algoritmo de Viterbi com fibra	61
Figura 63 - Decodificação de Viterbi após fibra	62
Figura 64 - Circuito para algoritmo BCJR com canal AWGN	62
Figura 65 - Circuito para algoritmo BCJR com fibra	62
Figura 66 - Decodificação BCJR após canal AWGN.....	63
Figura 67 - Probabilidade de erro em função de E_b/N_0 para um decodificador BCJR	64
Figura 68 - Sistema para a decodificação turbo iterativa num sistema de comunicações ópticas	65
Figura 69 - BER para 1 iteração turbo num canal AWGN	65
Figura 70 - BER para 2 iterações turbo num canal AWGN	66
Figura 71 - BER para 1 e 2 iterações turbo em fibra.....	66
Figura 72 - Circuito para igualização turbo	67
Figura 73 - BER em função de E_b/N_0 para 1 iteração de igualização turbo com vários tamanhos de entrelaçador	68
Figura 74 - Igualização turbo com 1 e 2 iterações para 1Mbits transmitidos com matriz Geradora $G=[1\ 0\ 0;0\ 0\ 0]$	69
Figura 75 - Igualização turbo com 1 e 3 iterações para 1Mbits transmitidos com matriz Geradora $G=[1\ 0\ 0\ 0;0\ 0\ 0\ 0]$	69
Figura 76 - BER teórico para Viterbi <i>hard-decision</i> em canal AWGN	70
Figura 77 - Igualização turbo para 1, 2 e 3 iterações[28]	71

Lista de tabelas

Tabela 1 - Dispersão da velocidade de grupo.....	8
--	---

Abreviaturas e Símbolos

Lista de abreviaturas (ordenadas por ordem alfabética)

AWGN	<i>Additive White Gaussian Noise</i>
BCJR	<i>Bahl, Cocke, Jelinek, Raviv</i>
BER	<i>Bit Error Ratio</i>
BPSK	<i>Binary Phase-Shift Keying</i>
DEEC	Departamento de Engenharia Electrotécnica e de Computadores
DWDM	<i>Dense Wavelength Division Multiplexing</i>
FEC	<i>Forward Error Correction</i>
FEUP	Faculdade de Engenharia da Universidade do Porto
ISI	<i>Intersymbol Interference</i>
MAP	<i>Maximum a posteriori</i>
ML	<i>Maximum Likelihood</i>
MZM	<i>Mach Zhender Modulator</i>
NRZ	<i>Non Return to Zero</i>
PMD	<i>Polarization Mode Dispersion</i>
PSK	<i>Phase-Shift Keying</i>
QAM	<i>Quadrature Amplitude Modulation</i>
RSC	<i>Recursive Systematic Convolutional</i>
SISO	<i>Soft-input Soft-output</i>
SNR	<i>Signal to Noise Ratio</i>

Lista de símbolos

a	Raio do núcleo da fibra
b	Constante de propagação normalizada
c	Velocidade da luz
D	Parâmetro dispersão
d_f	Distância livre de um código
E_b	Energia de bit
k	Número de bits à entrada do codificador
L	Comprimento da fibra
L_c	Fiabilidade do canal
M	Número de modos a propagar na fibra
NA	Abertura numérica
N	Comprimento de restrição
n	Número de bits à saída do codificador
n_1, n_2	Índice de refração na fibra
P_b	Probabilidade de bit errado
P_e	Probabilidade de símbolo errado
R	Taxa de código
V	Frequência normalizada
v	Memória do codificador
v_g	Velocidade de grupo
x_k	Sequência à saída do codificador
y_k	Sequência à entrada do decodificador
α	Atenuação na fibra
β	Constante de propagação
Δ	Diferença normalizada de índice de refração

ε	Permitividade elétrica
ϕ_c	Ângulo crítico
λ	Comprimento de onda
μ	Permeabilidade elétrica
ω	Frequência angular

Capítulo 1

Introdução

1.1. Enquadramento

As comunicações por fibra ótica têm vindo a ser uma realidade cada vez mais presente, e nesse contexto, tem existido uma evolução da tecnologia no sentido de atingir velocidades de transmissão de dados cada vez mais elevadas. Atualmente os operadores de rede consideram 100Gb/s em sistemas DWDM (*Dense wavelength division multiplexing*). Para estas taxas de transmissão o desempenho dos sistemas de comunicações de fibra ótica sofrem uma significativa degradação do sinal, devido a, não linearidades da fibra, à dispersão de modo de polarização (PMD) e à dispersão cromática [1]. Desta forma é necessário enveredar por outros caminhos para contrariar este problema e avançar com velocidades mais elevadas nas transmissões com canais de fibra ótica sem que o sinal se degrade, e possa chegar a um recetor de forma legível e com o mínimo de erros possível. Em relação ao DWDM, esta é uma tecnologia ótica que permite transmitir ao longo da fibra vários comprimentos de onda, cujos podem ser adicionados e removidos pelos denominados de componentes óticos passivos [2].

Num sistema de comunicações óticas, a maior parte das limitações situa-se no canal de transmissão (fibra ótica), daí ser necessário dar especial atenção a este problema. As principais limitações referidas devem-se a imperfeições na fibra, a dispersão nesta, interferência intersimbólica (ISI), suas não linearidades e por vezes problemas de instalação (ângulos de curvatura demasiado elevados) [3-5]. Para combater estes problemas é necessário então estudar e desenvolver técnicas que os anulem. Estas técnicas incluem, entre outras, o uso de códigos corretores de erros ou a colocação de repetidores e amplificadores de sinal ao longo do canal.

O trabalho a desenvolver centra-se no uso de técnicas de correção de erros em sistemas de comunicações óticos. Para tal, é proposto uma abordagem baseada em codificação turbo, uma técnica comumente utilizada em aplicações práticas (transmissão de imagem, comunicações sem fios, aplicações militares) com resultados comprovados [6, 7]. Esta codificação turbo será realizada por um transmissor com um código convolucional e posteriormente descodificada num recetor através de descodificador dimensionado apenas para tal. Neste recetor também existirá outro descodificador capaz de tratar da igualização do canal. Com a comunicação destes dois descodificadores e a realização da igualização e

2 Introdução

descodificação iterativamente, é possível obter uma probabilidade de erro bastante reduzida na receção dos dados.

1.2. Contribuições

O objetivo final é desenvolver a igualização turbo num sistema de fibra ótica, por forma a compensar a dispersão e não linearidades deste, assim obtendo uma probabilidade de erro reduzida no recetor e ainda a possibilidade de conseguir novas taxas de transmissão mais elevadas até às de então implementadas. Todo o trabalho foi implementado em MATLAB. No contexto deste trabalho as principais contribuições foram:

- Implementação dos algoritmos de Viterbi e BCJR de descodificação;
- Implementação do algoritmo de igualização turbo;
- Simulação e avaliação do impacto destes algoritmos em sistemas óticos baseados na tecnologia de rádio-fibra.

O MATLAB é uma ferramenta com imensas potencialidades, como o cálculo de matrizes, processamento de sinais, construção de gráficos, entre outros. Os aspetos referidos à cerca desta ferramenta são essenciais no desenvolvimento deste trabalho [8]. Com este apoio será possível, após desenvolvido o trabalho, enveredar para outros trabalhos.

1.3. Organização do documento

Este documento é dividido em sete capítulos por forma a dar um seguimento claro a todo o trabalho desenvolvido. O Capítulo 2 introduz os conceitos fundamentais para comunicações óticas, onde se dá especial atenção ao que diz respeito à fibra ótica.

No Capítulo 3 são introduzidos aspetos essenciais para a modulação, onde se convergiu para a modulação BPSK, pois, será esta a utilizada no trabalho.

No Capítulo 4 são revistos aspetos relacionados com codificação. Neste capítulo são abordados códigos de blocos e códigos convolucionais, onde os últimos são essenciais para o desenvolvimento do trabalho.

O Capítulo 5 apresenta vários algoritmos de descodificação, entre eles, Viterbi, BCJR e suas variantes. O BCJR e suas variantes como são utilizados no processamento turbo, têm maior ênfase nesta secção.

O Capítulo 6 é dedicado ao objetivo deste trabalho, a Igualização Turbo, onde é dada uma breve curiosidade histórica e uma introdução. Também é possível ver como funciona um sistema simples e seus constituintes.

Por fim, o Capítulo 7 contém os sistemas desenvolvidos e respetivos resultados obtidos. Neste são ainda retiradas as conclusões do trabalho desenvolvido.

Capítulo 2

Comunicações óticas

2.1. Sistema de Comunicações Óticas

Em comunicações óticas, o objetivo é transmitir o máximo de informação com o mínimo de interferências e perdas ao longo do canal de transmissão. Assim, é possível obter uma maior taxa de transmissão, um número de erros reduzido quando o sinal é detetado por um recetor e ainda aumentar a distância de transmissão entre estações [9].

Um sistema de transmissão por fibra ótica pode ser representado num diagrama de blocos. Este diagrama é constituído por uma fonte que gera uma mensagem, um transmissor que converte a mensagem num formato adequado para um canal de transmissão, um canal de transmissão que faz a conexão entre o transmissor e o recetor, um recetor que extrai do sinal recebido a informação transmitida e por fim envia a mensagem para o destinatário. Este sistema de transmissão pode ser visualizado na Figura 1.

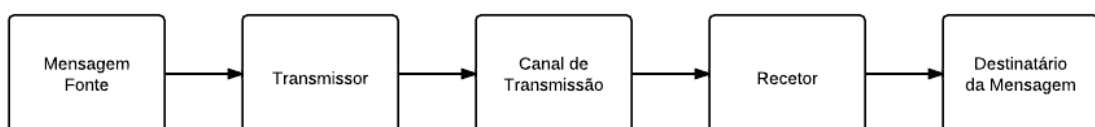


Figura 1 - Diagrama de Blocos de um Sistema de Comunicações Óticas

2.2. Principais Limitações

Num sistema de comunicações óticas os principais problemas estão associados ao canal de transmissão, ou seja à fibra ótica. Estes problemas são normalmente de acoplamento entre o transmissor e a fibra, a atenuação na fibra, a dispersão nesta e as suas não linearidades.

2.2.1. Fibra: Aspectos principais e atenuação

Uma fibra ótica é constituído pelo core (núcleo), cladding (bainha) e o buffer coating (revestimento). Cada fibra tem um índice de refração associado, dado por:

$$n = \sqrt{\mu_r \epsilon_r} \quad (2-1)$$

Como na maior parte dos materiais óticos $\mu_r = 1$, então o índice de refração pode ser reescrito da seguinte forma:

$$n = \sqrt{\epsilon_r} \quad (2-2)$$

Dado este facto, e sabendo que o núcleo da fibra e a bainha têm um índice de refração de n_1 e n_2 respetivamente, então é possível obter a equação de Snell [3]:

$$n_1 \sin\phi_1 = n_2 \sin\phi_2 \quad (2-3)$$

Os ângulos ϕ_1 e ϕ_2 estão representados na Figura 2.

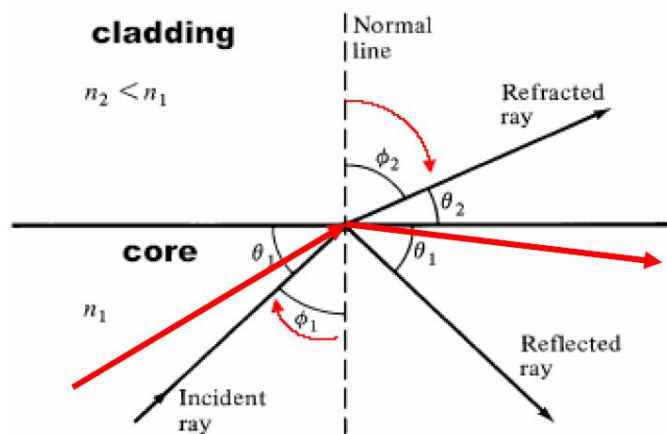


Figura 2 - Lei de Snell [3]

Para que haja condução na fibra, é necessário que haja reflexão total interna e para tal é preciso verificar-se a condição $n_1 > n_2$.

Aumentando o ângulo ϕ_1 até que $\phi_2 = \pi/2$, obtém-se o ângulo crítico ϕ_c , onde deixa de existir condução no interior da fibra. Este ângulo pode ser obtido pela expressão:

$$\sin\phi_c = \frac{n_2}{n_1} \quad (2-4)$$

É também relevante estudar o número de modos que se podem propagar numa fibra. Para tal obtém-se, a partir do raio a da fibra e do comprimento de onda λ , a frequência normalizada V :

$$V = \frac{2\pi a}{\lambda} \times \sqrt{(n_1^2 - n_2^2)} \quad (2-5)$$

Para o caso da fibra monomodo $V \leq 2,405$. Para o caso das fibras multimodo, e apenas quando o número de modos (M) é elevado, consegue-se a seguinte aproximação:

$$M \approx \frac{V^2}{2} \quad (2-6)$$

O gráfico da Figura 3 ilustra melhor como se visualiza o número de modos a transmitir numa fibra a partir da sua frequência normalizada.

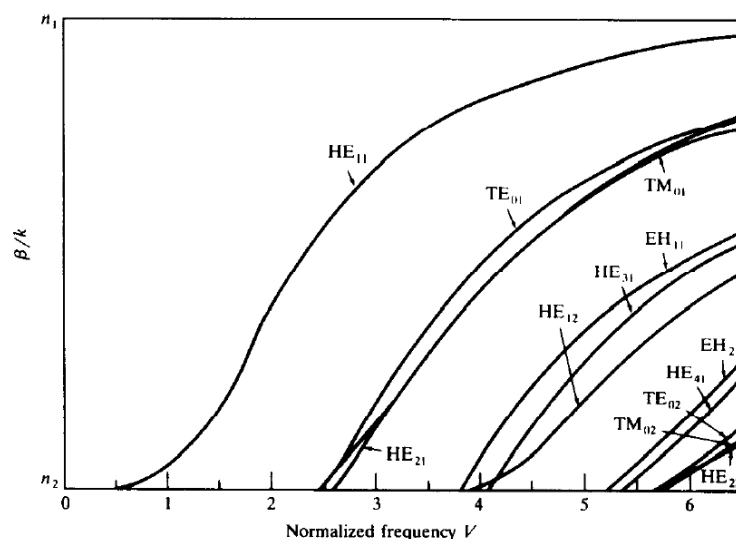


Figura 3 - Possíveis modos a propagar na fibra

Como o próprio nome sugere, na fibra monomodo apenas se propaga um modo, já na multimodo consegue-se propagar um ou mais modos. O facto de apenas se propagar um modo na fibra monomodo, deve-se à reduzida dimensão do diâmetro do seu núcleo. Já a fibra multimodo é distinguida pelo seu núcleo com diâmetro bastante mais elevado que a fibra monomodo. Devido a estas características as fibras monomodo conseguem obter débitos mais elevados e uma maior distância de transmissão.

A atenuação na fibra depende do comprimento de onda associado à mesma. O objetivo foi desde início encontrar o comprimento de onda para o qual a atenuação na fibra fosse mínima. O gráfico que descreve a atenuação em função do comprimento de onda pode ser visto na Figura 4.

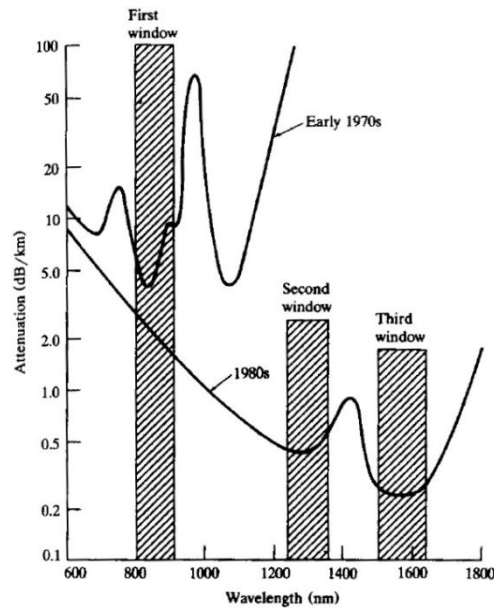


Figura 4 - Atenuação na fibra [3]

Inicialmente encontrou-se um mínimo local de 2dB/Km, situado na 1ª janela, onde o comprimento de onda variava entre os 800nm e os 900nm. Mais tarde chegou-se a um comprimento de onda mais elevado (1300nm), situado na 2ª janela, onde a atenuação era cerca de 0,5dB/Km. Atualmente, o mínimo situa-se em 1550nm correspondentes à 3ª janela, onde a atenuação está muito próxima dos 0,2dB/Km. O valor da atenuação de uma fibra pode ser calculado quando o conhecimento do comprimento da fibra (L), da potência de entrada (P_{IN}) e saída (P_{OUT}), sendo a expressão que traduz este valor [3]:

$$\alpha = -\frac{10}{L} \log_{10} \left(\frac{P_{OUT}}{P_{IN}} \right) \quad (2-7)$$

A atenuação obtida para uma determinada fibra também tem que abranger perdas de acoplamento e perdas de juntas. As perdas de acoplamento devem-se aos conectores utilizados e as perdas nas juntas tem que ver com a qualidade da fundição das juntas ao longo da fibra.

2.2.2. Dispersão na fibra

Para além da atenuação na fibra ainda existe o efeito de dispersão, onde este insere ainda mais distorção de sinal. O efeito dispersão é caracterizado por criar um alargamento temporal do impulso injetado na fibra. Este alargamento é tanto maior quanto maior a largura espectral da fonte ótica (LED ou laser). Ao contrário do laser que tem largura espectral na ordem dos 1nm a 2nm, o LED é conhecido pela sua grande largura espectral (cerca de 5% do comprimento de onda em questão).

A Figura 5 demonstra o espalhamento do impulso numa fibra e demonstra também que quanto maior o alargamento do impulso mais interferência intersimbólica (ISI) existirá. Esta interferência deve-se ao facto de dois impulsos se sobreporem em instantes de tempo adjacentes, causando ambiguidade na deteção de sinal por parte do recetor.

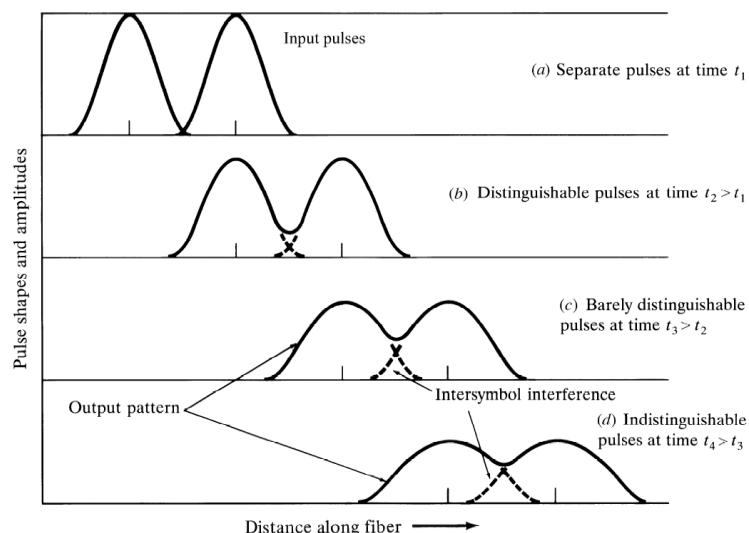


Figura 5 - Espalhamento espectral e interferência intersimbólica [3]

Monomodo

Nas fibras monomodo apenas existe um fator de dispersão, denominada de intramodal ou cromática. Esta dispersão é causada pela diferença de velocidades entre as diferentes componentes espectrais do mesmo modo, e divide-se em dispersão material (D_{mat}) e dispersão do guia de onda (D_{wg}). As diferentes componentes espectrais do mesmo modo resultam do facto de uma determinada fonte ótica, não emitir apenas uma frequência, mas sim várias centradas numa frequência central. Assim existirá um fator decisivo que se denomina de velocidade de grupo, que depende do comprimento de onda, e caracteriza o atraso que cada componente chega ao recetor ótico. A dispersão material surge da variação do índice de refração do núcleo da fibra em função do comprimento de onda. Esta dispersão é dominante em fibras monomodo. A dispersão de guia de onda ocorre porque as fibras monomodo transportam apenas conseguem confinar 80% da potência ótica no núcleo. Desta forma 20% desta potência, escapa-se e é transportada na bainha, de forma mais veloz que no núcleo. Esta diferença de velocidades cria a tal dispersão material.

Antes de ser referido como é possível calcular os dois tipos de dispersão será necessário demonstrar o conceito de velocidade de grupo (v_g) introduzido, onde β é a constante de propagação, ω a frequência da onda e v_p é a velocidade de fase.

$$v_g = \left(\frac{d\beta}{d\omega} \right)^{-1} \quad (2-8)$$

Sendo:

$$\frac{d\beta}{d\omega} = \frac{d}{d\omega} \left(\frac{\omega}{v_p} \right) = \frac{1}{v_p} - \frac{\omega}{v_p^2} \frac{dv_p}{d\omega} \quad (2-9)$$

Assim a velocidade de grupo vem:

$$v_g = \frac{v_p}{1 - \frac{\omega}{v_p} \frac{dv_p}{d\omega}} \quad (2-10)$$

8 Comunicações óticas

A partir da expressão da velocidade de grupo consegue-se saber se existe dispersão, e se sim em que caso se encontra (dispersão normal ou anormal).

Tabela 1 - Dispersão da velocidade de grupo

$\frac{dv_g}{d\omega}$	
< 0	Dispersão normal
0	Sem dispersão
> 0	Dispersão anormal

Como a dispersão da velocidade de grupo só ocorre quando o atraso de grupo (τ_g) depende da frequência ω , assim a expressão do atraso de grupo define-se

$$\tau_g = L\beta_1 + L\beta_2(\omega - \omega_0) + \frac{1}{2}L\beta_3(\omega - \omega_0)^2 \quad (2-11)$$

onde L é o comprimento da fibra.

Assim conclui-se que apenas existe dispersão da velocidade de grupo quando β_2 e β_3 não são nulos. O coeficiente D que caracteriza a dispersão da velocidade de grupo é dado por[3]

$$D = -\frac{2\pi c\beta_2}{\lambda^2} \quad (2-12)$$

onde c é a velocidade da luz, λ é o comprimento de onda e β_2 é dado por

$$\beta_2 = -\frac{1}{v_g^2} \frac{dv_g}{d\omega} \quad (2-13)$$

Na dispersão material o índice de refração ($n(\lambda)$) varia com o comprimento de onda e a constante de propagação β é dada por

$$\beta = \frac{2\pi n(\lambda)}{\lambda} = \frac{\omega \cdot n(\omega)}{c} \quad (2-14)$$

O índice de grupo é definido por

$$n_g = \frac{c}{v_g} = c \times \frac{d\beta(\omega)}{d\omega} = n(\omega) + \omega \frac{dn}{d\omega} \quad (2-15)$$

Então o atraso verificado no material τ_{mat}

$$\tau_{mat} = \frac{L}{v_g} = \frac{L}{c} \left(n - \lambda \frac{dn}{d\lambda} \right) \quad (2-16)$$

Assim, o parâmetro dispersão material é dado por

$$D_{mat} = \frac{1}{L} \frac{d\tau_{mat}}{d\lambda} = -\lambda \frac{d^2n}{d\lambda^2} \quad (2-17)$$

A partir do parâmetro da dispersão material, chega-se à expressão da largura do impulso com espalhamento espectral (σ_{mat})

$$\sigma_{mat} = |D_{mat}(\lambda)| \cdot L \cdot \sigma_\lambda \quad (2-18)$$

onde σ_λ é a largura do impulso da fonte ótica.

Na dispersão de guia de onda a constante de propagação é função de parâmetros da fibra (raio do núcleo e da diferença dos índices de refração no núcleo e na bainha) e ao mesmo tempo é função do comprimento de onda. As equações a seguir demonstradas, permitem chegar ao valor da dispersão em questão.

Sabendo que a constante de propagação normalizada b

$$b \approx \frac{\beta/k - n_2}{n_1 - n_2} \quad (2-19)$$

onde

$$\beta \approx n_2 \cdot k(b\Delta + 1) \quad (2-20)$$

e Δ muito pequeno ($\Delta \ll 1$).

Desta forma é possível obter a expressão para o atraso verificado no guia de onda τ_{wg}

$$\tau_{wg} = \frac{L}{c} \left(n_2 + n_2 \Delta \frac{dVb}{dV} \right) \quad (2-21)$$

onde V é a frequência normalizada, já introduzida. Assim, o parâmetro de dispersão de guia de onda D_{wg} é dado por:

$$D_{wg} = - \frac{n_2 \Delta}{c\lambda} \left[\frac{d^2(Vb)}{dV^2} \right] \quad (2-22)$$

A partir deste parâmetro é possível calcular a largura do impulso alargado (σ_{wg}) devido à dispersão do guia de onda

$$\sigma_{wg} = |D_{wg}(\lambda)| \cdot L \cdot \sigma_\lambda \quad (2-23)$$

Como a dispersão total na fibra monomodo é o somatório da dispersão material e da dispersão de guia de onda, vem que D_{Total}

$$D_{Total} = D_{mat} + D_{wg} \quad (2-24)$$

O gráfico da Figura 6 representa uma fibra monomodo adicionada de dispersão material e dispersão guia de onda. Como já foi referido, e pode-se verificar no gráfico, de facto a dispersão material tem uma maior contribuição para a dispersão total.

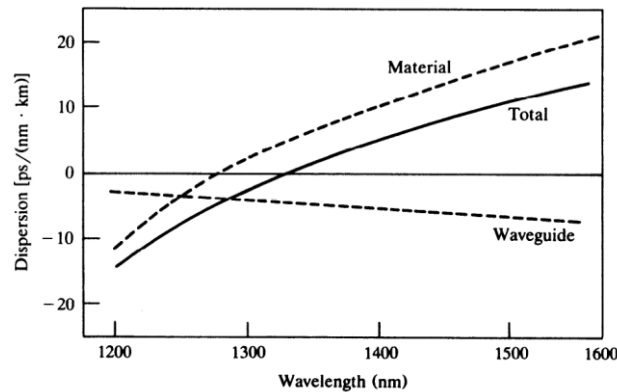


Figura 6 - Dispersão na fibra monomodo [3]

Multimodo

Nas fibras multimodo como existem vários modos em propagação, onde cada um tem o seu trajeto individual, existe dispersão cromática adicionada de dispersão intermodal. A dispersão intermodal afeta significativamente a largura de banda do sinal. Esta dispersão é causada pela diferença de comprimento do trajeto percorrido na fibra pelos diferentes modos, que causa uma diferença temporal na chegada entre modos ao recetor. A Figura 7 mostra uma possível forma de propagação numa fibra ótica.

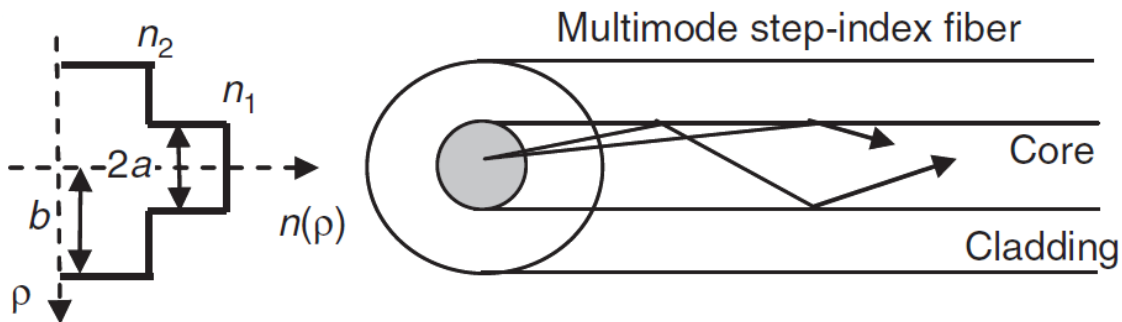


Figura 7 - Propagação numa fibra multimodo [3]

A máxima dispersão do impulso pode ser calculada através da diferença de tempo entre o modo de maior ordem (T_{max}) e o modo fundamental (T_{min})

$$T_{max} - T_{min} = \frac{Ln_1^2}{cn_2} - \frac{Ln_1}{c} \approx \frac{L}{2cn_2} (NA)^2 \quad (2-25)$$

onde NA é a abertura numérica, e que define a capacidade da fibra acoplar luz no seu núcleo. O valor da abertura numérica pode ser obtido por

$$NA = \sqrt{n_1^2 - n_2^2} \approx n_1 \sqrt{2\Delta} \quad (2-26)$$

onde Δ é a diferença normalizada de índice de refração dada por

$$\Delta = \frac{n_1 - n_2}{n_1} \quad (2-27)$$

e $\Delta \ll 1$.

Nas fibras multimodo *step index* apresentadas existe ainda mais uma restrição, pois estas conseguem produtos de bit rate por comprimento (BL) muito baixos. Assim foi necessário recorrer a outro tipo de fibras, denominadas de *graded index*, que conseguem obter melhores produtos BL. Este melhoramento foi possível devido à variação do seu índice de refração no núcleo da fibra ser mais suave (gradual) e não da forma de um impulso quadrado como acontecia nas fibras *step index*. Este facto é visualizado na Figura 8.

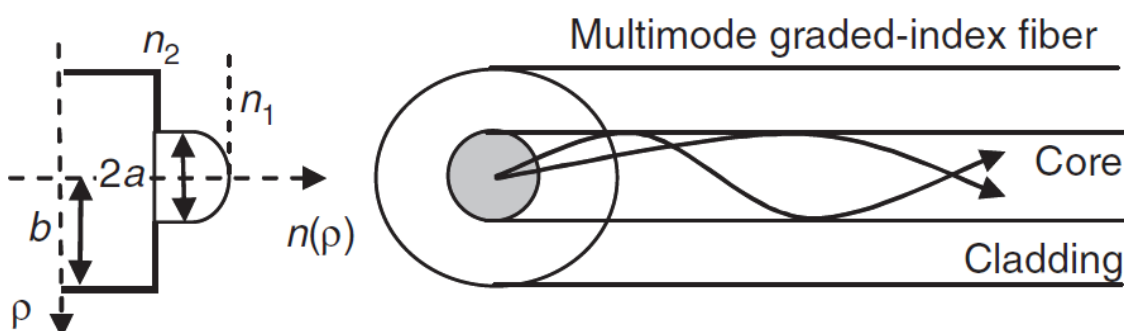


Figura 8 - Índice de refração numa fibra *graded index* [3]

2.2.3. PMD

A dispersão de modo de polarização é de facto muito mais inconstante do que a dispersão cromática, por esta razão as fibras monomodo são bastante afetadas por este tipo de dispersão, e assim existir a necessidade de dar especial atenção a este tipo de dispersão. Esta é causada essencialmente devido à birrefringência geométrica, que se deve à geometria da fibra não ser totalmente simétrica. O grau de birrefringência é dado pela diferença dos índices de refração dos modos polarizados ortogonalmente. A definição de atraso de grupo diferencial (DGD) é importante para se calcular o atraso devido à birrefringência ΔT em fibras de elevado comprimento

$$\Delta T = D_p \sqrt{L} \quad (2-28)$$

onde D_p é o valor médio do coeficiente da dispersão de modo de polarização e L é o comprimento da fibra. Ainda existe um coeficiente de segunda ordem, cujo ocorre quando DGD depende da frequência.

2.2.4. Função transferência para fibra-ótica monomodo

Após a abordagem dos fatores que limitam o desempenho da fibra ótica monomodo, é possível escrever a função transferência devido à dispersão, que caracteriza esta [10, 11]:

$$H_{SMF}(f) = \exp \left[-j \left(\frac{1}{2} \beta_2 (2\pi f)^2 L \right) \right] \quad (2-29)$$

Como se sabe que o parâmetro Dispersão é dado por:

$$D = \frac{-2\pi c}{\beta_2} \quad (2-30)$$

Assim obtém-se que:

$$H_{SMF}(f) = \exp \left[j \left(\frac{D\lambda^2 (2\pi f)^2 L}{4\pi c} \right) \right] \quad (2-31)$$

onde λ é o comprimento de onda, L é o comprimento da fibra e c é a velocidade da luz.

Outro fator a considerar é a atenuação, dada por

$$H_{SMF} = 10^{\left(\frac{-\alpha L}{20} \right)} \quad (2-32)$$

onde α representa a atenuação da fibra em dB/km e L o comprimento da fibra em km. Untando os dois termos das equações (2-31) e (2-32), resulta em

$$H_{SMF}(f) = 10^{\left(\frac{-\alpha L}{20} \right)} \cdot \exp \left[j \left(\frac{D\lambda^2 (2\pi f)^2 L}{4\pi c} \right) \right] \quad (2-33)$$

2.3. Rádio sobre Fibra

Esta tecnologia tem atraído a atenção da comunidade científica como tecnologia que permite explorar as características da fibra para a distribuição de transmissões *wireless*. Este facto deve-se ao seu baixo custo associado por forma a acompanhar o aumento da necessidade do aumento da largura de banda e das exigências do *wireless* [12].

Um sistema de rádio sobre fibra, de uma forma simples, pode ser descrito por um sinal RF (Rádio Frequência) que é modulado sobre uma portadora ótica e transmitido na fibra. Esta portadora ótica é gerada por um Laser ou LED e recebida por um fotodiodo. As bem conhecidas vantagens da fibra ótica, como as baixas perdas, o baixo peso, as características da grande largura de banda, o pequeno tamanho e o baixo custo do cabo, tornam esta a mais flexível e ideal solução para o eficiente transporte de sinais RF [13]. Para além disto, a invulnerabilidade dos cabos de fibra ótica à radiação eletromagnética é outro benefício no que diz respeito a implementações *wireless*.

Nestas redes *wireless* as CS (Central Station) são ligadas às BS (Base Station) através de cabos de fibra ótica. Um sistema de rádio-fibra é apresentado na Figura 9.

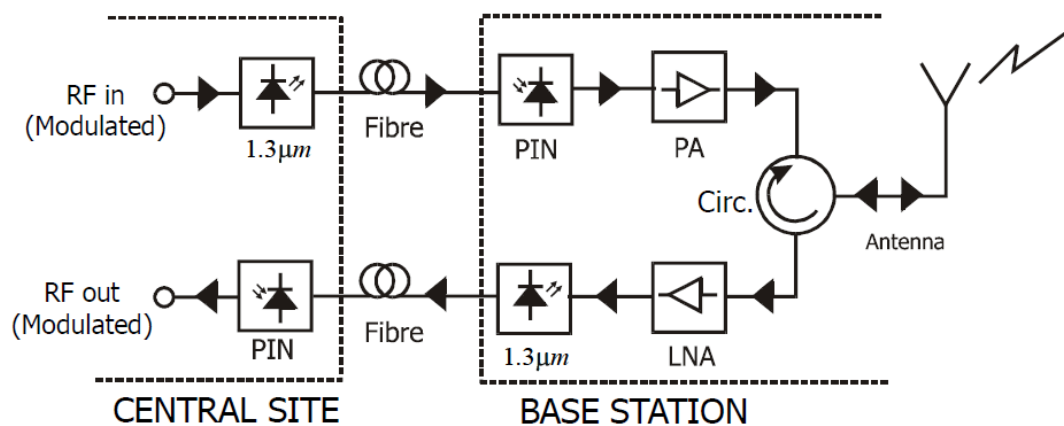


Figura 9 - Sistema de rádio-fibra [14]

2.4. Resumo

Até aqui foram referidas as principais preocupações, ou limitações, a ter em consideração em sistemas de comunicações óticas e as suas principais definições.

Neste capítulo foi dado um maior ênfase às fibras monomodo, pois o trabalho será baseado neste tipo de fibras.

No próximo capítulo serão apresentados vários tipos de modulações digitais, com maior foco na modulação BPSK.

Capítulo 3

Modulação

3.1. Introdução

A modulação é um processo de conversão de sinais para fins de transmissão, sendo definido como um sistema que recebe duas entradas (informação e portadora) e coloca na saída um sinal, que vai ser transmitido ao longo de um canal. A informação é transportada através de uma portadora na sua amplitude, na sua frequência ou na sua fase. Assim conclui-se que esta portadora é analógica e tem que possuir uma frequência maior que a frequência do sinal de informação.

Em modulações digitais os vários tipos de modulação, são os seguintes:

- ASK;
- FSK;
- PSK;
- QAM.

onde ASK (*Amplitude Shift Keying*) representa a modulação em amplitude, FSK (*Frequency Shift Keying*) diz respeito à modulação em frequência, PSK (*Phase Shift Keying*) corresponde à modulação em fase e por fim QAM (*Quadrature Amplitude Modulation*) que é a combinação das referidas anteriormente.

Alguns dos tipos de modulação referidos são apresentados, graficamente, na Figura 10.

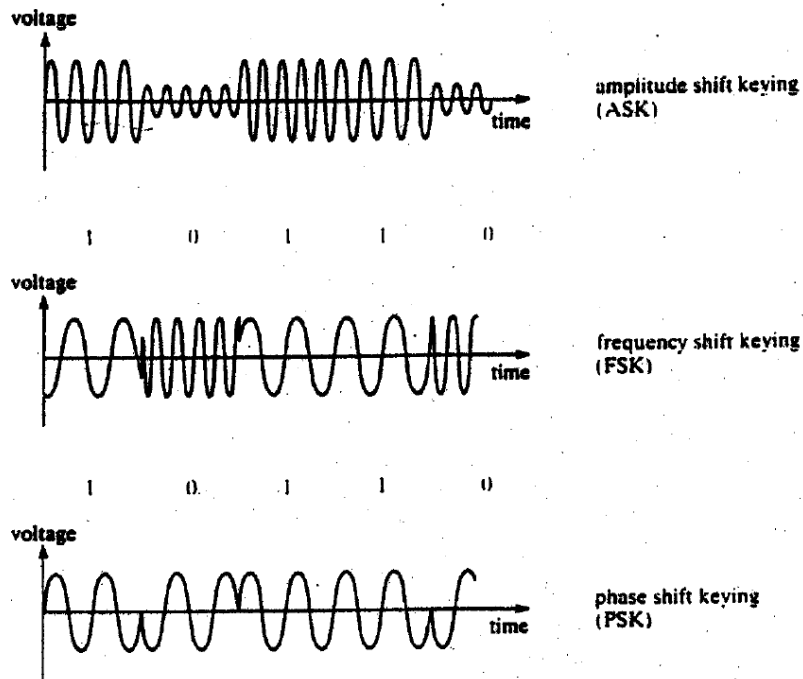


Figura 10 - Representação gráfica de alguns tipos de modulações [15]

Alguns dos circuitos que geram modulações digitais são apresentados na Figura 11.

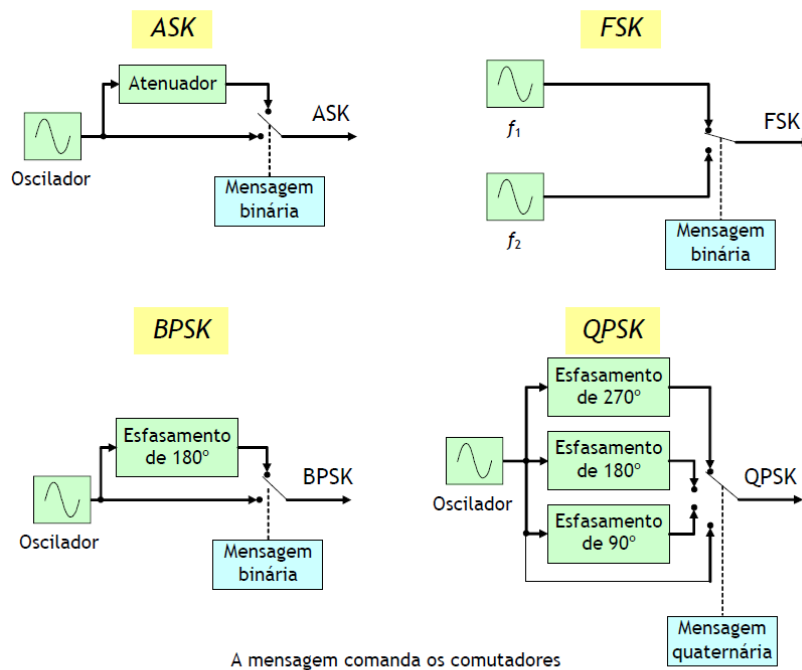


Figura 11 - Circuitos geradores de modulações digitais [15]

Como é possível verificar nos circuitos apresentados, existem modulações não-binárias que geram grupos de k bits (denominados de símbolos) e que dão origem a uma de $M = 2^k$ formas de onda. Desta forma passa-se a ter:

- MPSK;
- MFSK;
- M-QAM.

onde em MPSK existem M fases diferentes, em MFSK existem M frequências distintas e em M-QAM M diferentes combinações de amplitude e fase.

Sabe-se que em MPSK, para a mesma relação sinal-ruído a probabilidade de símbolo e de bit errado (P_e e P_b , respectivamente) degradam-se com o aumento do número M fases. Em MFSK, para a mesma relação E_b/N_0 as mesmas probabilidades melhoram com o aumento do número M de frequências. Posto isto seria desaconselhável utilizar PSK com muitas fases, mas este não é o único critério para o fator qualidade. Os outros fatores são [15]:

- Largura de Banda;
- Potência;
- Eficiência;
- Complexidade.

Com sinalização de fase múltipla (MPSK) a largura de banda não aumenta com M, logo, para a mesma largura de banda, o débito binário aumenta se se aumentar o número de fases do sinal [15].

Neste trabalho será utilizado BPSK (Binary Phase Shift Keying), pois, é uma modulação simples e de baixa complexidade. Para além disto ainda é possível verificar que tem uma taxa de erros (BER) mais baixa que as outras modulações PSK, como se observa na Figura 12.

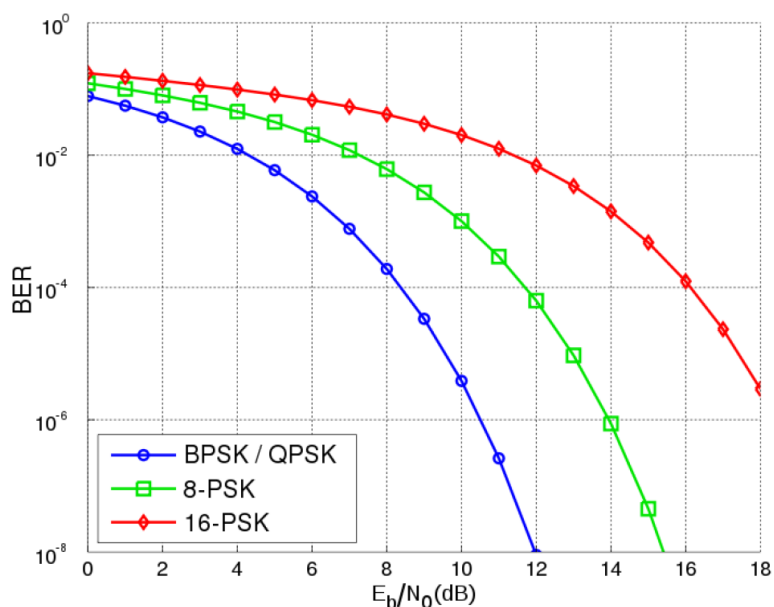


Figura 12 - BER em função da relação sinal-ruído para alguns tipos de modulação [16]

Outra forma possível de verificar que BPSK é muito mais imune a erros do que outras modulações é a partir da constelação de sinal que mostra que os símbolos de BPSK têm maior “folga” entre si, para que não exista interferência intersimbólica, do que nas outras modulações. Este facto pode ser comparado entre a Figura 13 e a Figura 14, onde existe algumas constelações sem ruído e outras perturbadas por este.

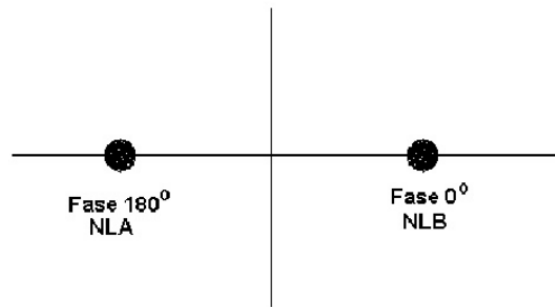


Figura 13 - Constelação de um sinal BPSK

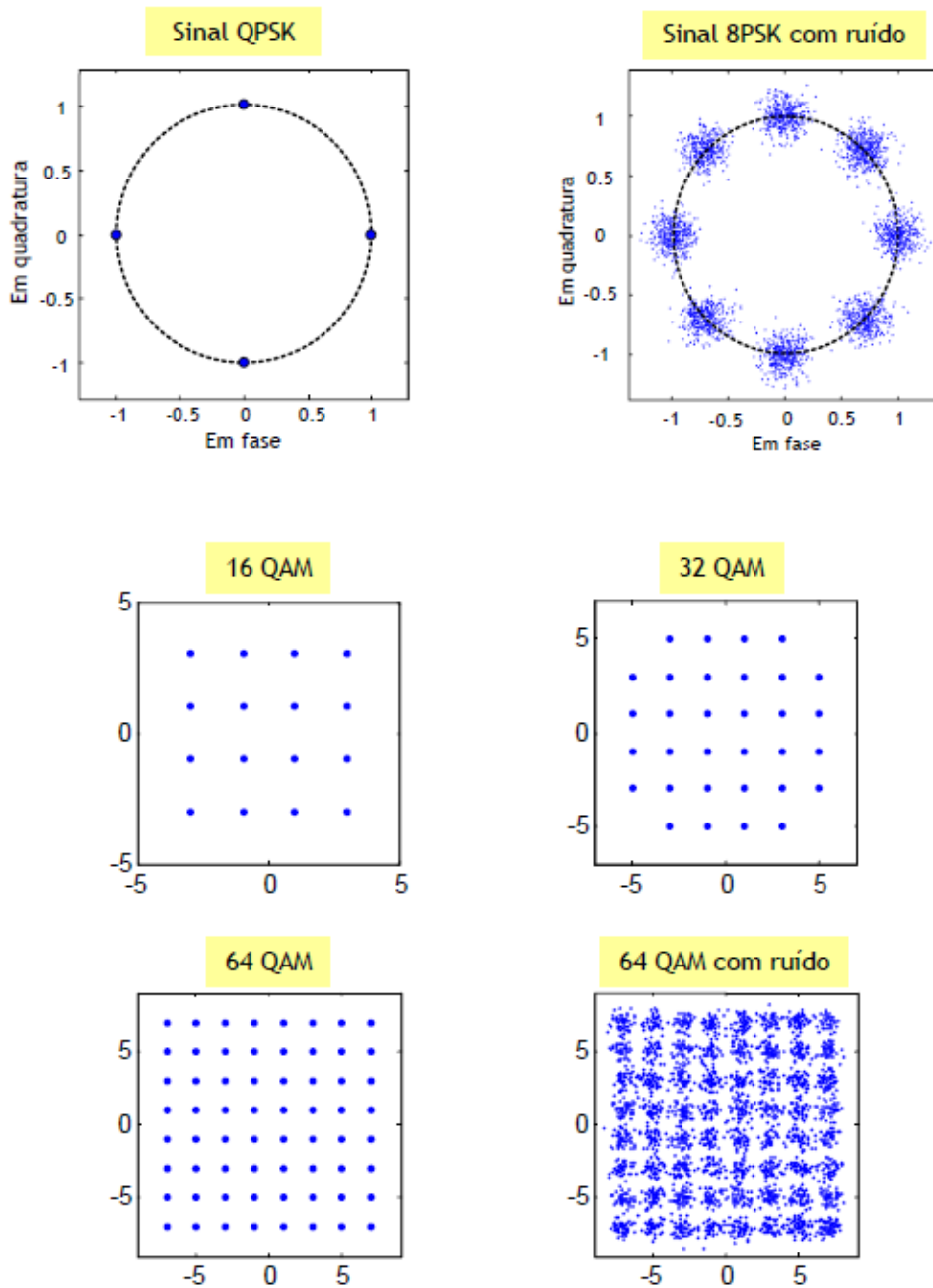


Figura 14 - Constelações de vários sinais modulados [15]

As principais razões para a utilização de modulação em comunicações é a redução de ruído e pelo comportamento em frequência do canal de transmissão que não é constante em todo o espectro [17]. Desta forma é imperativo projetar o modulador e conseqüentemente o desmodulador.

O desmodulador, ou detetor, tem como função extrair a informação presente na portadora, onde este processo, detecção, pode ser realizado de duas formas:

- Detecção coerente;
- Detecção não coerente.

Nos dois tipos de detecção é necessário saber a frequência da portadora recebida e na detecção coerente é ainda necessário conhecer a fase desta portadora.

Desta forma se $s_i(t) = A\cos(2\pi f_c t - \theta_i)$ corresponder à portadora modulada em fase e se $r_i(t) = B\cos(2\pi f_c t - \phi_i(t))$ for o sinal recebido após canal sem ruído, sendo $\phi_i(t)$ uma fase desconhecida, $\phi_i(t) = \theta_i + \theta(t)$, um recetor coerente tem, necessariamente, de conhecer $\theta(t)$. Assim $\theta(t)$ tem de ser estimado por um processo de sincronização e estimação adequado [18].

3.2. BPSK

O sinal PSK (*Phase Shift Keying*) baseia-se na alteração de fase da portadora em função da informação a ser transmitida.

O sinal BPSK é uma variante do PSK. Este utiliza duas fases separadas por 180° (0° e 180°), o que leva a uma forte imunidade a erros, como já foi dito, pois para a sua ocorrência implica a existência de uma forte distorção do sinal. Na modulação BPSK um símbolo corresponde a um bit.

Para um sinal PSK o seu espectro é idêntico ao da Figura 15.

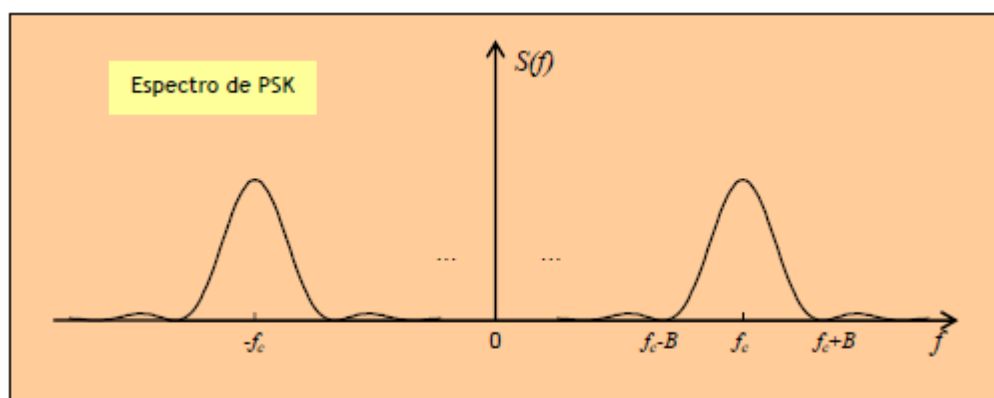


Figura 15 - Espectro do Sinal PSK [15]

Em BPSK existem duas formas de onda antipodais, definidas por[18]:

$$\begin{cases} s_1(t) = \sqrt{\frac{2E_b}{T}} \cos(\omega_c t + 0) \\ s_2(t) = \sqrt{\frac{2E_b}{T}} \cos(\omega_c t + \pi) \end{cases} \quad (3-1)$$

onde t está compreendido entre [0, T] e $s_2(t) = -s_1(t)$.

Como a Figura 17 sugere, no recetor, existe uma função base:

$$\psi_1(t) = \sqrt{\frac{2}{T}} \cos(\omega_c t) \quad (3-2)$$

Desta forma é possível exprimir os sinais PSK:

$$s_i(t) = s_{i1} \psi_1(t) \begin{cases} s_1(t) = s_{11} \psi_1(t) = \sqrt{E_b} \psi_1(t) \\ s_2(t) = s_{21} \psi_1(t) = -\sqrt{E_b} \psi_1(t) \end{cases} \quad (3-3)$$

Obtendo-se então a constelação de um sinal BPSK:

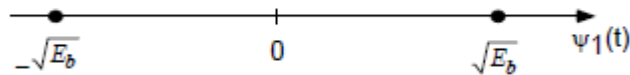


Figura 16 - Constelação do sinal BPSK

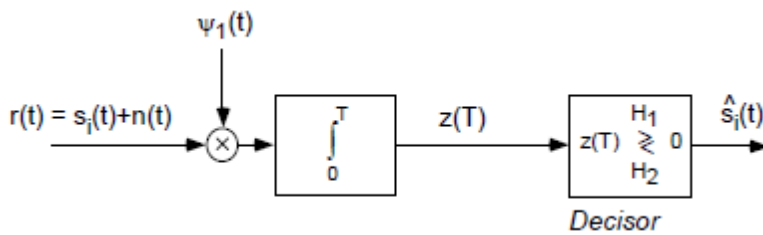


Figura 17 - Desmodulador BPSK [18]

Para saber qual o símbolo obtido na saída do integrador, calcula-se:

- Se s_1 recebido:

$$E[z(T)|s_1] = \sqrt{E_b} \quad (3-4)$$

- Se s_2 recebido:

$$E[z(T)|s_2] = -\sqrt{E_b} \quad (3-5)$$

Após o processo de desmodulação, obviamente, tem-se que calcular a probabilidade de erro obtida [18]:

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (3-6)$$

3.3. Resumo

Neste capítulo foram abordados os principais tipos de modulações para comunicações digitais, as suas vantagens e desvantagens e uma pequena comparação entre estas.

Optou-se por dar mais importância à modulação BPSK, pois pelos motivos já referidos, será a selecionada para prosseguir o trabalho.

No próximo capítulo serão introduzidos dois tipos de códigos corretores de erros (códigos de blocos e códigos convolucionais), com especial atenção sobre os códigos convolucionais.

Capítulo 4

Códigos Corretores de Erros

4.1 Introdução

Esta secção dedica-se aos mais adequados códigos para correção de erros (FEC) num sistema de comunicações óticas. Será dada mais atenção a códigos de blocos e códigos convolucionais, pois é onde se incluem os códigos LDPC e turbo respetivamente. A Figura 18 mostra esta situação descrita. Os códigos turbo serão a base de desenvolvimento deste trabalho, daí a sua abordagem.

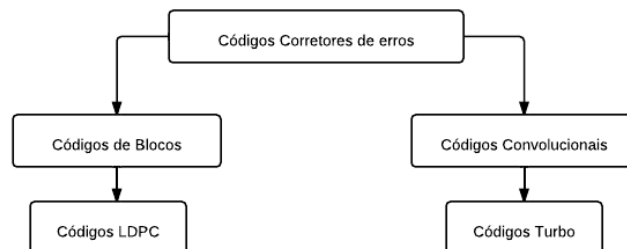


Figura 18 - Tipos de Códigos Corretores de Erros

Quando se converge para a situação em que já não é possível obter um determinado BER a partir de uma pretendido SNR (relação sinal-ruído), então é necessário recorrer à codificação de canal. A quantidade de energia que consegue ser conservada é comumente descrita pelo ganho de codificação [1]. Com estes códigos será possível reduzir a probabilidade de erro e tornar o sistema mais fidedigno.

A codificação tem como princípio de funcionamento a receção de uma sequência de informação de tamanho k numa palavra de código com tamanho n . A principal ideia nos FEC é a introdução de símbolos de redundância na mensagem a ser transmitida, e o uso dos mesmos na descodificação, o que tem como objetivo a correção dos erros introduzidos pelo canal de transmissão.

Num determinado código (n, k) a sua taxa de transmissão R_c é dada por k/n . A taxa do código vai ter influência na taxa de erros, BER, obtida após a transmissão da mensagem, o que pode ser observado na Figura 19.

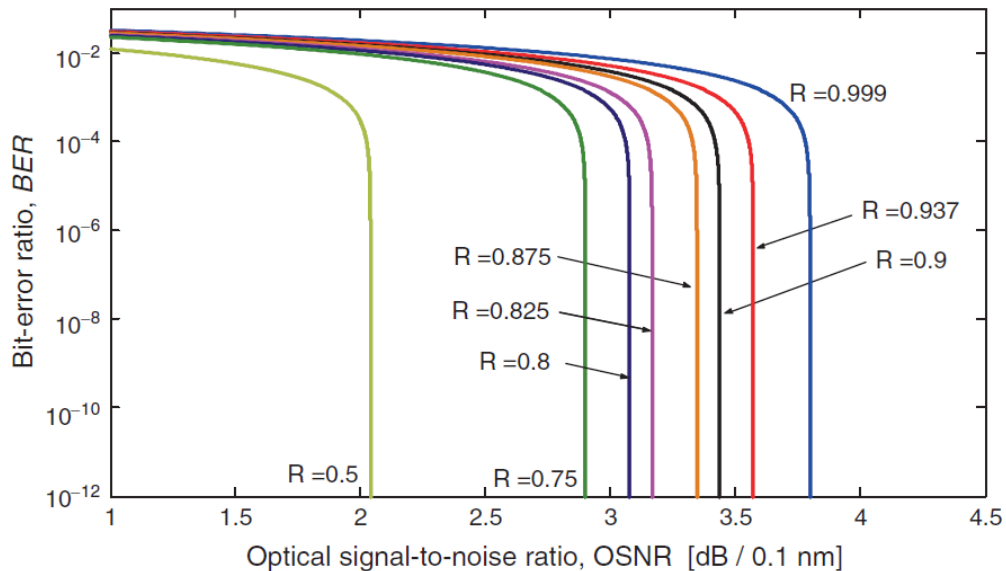


Figura 19 - BER para vários SNR e várias taxas de código [1]

4.2 Código de Blocos

Um código de blocos consiste na divisão de uma determinada mensagem em vários blocos, onde após a separação serão adicionados k símbolos a cada bloco. Estes símbolos denominam-se de símbolos de paridade e ao serem acrescentados, cada bloco passa a conter n símbolos e assim estes blocos com n símbolos passam a denominar-se por palavras de código. O código tem dimensão k e comprimento n . Desta forma utiliza-se, como já foi referido, a nomenclatura (n, k) para representar o código e $R_c = k/n$ para calcular a sua taxa. Como os códigos a utilizar serão de ordem binária, então o código terá apenas 2^k palavras diferentes (verdadeiras). Uma possível representação de códigos de blocos a partir de uma mensagem, pode ser visualizada na Figura 20.

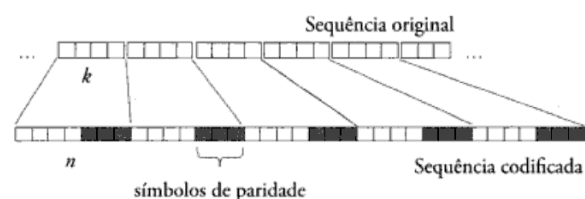


Figura 20 - Código de blocos a partir da mensagem original [19]

Cada mensagem será representada por uma matriz X (constituída por k símbolos) e que com os bits de paridade C fará parte de uma palavra de código representada por Y . Desta forma vem que

$$X = [x_1 \ x_2 \ \dots \ x_k] \text{ e } Y = [x_1 \ x_2 \ \dots \ x_k \ c_1 \ \dots \ c_{n-k}] = [X|C]$$

A este tipo de códigos com esta localização dos bits paridade chama-se de códigos sistemáticos [19].

Cada código é também caracterizado por um peso $w(X)$, que dado pelo número de elementos não nulos do vetor, e pela distância de Hamming $d(X, Y)$, que obtém-se a partir do número de elementos diferentes dos dois vetores. A menor distância entre as 2^k palavras de n bits, chama-se de distância mínima do código (d_{\min}). Esta distância mínima também pode ser obtida através da palavra de código com menor peso não nulo.

Na receção de uma palavra Z (diferente da pretendida Y), como existe erro, então tem que existir um vetor E , denominado de padrão de erro, que coloca a 1 os bits transmitidos com o valor errado, sendo

$$Z = Y + E$$

Pode existir ambiguidade e o recetor pensar que recebe uma palavra de código verdadeira, pois está na sua lista, mas não ter sido esta a ser transmitida, mas sim outra com um erro que a faz ser confundida. Neste caso diz-se que um erro não foi detetado. Através da distância mínima consegue-se saber o número de erros detetáveis (l) e corrigíveis (t) a partir das expressões

$$d_{\min} \geq l + 1 \quad (4-1)$$

$$d_{\min} \geq 2t + 1 \quad (4-2)$$

$$d_{\min} \geq t + l + 1 \quad (4-3)$$

Na transmissão de uma mensagem é necessário utilizar uma matriz constituída por palavras de código (Y), que resulta do produto da mensagem (X) pela matriz geradora (G). Esta matriz geradora tem k linhas e n colunas.

$$Y = XG$$

onde $G = [I_k|P]$ (I_k é a matriz identidade $k \times k$), isto se os bits paridade colocados no fim de cada palavra, e $Y = [X|C]$. As k linhas da matriz geradora são linearmente independentes, o que significa que não se consegue obter nenhuma destas linhas a partir da soma de outras nela contida. Desta forma também é possível concluir que cada linha da matriz G é uma palavra de código, e que os símbolos paridade se podem obter através de $C = XP$. Um exemplo de um circuito combinatório para codificação segue na Figura 21.

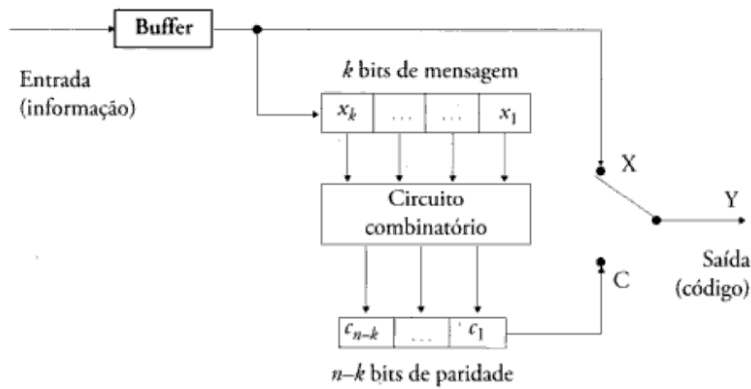


Figura 21 - Circuito combinatório para codificação [19]

A partir da expressão $C = XP$, é possível criar um grafo de Tanner ou bipartido. Este tipo de grafo será útil para perceber como se realiza a descodificação iterativa em códigos LDPC e Turbo. Com o grafo de *Tanner* (Figura 22) é possível obter a síndrome do código. Esta síndrome será explicada mais à frente em forma de matriz.

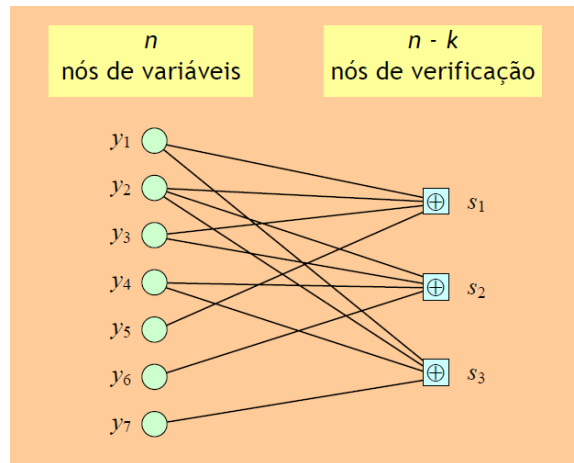


Figura 22 - Grafo de *Tanner* [20]

O grafo tem dois tipos de nós, os nós de variáveis (x_n) e os nós de paridade (h_n) (ou nós de verificação ou controlo). Este grafo serve essencialmente para verificar se determinada palavra pertence ao código ou não. Se a soma dos nós que convergem num determinado nó de paridade for zero, então esta palavra pertence ao código, caso contrário, basta existir um nó de paridade cujo valor não seja nulo, então a palavra não pertence ao código.

Para verificar a paridade também existe uma matriz de verificação de paridade H , e cujas suas linhas também são linearmente independentes. Na descodificação é utilizada a sua matriz transposta e que tem a seguinte estrutura:

$$H^T = \begin{bmatrix} P \\ I_{n-k} \end{bmatrix}$$

Esta matriz é utilizada na descodificação para ser possível detetar se uma dada palavra pertence ou não ao código. Através da multiplicação de $YH^T = XGH^T$, se o produto resultar em zero, então a palavra pertence ao código.

A síndrome já introduzida constitui um vetor $S = ZH^T$, onde $Z = Y + E$ é a palavra recebida. Se o valor obtido para a síndrome for diferente de zero, então cai-se no caso em que Z não pertence ao código. Caso o valor dado pela síndrome seja zero, então ou não existiram erros na transmissão, ou existiram, mas estes conseguiram dar a entender ao decodificador que Z é uma palavra de código.

Na descodificação com síndromes existe uma técnica denominada de “Descodificação de máxima verosimilhança”, onde existe uma estimativa das palavras de erro mais frequentes (\hat{E}) guardadas numa tabela e realizando uma comparação com estas aquando a receção. Como $Z = Y + E$, se for adicionado \hat{E} a Z e se $\hat{E} = E$, então $Z = Y$, o que faz com que a mensagem detetada, seja a que se pretendia receber. Nesta descodificação o número de padrões de erros corrigíveis é de 2^{n-k} . Um possível diagrama de blocos nesta deteção é mostrado na Figura 23.

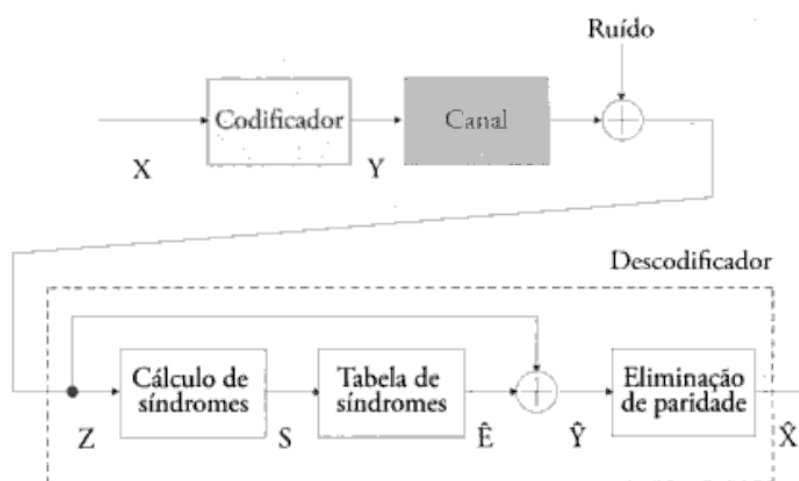


Figura 23 - Diagrama de codificação e descodificação com síndrome[19]

Como um código (n, k) deteta $2^n - 2^k$ padrões de erro e corrige 2^{n-k} padrões de erro, então é possível estimar a probabilidade de erro não detetado (P_{end}) e a probabilidade de erro não corrigido [19].

$$P_{end} = 2^{-(n-k)} B(1 - 2p) - (1 - p)^n \quad (4-4)$$

onde p é a probabilidade de bit errado.

$$P_{enc} = 1 - \sum_{i=0}^n \alpha_i p^i (1 - p)^{n-i} \quad (4-5)$$

Desta forma e sabendo a distância mínima de um código é possível obter a probabilidade de bit errado (P_b)

$$P_b \approx \frac{2t+1}{n} P_{enc} \quad (4-6)$$

4.3 Códigos Convolucionais

Os códigos convolucionais são conhecidos por serem mais simples e eficazes que os códigos de blocos, quando se comparam códigos da mesma complexidade[19]. Nos códigos convolucionais deixa de existir o termo bloco e nestes passa a existir memória, no sentido em que a saída do codificador depende do símbolo de entrada corrente e também de alguns símbolos de entrada passados. Na Figura 24 pode ver-se um possível codificador convolucional.

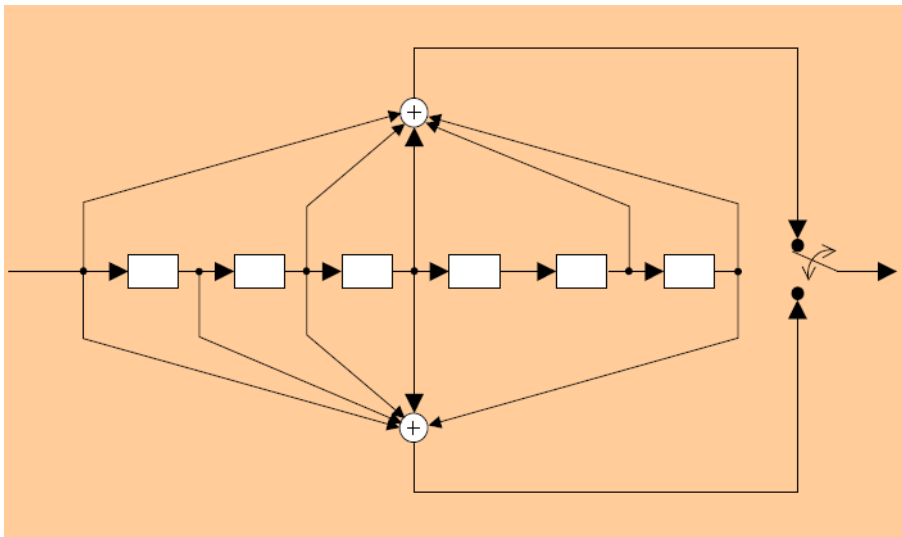


Figura 24 - Codificador Convolucional NASA[21]

Os códigos podem ser representados através de:

- Árvore;
- Treliça;
- Diagrama de estados;
- Vetores geradores;
- Polinómios geradores;
- Resposta impulsional.

Na codificação convolucional, grupos de k bits da mensagem são introduzidos à entrada do codificador e são deslocados de k posições para a direita de cada vez, à medida que novos conjuntos chegam, colocando fora os k bits do bloco (N) mais à direita. A Figura 25 representa este procedimento.

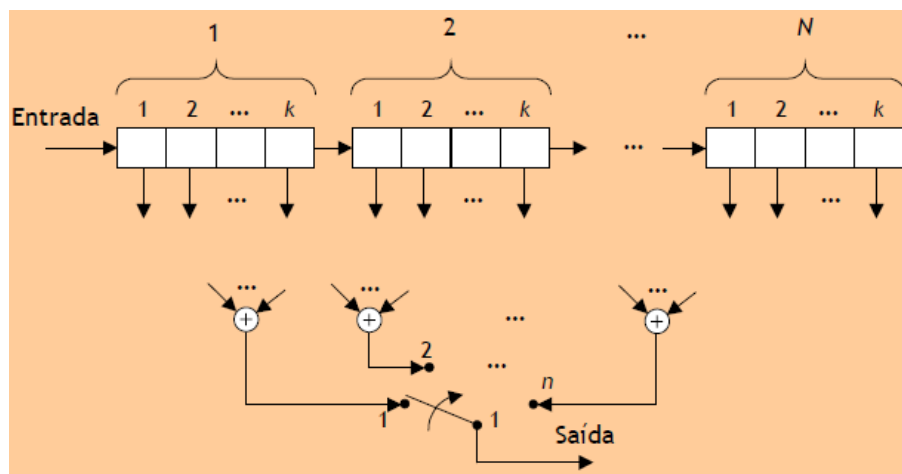


Figura 25 - Diagrama de blocos de um codificador convolucional[21]

Este código tem uma taxa de k/n , devido a na entrada conter k bits e na saída n bits. Estes bits obtidos na saída dependem dos k bits da entrada e ainda dos $v = k(N - 1)$ bits à sua direita. Os v bits referidos são a memória do codificador e denominam-se por bits de estado, podendo ser combinados de 2^v formas, onde cada uma destas formas define o estado do codificador.

Por vezes também se representa um codificador convolucional por (n,k,N) , onde N é o comprimento de restrição e representa o número de deslocamentos de um bit, na entrada do codificador, que influenciam os bits de saída.

Um codificador com taxa de k/n , é caracterizado por um vetor polinomial, denominado de mensagem, à entrada

$$M(D) = [m_1(D) m_2(D) \dots m_k(D)]$$

onde k é o número de subsequências à entrada, e na saída é representado pelo vetor

$$Y(D) = [y_1(D) y_2(D) \dots y_n(D)]$$

Para a obtenção da saída $Y(D)$, basta

$$Y(D) = M(D)G(D)$$

onde $G(D)$ é a matriz geradora. Esta matriz pode estar na forma polinomial, octal ou binária.

Os codificadores convolucionais são caracterizados pela sua distância livre. O valor da distância livre é determinado através do percurso finito, que sai do estado inicial zero e a este com o menor peso possível. Para demonstrar melhor esta situação vê-se a distância livre $d_f = 5$ calculada na Figura 26.

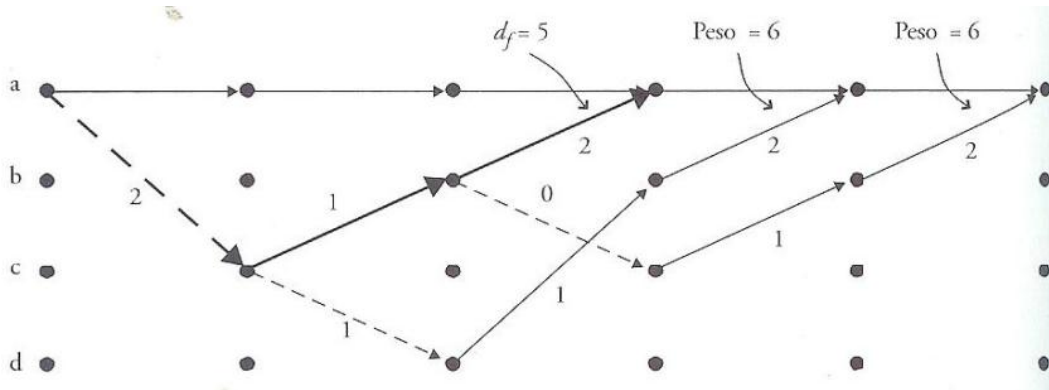


Figura 26 - Distância livre[19]

Em códigos convolucionais com maiores distâncias mínimas

$$d_f \leq \min \left[\frac{2^{l-1}}{2^{l-1}} (l + N - 1)n \right] \quad (4-7)$$

Para ser possível a escolha de um bom código recorre-se à distância livre mínima. Um exemplo de bons códigos são demonstrados na Figura 27, para uma taxa de código 1/2, e na Figura 28 para um código com taxa de 1/3.

Comprimento de restrição, N	Polinómios geradores		d_f	$d_{f_{max}}$
2	1	3	3	4
3	5	7	5	5
4	15	17	6	6
5	23	35	7	8
6	53	75	8	9
7	133	171	10	10
8	247	371	10	11
9	561	753	12	12
10	1131	1537	12	13

Figura 27 - Distância livre para vários tipos de código com taxa 1/2 [21]

Comprimento de restrição, N	Polinómios geradores			d_f	$d_{f_{\max}}$
2	1	3	3	5	6
3	5	7	7	8	8
4	13	15	17	10	10
5	25	33	37	12	12
6	47	53	75	13	13
7	117	127	155	15	15
8	225	331	367	16	17
9	575	623	727	18	18
10	1167	1375	1545	20	20

Figura 28 - Distância livre para vários tipos de código com taxa 1/3 [21]

O exemplo de um bom código com taxa de 1/2 é o código com comprimento de restrição de 3 e matriz geradora $G=[7,5]$, o qual se representa com $(2,1,3)$. A matriz geradora apresentada é constituída pelos dois polinómios geradores correspondentes e está na forma octal, mas este também pode ser representada em binário ou através da transformada em D , da seguinte forma:

- Binário: $G=[111, 101]$;
- Transformada: $G=[1 + D + D^2, 1 + D^2]$.

A sua representação gráfica é apresentada na Figura 29.

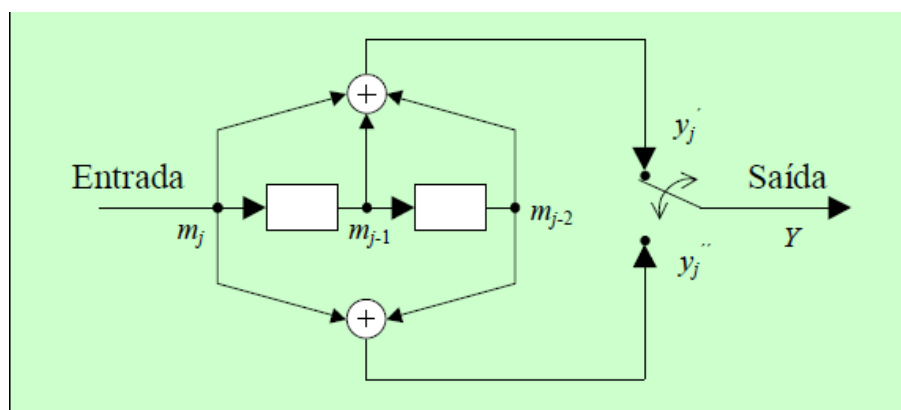


Figura 29 - Codificador convolucional $(2,1,3)$ de taxa 1/2 [21]

A treliça correspondente a este codificador é a apresentada na Figura 30 e a partir desta é possível determinar o comportamento do codificador. Na treliça o traço a cheio corresponde a um bit $u_k = -1$ recebido e o traço descontínuo corresponde a um bit $u_k = +1$ recebido. Existe um estado atual e um estado seguinte que se interligam e dão origem a dois bits na saída, onde o valor lógico 0 e 1 na saída correspondem a -1 e $+1$ respetivamente.

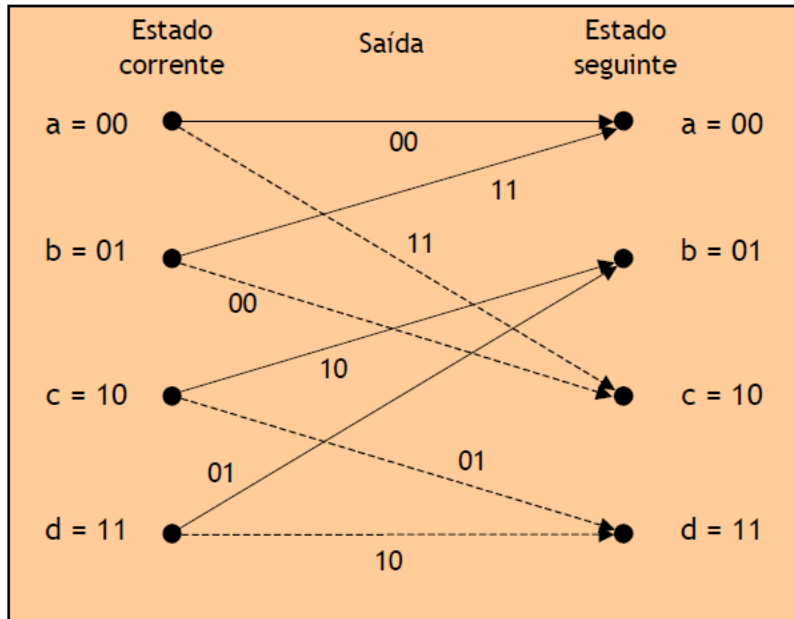


Figura 30 - Treliza do codificador convolucional (2,1,3) [21]

Este mesmo codificador pode também ser representado através de uma árvore ou um diagrama de estados, apresentados na Figura 31 e na Figura 32 respetivamente.

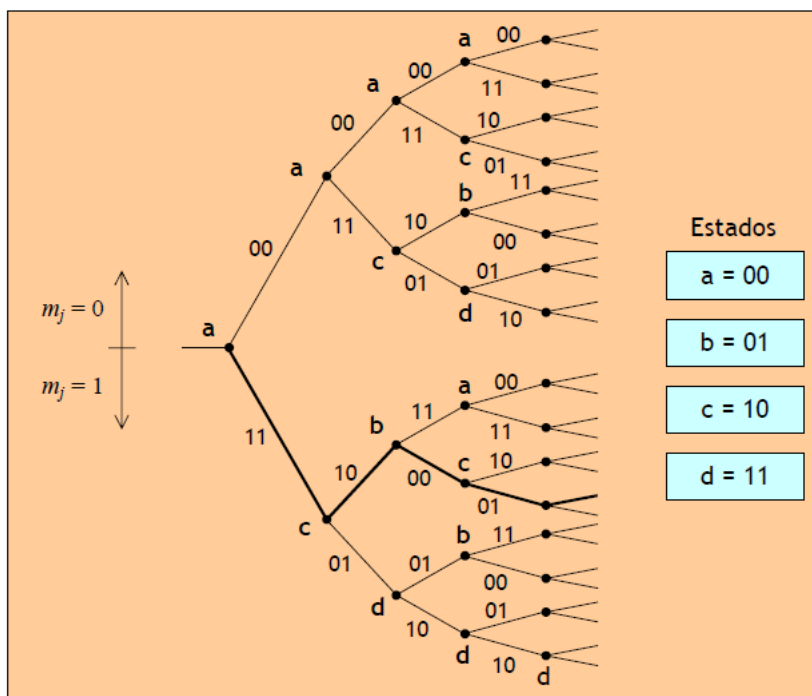


Figura 31 - Árvore do código (2,1,3) [21]

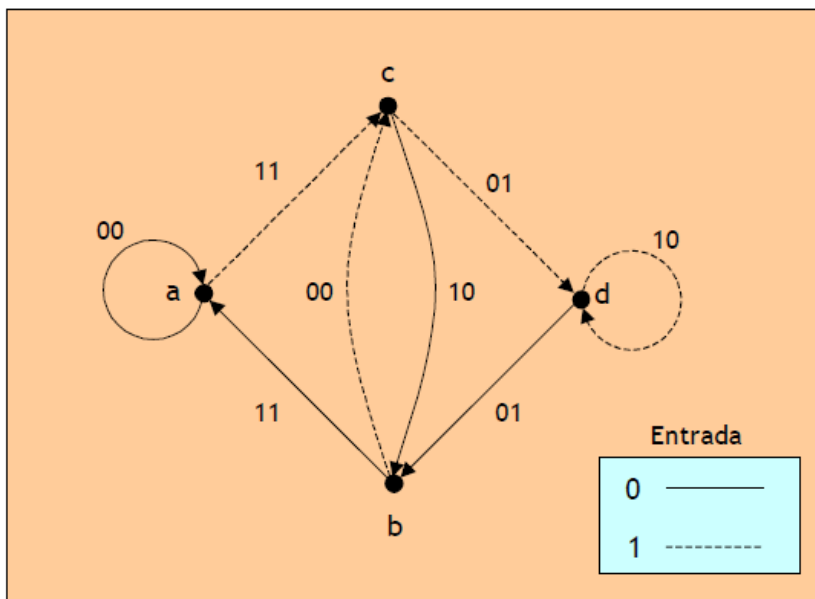


Figura 32 - Diagrama de estados do código (2,1,3) [21]

Nestes codificadores é necessário acrescentar $N - 1$ ou menos bits de cauda à sequência de entrada para que de um estado qualquer se regresse ao estado inicial da treliça.

Um outro fator importante nos códigos convolucionais é o ganho de codificação, pois em qualquer sistema existe a necessidade de estimar a razão sinal-ruído ($SNR = E_b/N_0$). Para se obter um ganho de codificação máximo é necessário estar no caso em que a relação sinal-ruído é bastante elevada. O valor de ganho de codificação é dado perante duas situações:

- Decisões rígidas [19]

$$\text{Ganho de codificação} \leq 10 \log \frac{d_f R_c}{2} \quad (4-8)$$

- Decisões brandas não quantizadas

$$\text{Ganho de codificação} \leq 10 \log d_f R_c \quad (4-9)$$

Com as expressões para cada tipo de decisão consegue-se concluir que as decisões brandas conseguem obter um desempenho de 3dB superior às decisões rígidas. A Figura 33 apresenta os códigos com melhor ganho de codificação para decisões rígidas.

Comprimento de restrição, N	<i>Taxa 1/2</i>		<i>Taxa 1/3</i>	
	Distância livre	Ganho de codificação máximo (dB)	Distância livre	Ganho de codificação máximo (dB)
3	5	0,97	8	1,25
4	6	1,76	10	2,22
5	7	2,43	12	3,01
6	8	3,01	13	3,36
7	10	3,98	15	3,98
8	10	3,98	16	4,26
9	12	4,77	18	4,77
10	12	4,77	20	5,23

Figura 33 - Ganho de codificação em dB para vários códigos com taxa de 1/2 e 1/3 [21]

Neste trabalho é inevitável introduzir os codificadores sistemáticos recursivos (RSC), pois, são utilizados em turbo-códigos. Estes codificadores podem ser obtidos a partir dos não-sistemáticos não-recursivos, usando a matriz de verificação de paridade. A distância livre mantém-se, conseqüentemente o ganho de codificação mantém-se igual aos codificadores anteriormente abordados. Os codificadores recursivos apresentam ainda melhor relação sinal-ruído comparativamente com os não-recursivos.

A Figura 34 mostra o codificador RSC equivalente ao codificador não-sistemático e não-recursivo (2,1,3) a cima apresentado. A partir deste codificador é possível verificar que existe um bit sistemático y_{k1} , que é obtido diretamente a partir do bit de entrada do codificador, e um outro bit, este denominado de paridade (y_{kp1}), que é gerado pelo codificador.

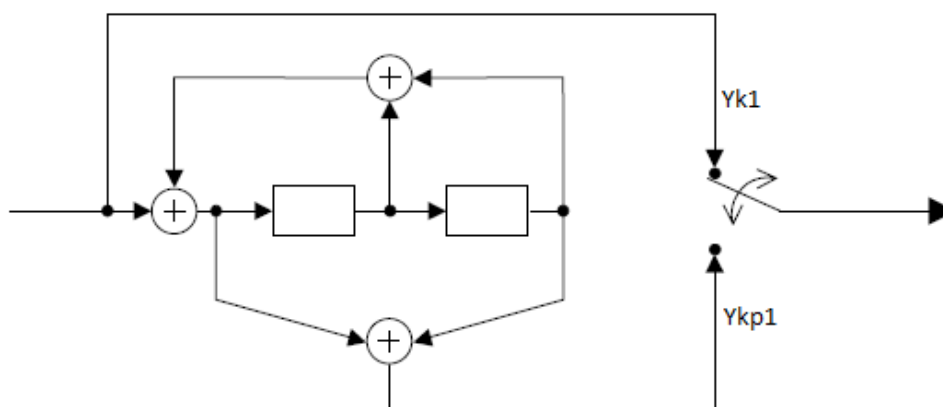


Figura 34 - Codificador RSC [21]

Após a obtenção deste codificador nota-se que a matriz geradora do mesmo será alterada e será da seguinte forma:

$$\bullet \quad G = \left[1 \quad \frac{1+D^2}{1+D+D^2} \right]$$

A partir da matriz geradora pode verificar-se o que já foi descrito, onde a primeira saída do codificador é o bit sistemático e a segunda saída, bit de paridade, é obtida com a divisão dos polinómios geradores que constituem o codificador. A treliça do codificador é apresentada na Figura 35 e como se pode ver sofre alterações em relação a treliça do codificador não-sistemático e não-recursivo equivalente.

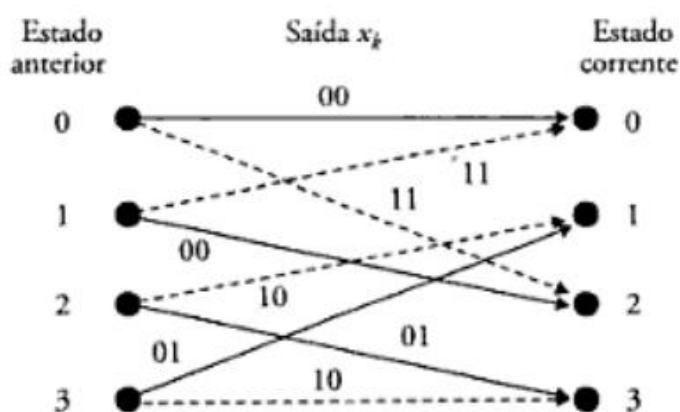


Figura 35 - Treliça do codificador sistemático recursivo (2,1,3) [19]

4.4 Resumo

Este capítulo 4 abordou dois tipos de códigos, os de blocos e os convolucionais. Foi apresentado um gráfico no qual se verifica qual a melhor taxa de código tendo em conta o BER em função da relação sinal-ruído.

Identificou-se que para este trabalho, como se utiliza codificação turbo, o melhor é utilizar códigos convolucionais, onde foi fundamental referir os códigos convolucionais sistemáticos e recursivos.

No próximo capítulo serão apresentados vários tipos de algoritmos de descodificação, onde a atenção recai sobre o algoritmo BCJR e suas variantes.

Capítulo 5

Algoritmos de Descodificação

5.1 Introdução

Após já se ter descrito a modulação, a respetiva desmodulação e a codificação de um canal, agora é necessário abordar a descodificação para completar o sistema de comunicações. Este sistema é baseado na Figura 36, onde se representa o canal de transmissão por um canal de ruído Gaussiano branco (AWGN). Neste capítulo serão descritos dois algoritmos de descodificação (Viterbi e BCJR), onde ainda serão descritas, também, duas simplificações deste último, que se denominam de log-MAP e Max-log-MAP.

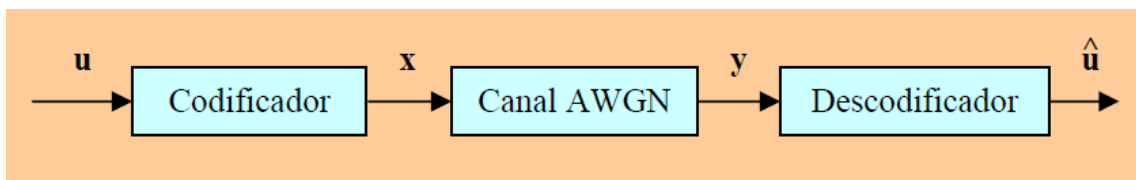


Figura 36 - Sistema de codificação e descodificação com canal AWGN [22]

5.2 Viterbi

Define-se uma sequência binária codificada, Y , composta por nL bits que é modulada e transmitida através de um canal afetado por ruído. Após a passagem pelo canal, o sinal é desmodulado e entrega o sinal resultante ao descodificador. Perante o tipo de sequência de saída, será realizada descodificação rígida se esta for binária, descodificação branda não-quantizada se a saída contiver valores reais ou descodificação branda quantizada em Q níveis [19].

A partir de uma mensagem com kM bits obtém-se um percurso constituído por M ramos, o que origina 2^M percursos possíveis e onde apenas um destes percursos será o verdadeiro [19]. Neste problema seria muito difícil conseguir encontrar o percurso verdadeiro, ou percurso da mensagem, mas como nem todos os percursos têm a mesma probabilidade, então tenta-se seleccionar o percurso mais verosímil. No algoritmo de Viterbi é necessário calcular o peso de cada transição na treliça e posteriormente efetua-se a soma dos pesos de cada trajeto e selecciona-se o mais verosímil.

Para encontrar o percurso de ML (Máxima Verosimilhança) com o algoritmo de Viterbi existem três tipos de descodificação:

- Descodificação branda não-quantizada;
- Descodificação branda quantizada;
- Decodificação rígida.

Cada um destes tipos de descodificação atinge o percurso de máxima verosimilhança de forma distinta, daí a abordagem e explicação dos três.

Em qualquer uma destas descodificações, a maximização da verosimilhança equivale a maximizar, ou minimizar, um certo parâmetro de distância que se designa de métrica acumulada e que é igual à soma das métricas de cada ramo de percurso.

Descodificação branda não-quantizada

A descodificação branda não-quantizada tem como métrica a distância euclidiana quadrática, a qual selecciona o percurso de ML, entre os percursos válidos, aquele que se situa a menor distância euclidiana da sequência recebida [19]. Esta distância euclidiana é a distância entre dois pontos do espaço de sinal, como mostra a Figura 37.

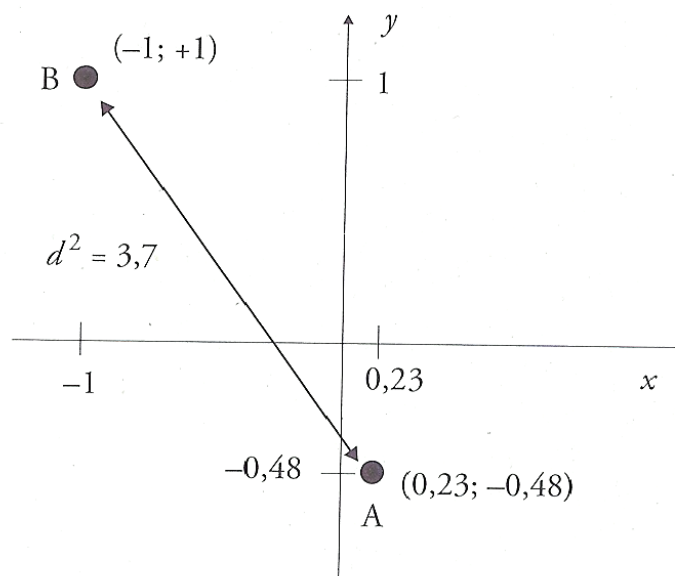


Figura 37 - Distância euclidiana [19]

A distância entre A e B é dada pela expressão

$$d_{AB} = \|\overline{A - B}\| \quad (5-1)$$

Desta forma vem que a distância euclidiana quadrática entre os pontos associados a cada ramo do percurso i é dada por [19]

$$m_l^{(i)} = \sum_{j=1}^n (z_{lj} - y_{lj}^{(i)})^2 \quad (5-2)$$

e a métrica acumulada de um percurso de L ramos é igual à soma das L distâncias quadráticas.

$$\sum_{l=1}^L m_l^{(i)} \quad (5-3)$$

Como pela regra de máxima verosimilhança o objetivo é encontrar a menor das métricas acumuladas, então

$$\min_{(i)} \left[\sum_{l=1}^L m_l^{(i)} \right] \quad (5-4)$$

Outra forma possível de alcançar o percurso de máxima verosimilhança é através da maximização da soma dos produtos internos dos vários pares de pontos y_l e z_l com a seguinte expressão

$$\max_{(i)} \left[\sum_{l=1}^L z_l \cdot y_l^{(i)} \right] \quad (5-5)$$

Conclui-se então que na primeira possibilidade de cálculo do percurso de ML corresponde à menor distância acumulada e no segundo caso corresponde à maior métrica acumulada.

Descodificação branda quantizada

Nesta descodificação a métrica rege-se pela soma de logaritmos

$$m_l^{(i)} = \sum_{j=1}^n \ln P(z_{lj} | y_{lj}^{(i)}) \quad (5-6)$$

Para obter as probabilidades $P(z_{lj} | y_{lj}^{(i)})$ necessárias no cálculo da métrica, é necessário seguir a Figura 38 que contém estas probabilidades que se situam nas transições do canal discreto sem memória.

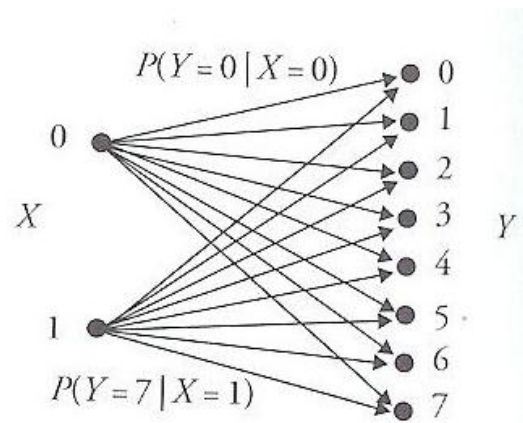


Figura 38 - Canal simétrico de entrada binária e saída com 8 níveis [19]

O percurso ML é o percurso $Y^{(i)}$ para o qual a log-verosimilhança

$$\ln P(Z|Y^{(i)}) = \sum_{l=1}^L \ln P(z_l|y_l^{(i)}) = \sum_{l=1}^L \sum_{j=1}^n \ln P(z_{lj}|y_{lj}^{(i)}) \quad (5-7)$$

é máxima [19].

Descodificação rígida

Com este tipo de descodificação a métrica que se usa é a distância de Hamming entre palavras binárias, que se obtém a partir do somatório de XOR's (OU exclusivo) entre os bits de mensagem recebidos no descodificador e os bits de saída da treliça

$$d_l = \sum_{j=1}^n z_{lj} \oplus y_{lj}^{(i)} \quad (5-8)$$

Após este cálculo chega-se ao percurso de máxima verosimilhança $Y^{(i)}$, verificando qual destes se encontra à menor distância de Hamming da sequência recebida Z, através de

$$\min_{(i)} [\sum_{l=1}^L d_l] \quad (5-9)$$

A partir de uma treliça como a da Figura 39, é possível verificar como se processará a descodificação de Viterbi, quais os estados que convergem para que estados, quais os que divergem a partir do próximo estado e ainda qual a saída da treliça para realizar os cálculos de descodificação.

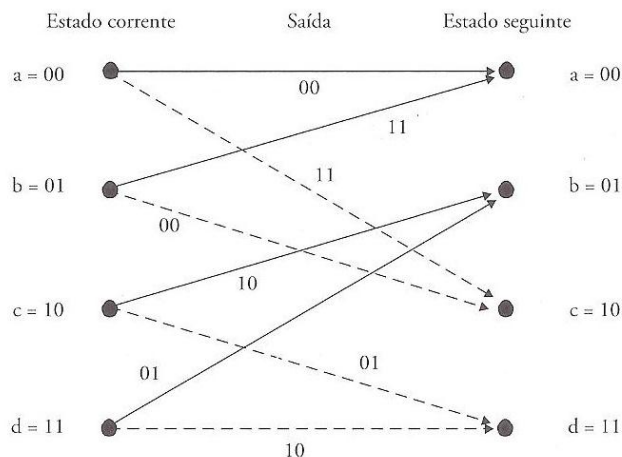


Figura 39 - Treliça exemplo [19]

No algoritmo de Viterbi a partir do nível $l = 3$, existem 2^l percursos, pois em cada estado convergem dois ramos e divergem outros dois, desta forma seleciona-se sempre o percurso mais parecido com a sequência recebida, isto é, o que tiver menor métrica acumulada. Verifica-se, então, que a cada dois ramos que convergem num estado, conserva-se o de menor métrica acumulada e o outro descarta-se, como se pode ver na Figura 40, os percursos descartados estão marcados com uma cruz. Em caso de dois percursos obterem a mesma métrica, então descarta-se um à sorte.

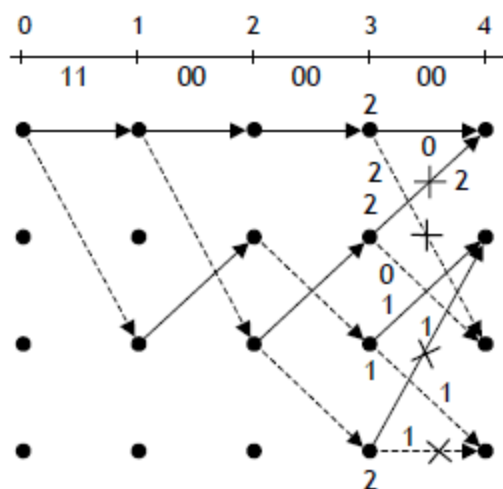


Figura 40 - Treliça no estado $l=4$ [19]

Após a realização de todo o algoritmo, ou seja, quando se concluir a descodificação de toda a sequência de entrada atinge-se o percurso de máxima verosimilhança pretendido. De facto não é possível selecionar o melhor percurso só no fim da mensagem, pois isso levaria a um tempo de atraso enorme no algoritmo (no caso em que se tem milhões de bits para descodificar). Desta forma é necessário truncar o algoritmo, isto é, obrigar o algoritmo a optar por um percurso ao fim de um certo atraso finito tolerável pelo sistema. Se se voltar cinco vezes o comprimento de restrição ($5N$) para trás, todos os percursos sobreviventes normalmente já partilham um trajeto comum. Verifica-se que este retrocesso no trajeto pouco degrada o desempenho do descodificador, sendo viável este processo. A estas cinco

vezes o comprimento de restrição dá-se o nome de janela de observação, ou janela de descodificação, de profundidade $5N$ [19].

No caso em que a janela de observação é iniciada e fechada no estado nulo, é necessário colocar na cauda da mensagem zeros, o que determina uma menor eficiência da comunicação, pois esses zeros não fazem parte da informação útil.

O algoritmo de Viterbi tem, a cada passo e de todos os 2^k percursos que convergem num estado, armazenar:

- 2^v percursos sobreviventes;
- 2^v métricas acumuladas.

onde v é a memória do codificador, como já se tinha visto na introdução dos códigos convolucionais.

A descodificação de Viterbi por máxima verosimilhança, pode ser aplicada a qualquer código que possa ser representado por treliça. Visto os códigos de blocos terem a possibilidade de se representarem com treliça, então o algoritmo pode ser aplicado não só aos códigos convolucionais, como também aos códigos de blocos.

5.3 BCJR

O BCJR e os códigos turbo

Os códigos turbo foram introduzidos a pensar numa descodificação iterativa, onde os códigos seriam concatenados e entrelaçados. Utilizava-se a concatenação em paralelo e para tal o uso também de codificadores convolucionais recursivos sistemáticos (RSC), já referidos anteriormente.

A descodificação turbo é realizada através de uma malha de realimentação, onde circulam informações brandas, pois a descodificação turbo aceita na entrada decisões brandas e produz de igual forma na saída, que dizem respeito a probabilidades calculadas com o algoritmo BCJR. Um sistema de descodificação turbo iterativa é constituído por dois descodificadores, um entrelaçador e um desentrelaçador, o que pode ser visualizado na Figura 41. Os descodificadores podem funcionar com o algoritmo BCJR, onde por vezes podem ser implementados com uma de duas aproximações possíveis deste algoritmo. Estas aproximações são denominadas de log-MAP e max-log-MAP, as quais trabalham com logaritmos por forma a simplificar as tradicionais multiplicações e divisões do algoritmo BCJR em somas e subtrações respetivamente. No que diz respeito ao entrelaçador e desentrelaçador, estes são introduzidos por forma a obter um melhor ganho de codificação, e será referido e detalhado mais aquando a introdução do algoritmo BCJR.

A determinada probabilidade de erro, o sistema deixa de conseguir obter melhor desempenho, pois atingiu um nível de saturação, a partir do qual não se consegue obter mais ganhos provenientes de iterações adicionais. A Figura 42 mostra um exemplo de BER (Bit Error Ratio) em função da relação sinal-ruído para consecutivas iterações.

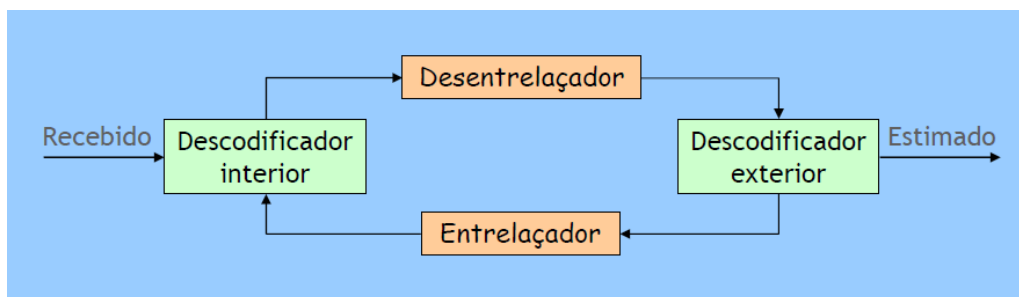


Figura 41 - Diagrama de blocos de decodificação iterativa série [23]

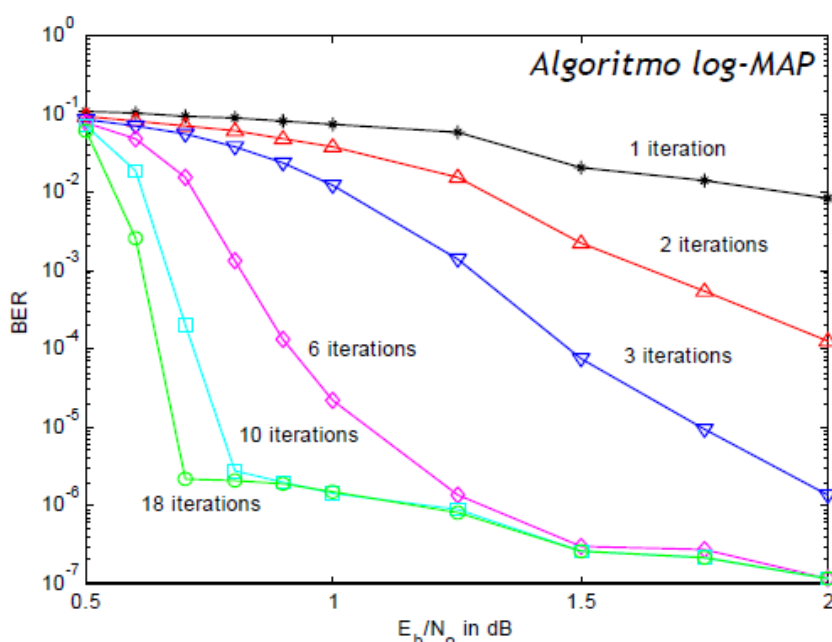


Figura 42 - Desempenho da decodificação iterativa [23]

Em resumo, pode dizer-se que, o princípio turbo faz com que o desempenho de um recetor melhore significativamente utilizando:

- Codificação concatenada no emissor;
- Entrelaçamento e desentrelaçamento;
- Deteção e decodificação iterativa no recetor.

A codificação e decodificação turbo com BCJR e max-log-MAP

Nesta fase, após a codificação, é necessário trocar os valores lógicos 0 e 1 por dois valores reais simétricos -1 e +1 respetivamente. A codificação pode ser realizada com dois codificadores sistemáticos recursivos, como se vê na Figura 43, e após isso passa-se o sinal por um canal onde será adicionado ruído.

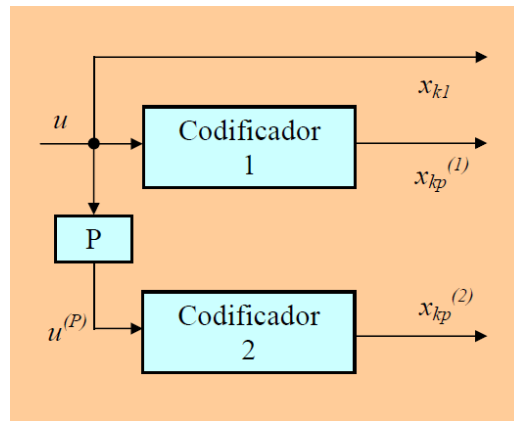


Figura 43 - Codificação turbo em paralelo[22]

Na Figura 43 está incluído um entrelaçador que deve ter grande profundidade de entrelaçamento. Este entrelaçador tem como função “baralhar” a informação, recebida à entrada, de acordo com o padrão de entrelaçamento aleatório, mas que depois terá de ser o mesmo durante todo o processo de descodificação. O entrelaçamento é realizado com uma matriz quadrada de valores binários, onde cada coluna apenas contém um valor ‘1’ correspondente ao número da linha indicada no padrão entrelaçamento. A probabilidade de bit errado obtida com turbo-códigos é reduzida, quanto maior for o comprimento do entrelaçador, no caso de codificadores RSC. Diz-se assim que o entrelaçador proporciona um ganho de entrelaçamento [19].

A razão de probabilidades a priori é dada por

$$\frac{P(u_k=+1)}{P(u_k=-1)} \quad (5-10)$$

Onde u_k é o bit de informação. Assim vem que o logaritmo natural é

$$L(u_k) = \ln \frac{P(u_k=+1)}{P(u_k=-1)} \quad (5-11)$$

Este $L(u_k)$ designa-se por LLR (Log-Likelihood Ratio). Se os bits $u_k = \pm 1$ forem equiprováveis LLR é zero, o que acontecerá na primeira iteração da descodificação turbo, no primeiro descodificador. Como a mensagem passa num canal AWGN (ruído gaussiano branco), o descodificador recebe uma mensagem com valores reais aleatórios e para tentar obter a mensagem original, então inicia-se o algoritmo BCJR que calcula a razão de probabilidade à posteriori

$$\frac{P(u_k=+1|y)}{P(u_k=-1|y)} \quad (5-12)$$

e que também pode ser descrito através de LLR a posteriori [19]

$$L(u_k|y) = \ln \frac{P(u_k=+1|y)}{P(u_k=-1|y)} \quad (5-13)$$

onde o sinal positivo ou negativo de LLR indica se o bit enviado foi +1 ou -1 respetivamente. Quanto mais afastado o valor de $L(u_k|y)$ do limiar de decisão nulo, mais confiança existe na estimativa de bit. Depois disto a informação contida em $L(u_k|y)$ é passada a outro elemento

de decodificação e aí se $L(u_k|y) < 0$, então o bit transmitido foi -1, caso contrário, o bit transmitido foi +1.

Por norma os códigos turbo são constituídos por dois codificadores RSC iguais onde cada palavra de código no instante k é constituída pelo bit sistemático x_{k1} , pelos bits paridade $x_{kp}^{(1)}$ e $x_{kp}^{(2)}$. Após a passagem pelo canal, a palavra de código dá origem ao vetor $[y_{k1} y_{kp}^{(1)} y_{kp}^{(2)}]$. Estes valores são depois atribuídos a dois decodificadores, onde P e P^{-1} correspondem ao entrelaçador e desentrelaçador respetivamente, como se pode ver na Figura 44.

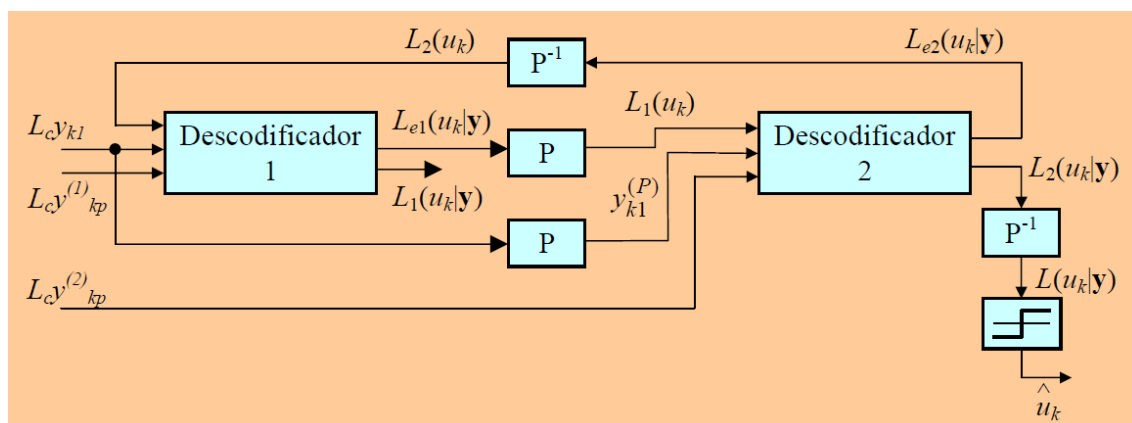


Figura 44 - Diagrama de blocos de decodificador turbo para concatenação em paralelo [22]

Para o caso da concatenação em série, mostra-se o diagrama de blocos do decodificador turbo apresentado na Figura 41.

Através das mensagens anteriormente descritas, e a partir dos diagramas de blocos, consegue-se chegar a

$$L(u_k|y) = L(u_k) + L_c y_{k1} + L_e(u_k) \quad (5-14)$$

onde L_c é a medida de fiabilidade do canal, e num canal AWGN, é dada por

$$L_c = 4aR_c \frac{E_b}{N_0} \quad (5-15)$$

onde a é a amplitude de *fading*, onde se não existir *fading* no canal então $a = 1$, R_c refere-se à taxa do código convolucional e E_b/N_0 representa a relação sinal-ruído do canal AWGN em questão.

Desta forma a decodificação iterativa pode descrever-se como um processo constituído pelas seguintes fases:

- Na primeira iteração o decodificador 1 recebe $L(u_k) = 0$ e fornece a informação extrínseca $L_{e1}(u_k|y)$ sobre o bit sistemático (y_{k1}), que obteve do primeiro bit paridade ($y_{kp}^{(1)}$).
- Seguido do entrelaçamento a informação extrínseca $L_{e1}(u_k|y)$ do decodificador 1, é entregue ao decodificador 2 como $L_2(u_k)$, cujo é um dado melhor que $L(u_k)$. De seguida o decodificador 2 cede $L_{e2}(u_k|y)$, que é a sua própria informação extrínseca sobre o bit sistemático (y_{k1}), mas agora baseada no outro bit de paridade ($y_{kp}^{(2)}$). Concluído o desentrelaçamento, esta informação fica a cargo do decodificador 1 como $L_1(u_k)$, a qual é um palpite ainda bem mais elaborado à cerca dos anteriores $L(u_k)$.

- Começa, então, uma nova iteração e repete-se todo o processo da iteração 1, mas agora o decodificador 1 tem à entrada uma $L(u_k)$ bem mais apurada que na iteração anterior, o que de facto vai criar uma maior confiança na decisão do bit recebido.
- Depois de um número pré-determinado de iterações, a LLR $L_2(u_k|y)$ na saída do decodificador 2, é desentrelaçada e entregue como $L(u_k|y)$ ao dispositivo de decisão rígida, o qual estima o bit de informação de acordo exclusivamente com o sinal, positivo ou negativo [19].

Como já foi referido, os decodificadores turbo funcionam com um dos três algoritmos enunciados (BCJR, log-MAP ou max-log-MAP), o qual calcula a LLR à posteriori ($L(u_k|y)$). Para tal é necessário efetuar cálculos sucessivos que respeitam a treliça do codificador convolucional utilizado. O algoritmo BCJR é conhecido por *forward-backward algorithm*, pois ao contrário do algoritmo de Viterbi, que encontra o percurso de máxima verosimilhança e conclui através deste qual a sequência enviada, o BCJR faz a estimação dos bits de entrada do codificador instante a instante [19]. Assim sendo, será introduzido inicialmente o princípio de funcionamento do algoritmo BCJR e após isto a sua correspondente aproximação max-log-MAP.

Como já é sabido o objetivo do BCJR é o cálculo da LLR à posteriori $L(u_k|y)$, que vai determinar qual o bit recebido no decodificador, cuja é dada por (5-13), e que para a obtenção desta LLR é necessário o cálculo prévio de

$$P(u_k = +1|y) = \sum_{R_1} P(s', s, y) = \sum_{R_1} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) \quad (5-16)$$

e de

$$P(u_k = -1|y) = \sum_{R_0} P(s', s, y) = \sum_{R_0} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) \quad (5-17)$$

onde $P(s', s, y)$ representa a probabilidade conjunta de a sequência recebida ser y , s corresponde ao estado atual, s' ao estado anterior e y é o símbolo recebido no decodificador, tal como se pode notar na treliça da Figura 35. Sabe-se também que R_0 diz respeito às transições a traço contínuo, as quais são provocadas por um bit de mensagem -1 , e R_1 às transições a tracejado que são provocadas por um bit de mensagem $+1$. Desta forma tem-se que

$$L(u_k|y) = \ln \frac{\sum_{R_1} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)}{\sum_{R_0} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)} \quad (5-18)$$

No cálculo da probabilidade conjunta $P(s', s, y)$ é necessário em cada instante k associar o passado, presente e futuro do símbolo y_k às probabilidades α, γ e β respetivamente. Inicialmente calcula-se o valor de todas as probabilidades $\gamma_k(s', s)$, no caso de canais AWGN, da seguinte forma

$$\gamma_k(s', s) = C_k \exp\left(\frac{u_k L(u_k)}{2}\right) \times \exp\left(\frac{L_c}{2} \sum_{l=1}^n x_{kl} y_{kl}\right) \quad (5-19)$$

onde a C_k pode ser atribuído o valor unitário, pois aquando o cálculo da LLR à posteriori este valor está presente no numerador e denominador e anula-se. O valor de u_k é dado pelo

transição que ocorre na treliça, se o traço da transição for a cheio $u_k = -1$, no caso contrário $u_k = +1$.

Após obtidos todos os valores das probabilidades $\gamma_k(s', s)$ é então possível iniciar o cálculo recursivo das probabilidades $\alpha_k(s)$, tendo em atenção as condições iniciais na treliça

$$\alpha_k(s) = \sum_{s'} \gamma_k(s', s) \alpha_{k-1}(s') \quad (5-20)$$

onde as condições iniciais são

$$\alpha_0(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases}$$

Os valores de α são calculados consoante a receção da sequência y e desta forma diz-se que o sentido é *forward*, ou seja do início para o fim da treliça. Esta situação é devida à necessidade de para um determinado $\alpha_k(s)$ ser necessário o $\alpha_{k-1}(s')$. Após a conclusão do cálculo de todos os $\alpha_k(s)$ é possível então começar a calcular os $\beta_{k-1}(s')$ com a seguinte expressão

$$\beta_{k-1}(s') = \sum_s \gamma_k(s', s) \beta_k(s) \quad (5-21)$$

com condições iniciais

$$\beta_N(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases}$$

No cálculo de $\beta_{k-1}(s')$ cai-se no caso *backward* do algoritmo, ou seja inicia-se o cálculo do fim para o início da treliça. Os valores iniciais $\alpha_0(s)$ e $\beta_N(s)$ implicam o início e término da treliça no estado nulo.

Devido ao uso de exponenciais no algoritmo BCJR cria-se o problema da instabilidade numérica, o que significa que após algumas iterações o algoritmo gera *underflow* ou *overflow*. Para evitar esta situação não se usa diretamente α e β que se calcularam recursivamente, mas sim seus valores normalizados através da soma em cada instante. Para o cálculo destes valores normalizados calcula-se inicialmente $\alpha'_k(s)$ e $\beta'_{k-1}(s')$ como se fosse o cálculo recursivo anterior

$$\alpha'_k(s) = \sum_{s'} \gamma_k(s', s) \alpha_{k-1}(s') \quad (5-22)$$

$$\beta'_{k-1}(s') = \sum_s \gamma_k(s', s) \beta_k(s) \quad (5-23)$$

De seguida efetua-se o somatório $\sum_s \alpha'_k(s)$ e $\sum_{s'} \beta'_{k-1}(s')$ em cada instante. Agora sim é possível a determinação dos valores normalizados $\alpha_k(s)$ e $\beta_{k-1}(s')$.

$$\alpha_k(s) = \frac{\alpha'_k(s)}{\sum_s \alpha'_k(s)} \quad (5-24)$$

$$\beta_{k-1}(s') = \frac{\beta'_{k-1}(s')}{\sum_{s'} \beta'_{k-1}(s')} \quad (5-25)$$

Calcula-se para cada instante o somatório

$$\sum_{R0,R1} p_k = \sum_{R0} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) + \sum_{R1} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) \quad (5-26)$$

Desta forma é possível obter a probabilidade conjunta normalizada

$$P_{norm}(s', s, y) = \frac{P(s', s, y)}{\sum p_k} \quad (5-27)$$

Finalmente, o objetivo, a LLR à posteriori

$$L(u_k|y) = \ln \frac{\sum_{R1} P_{norm}(s', s, y)}{\sum_{R0} P_{norm}(s', s, y)} \quad (5-28)$$

5.4 Max-Log-MAP

Como já foi dito o algoritmo BCJR exige bastante tempo para o cálculo da LLR à posteriori, devido às multiplicações que efetua. No intuito de reduzir a complexidade deste algoritmo criaram-se algoritmos como o SOVA (Soft-Output Viterbi Algorithm), log-MAP e max-log-MAP [19]. Neste último algoritmo, que será abordado, as multiplicações do algoritmo BCJR são trocadas por adições, continuando na mesma a ser um forward-backward algorithm. Como são usadas estas aproximações, existe um desempenho ligeiramente inferior. Quanto ao algoritmo log-MAP, este tem um desempenho muito parecido com o BCJR, pois usa fórmulas exatas.

O Max-log-MAP utiliza como base a função

$$\max(x, y) \quad (5-29)$$

na decisão entre dois valores, degradando a performance em 0.5dB comparado com o algoritmo Log-MAP [24].

Em relação ao BCJR são definidas novas variáveis (A, B e Γ) para realizar a posterior simplificação:

$$A_k(s) = \ln \alpha_k(s) \quad (5-30)$$

$$B_k(s) = \ln \beta_k(s) \quad (5-31)$$

$$\Gamma_k(s', s) = \ln \gamma_k(s', s) \quad (5-32)$$

Desta forma chega-se as expressões que dão origem às somas e subtrações que possibilitam a aproximação ao algoritmo BCJR:

$$A_k(s) = \max_{s'} (A_{k-1}(s') + \Gamma_k(s', s)) \quad (5-33)$$

$$B_{k-1}(s') = \max_s (B_k(s) + \Gamma_k(s', s)) \quad (5-34)$$

$$\Gamma_k(s', s) = \ln(C_k) + \frac{u_k L(u_k)}{2} + \frac{L_c}{2} \sum_{l=1}^n x_{kl} y_{kl} \quad (5-35)$$

onde $\ln(C_k)$ pode ser desprezado, pois, será anulado na subtração no cálculo da LLR à posteriori.

Para as expressões de $A_k(s)$ e $B_{k-1}(s')$, é necessário atender às condições iniciais:

$$A_0(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases}$$

e

$$B_N(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases}$$

Desta forma é possível alcançar a expressão da LLR à posteriori $L(u_k|y)$, onde a anterior divisão na expressão da LLR passa a ser uma subtração

$$L(u_k|y) = \max_{R1} (A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)) - \max_{R0} (A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)) \quad (5-36)$$

5.5 Log-MAP

Este algoritmo é também uma aproximação do BCJR e que consegue obter um desempenho idêntico a este. O algoritmo log-MAP utiliza logaritmos, o que reduz a complexidade em relação ao BCJR, pois tal como o max-log-MAP, utiliza somas e subtrações ao contrário das tradicionais multiplicações e divisões do BCJR. O único ponto em que difere do algoritmo max-log-MAP é no cálculo da função *max* [19]

$$\max(x, y) + \ln(1 + e^{-|a-b|}) \quad (5-37)$$

5.6 Entrelaçador e desentrelaçador

Como para a codificação e descodificação turbo é necessário um entrelaçador e respetivo desentrelaçador, então será feita uma breve referência a este assunto. Estes servem para que erros de rajada se apresentem descorrelacionados no recetor [25]. O entrelaçamento pode ser realizado a partir de um vetor-linha, denominado de padrão de entrelaçamento ou padrão de permutação, que tem o tamanho da mensagem. Uma outra hipótese é utilizar uma matriz binária quadrada, designada por matriz de permutação, onde cada coluna contém apenas um

“1”. A linha da matriz em que se coloca, numa dada coluna, este valor lógico “1”, corresponde ao número definido no vetor-linha. A Figura 45 apresenta um padrão de entrelaçamento e a correspondente matriz de permutação.

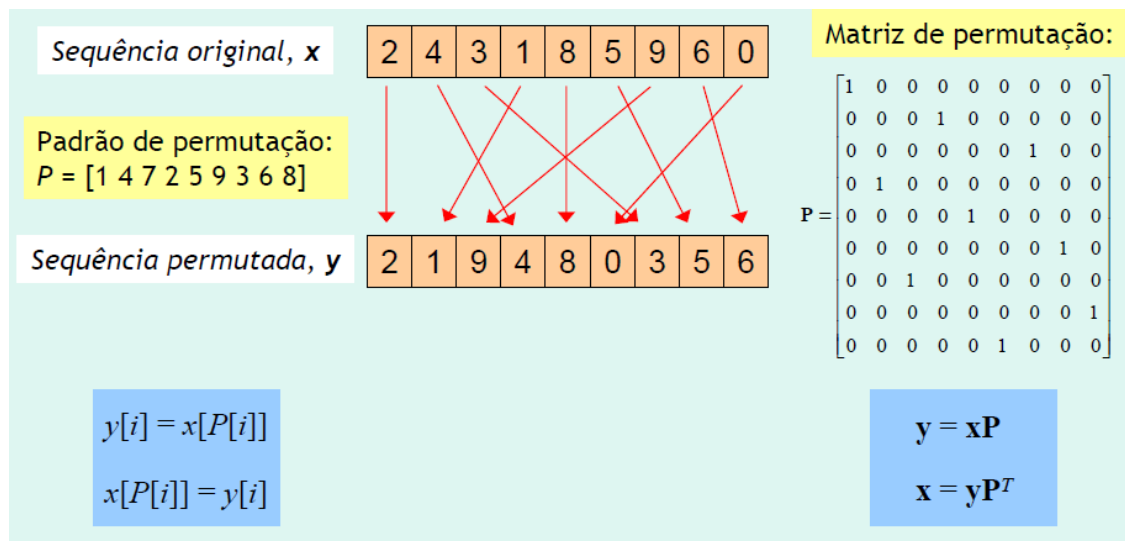


Figura 45 - Entrelaçamento de uma mensagem[23]

Como também é possível ver na Figura 45, a mensagem entrelaçada, y , é obtida a partir da multiplicação matricial $y = xP$ e o correspondente desentrelaçamento é obtido com a expressão $x = yP^T$, onde x é o vetor-linha que contém a mensagem original, y é o vetor correspondente à mensagem entrelaçada, P é a matriz de permutação e P^T é a matriz de permutação transposta.

No âmbito da codificação turbo, estes entrelaçadores estão associados a codificadores convolucionais recursivos, de modo que devem ser o mais profundos possível, isto é, devem ser longos. Sabe-se que a probabilidade de bit errado, em turbo-códigos, reduz-se com o aumento do comprimento do entrelaçador, isto se os codificadores forem do tipo RSC tal como já foi referido. Desta forma diz-se que o entrelaçador proporciona um ganho de entrelaçamento [19]. Um outro aspeto a ter em consideração é a latência acrescentada no sistema devido ao tamanho do entrelaçador. Devido a isto é necessário criar um compromisso entre a latência e o BER na escolha do tamanho deste entrelaçador [26].

5.7 Resumo

Em relação a algoritmos de descodificação foram apresentadas diversas opções, e o estudo incidiu mais sobre o algoritmo BCJR e suas variantes, pois estes serão utilizados no trabalho.

A conclusão deste capítulo remete para o cuidado no uso destes algoritmos devido à possível instabilidade numérica e leva ainda a um mais provável uso do log-MAP e do max-log-MAP. Outro aspeto a realçar é a necessidade da utilização de entrelaçadores profundos.

No próximo capítulo chega-se finalmente à igualização turbo que é o objetivo final deste trabalho.

Capítulo 6

Igualização Turbo

6.1. Introdução

A igualização turbo foi proposta por Douillard em 1995 para um sistema de código convolucional de taxa $1/2$ e modulação BPSK. Douillard demonstrou que a igualização turbo é capaz de suavizar os efeitos da interferência intersimbólica (ISI), sabendo que a resposta impulsional do canal é conhecida. O melhoramento da performance foi obtido através da igualização de canal e da descodificação iterativa do canal [27].

Com a crescente procura pela largura de banda, os sistemas atuais requerem o aumento da eficiência espectral pela necessidade de códigos de taxa elevada. Esta foi uma motivação para a pesquisa dentro dos códigos de blocos, cujos foram demonstrados por Hagenauer, para superar os códigos turbo convolucionais usando códigos convolucionais perfurados de elevada taxa, quando a taxa de código é superior a $2/3$ [27].

Outro método de gerar códigos turbo de taxa elevada foi proposta por Açikel e Ryan, onde uma codificação turbo de taxa $1/3$ corresponde a dois codificadores convolucionais de taxa $1/2$ perfurados. Este tipo de códigos têm mostrado ser melhores que os códigos turbo propostos por Hagenauer. Os padrões de perfuração foram otimizados para transmissão sobre um canal não dispersivo adicionado de ruído branco Gaussiano (AWGN) [27].

O objetivo da igualização turbo é reduzir o efeito da interferência intersimbólica (ISI) causada pela dispersão temporal do canal [25]. O sistema inicial foi abordado em [28] dando ênfase à ideia da descodificação turbo e relacionando esta com a igualização e descodificação [29]. O primeiro sistema de igualização turbo é composto pelo codificador externo (codificador convolucional), codificador interno (modulador, canal de transmissão e desmodulador) e no receptor é constituído por um descodificador interno (igualizador) e por um descodificador externo (descodificador convolucional). Neste esquema os símbolos transmitidos são codificados através do codificador externo, passando pelo codificador interno e na receção o processo da igualização turbo realiza sucessivas trocas de informação entre o descodificador interno e o descodificador externo, tal como ocorre na descodificação turbo [25, 30]. Foi conseguido através deste sistema resultados que praticamente eliminam toda a ISI do canal sem o aumento da SNR. Desta forma conclui-se que o seu desempenho se

assemelha ao desempenho de um sistema com codificação de canal para um canal AWGN e sem ISI [29].

O exemplo, meramente ilustrativo, de um sistema deste tipo pode ser visualizado na Figura 46.

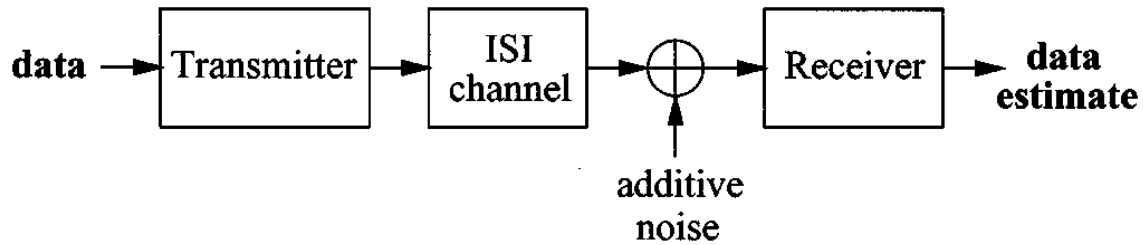


Figura 46 - Exemplo de um sistema com interferência intersimbólica [31]

O bloco transmissor é composto pelo codificador convolucional, por um entrelaçador e um modulador BPSK que envia a informação para o canal, tal como sugere Figura 47.

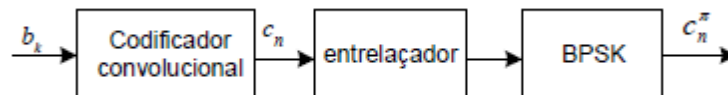


Figura 47 - Constituição do bloco de transmissão [25]

O modelo discreto do canal com interferência intersimbólica (ISI), pode ser representado por uma treliça associada a uma cadeia de Markov [25], sendo o seu codificador composto por M atrasadores, o que corresponde a uma treliça de 2^M estados. A Figura 48 mostra o exemplo de um canal deste tipo.

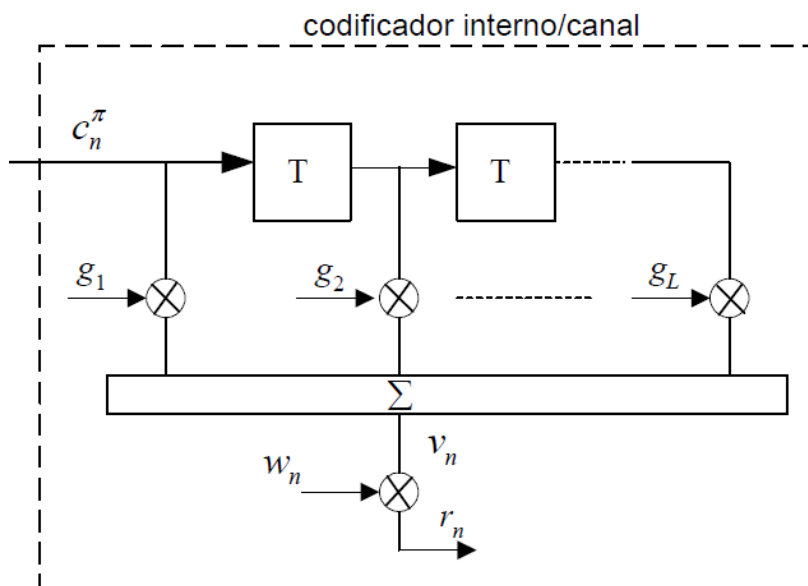


Figura 48 - Representação de um canal com ISI [25]

Quanto ao bloco recetor pode ser decomposto em três módulos mais pequenos (igualizador, desentrelaçador e descodificador convolucional), como mostra a Figura 49

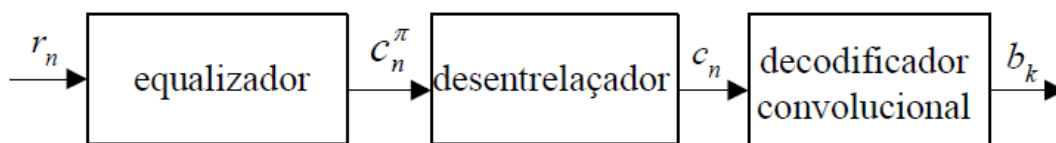


Figura 49 - Diagrama de blocos do recetor [25]

6.2. Codificador externo

O codificador externo é composto pelo codificador convolucional que pode ser um codificador simples de taxa 1/2, tal como o apresentado na Figura 50. Este codificador é caracterizado por uma treliça (Figura 51) que representa a transição de estados do mesmo. Este codificador coloca na sua saída os bits c_n , que serão entrelaçados e de seguida modulados no bloco BPSK da Figura 47, dando origem, na sua saída, aos bits c_n^π .

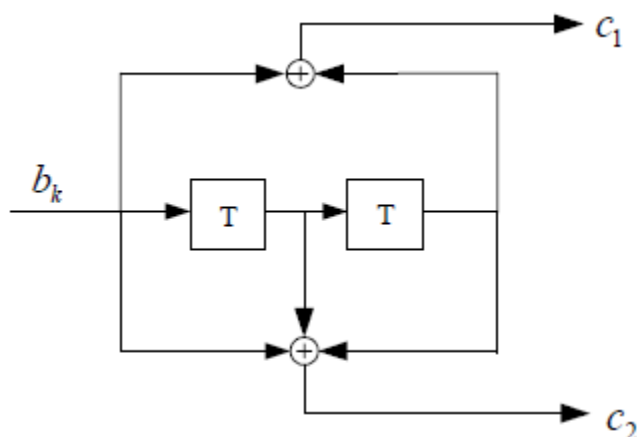


Figura 50 - Codificador convolucional de taxa 1/2 [25]

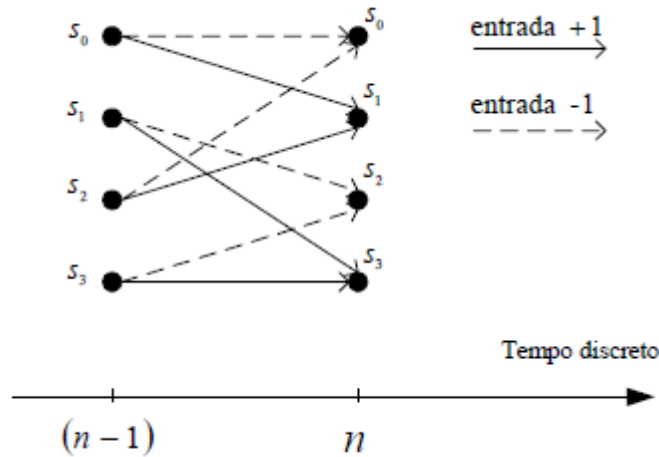


Figura 51 - Treliça do codificador convolucional [25]

6.3. Codificador interno

O codificador interno é, como já foi referido, formado pelo modulador, pelo canal de transmissão e pelo desmodulador. Será essencial focar o estudo sobre o canal com ISI, pois este é o equivalente a um codificador convolucional, com uma treliça associada.

O canal ISI, associado a um codificador, pode ser encarado a um codificador de taxa 1 no corpo dos reais [23].

6.4. Igualizador e decodificador

Na igualização turbo, tanto igualizador como decodificador são SISO (*soft-input soft-output*), desta forma todas as probabilidades, durante o processo iterativo, são tomadas de forma branda, exceto na última iteração onde a decisão para a estimação da sequência recebida é realizada de forma rígida.

O igualizador tem como função transferência [1]

$$H_{eq}(\omega) = \frac{e^{-j\omega t_0}}{H_c(\omega)} \tag{6-1}$$

O igualizador tem duas entradas, r_n (vetor de amostras recebidas do canal com ISI) e $L_a^E(c_n^\pi)$ (LLR à priori), como se pode ver na Figura 52. Na sua saída é colocada uma LLR à posteriori ($L^E(c_n^\pi)$), calculada com um dos algoritmos anteriormente descritos (BCJR, log-MAP ou max-log-MAP). Na primeira iteração a LLR à priori terá o valor zero, pois, considera-se os bits equiprováveis.

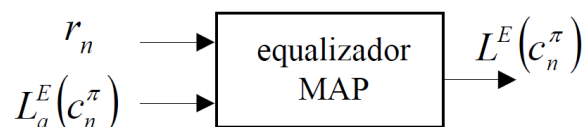


Figura 52 - Igualizador MAP (*Maximum a posteriori*) [25]

A partir de $L^E(c_n^\pi)$ é possível obter a informação extrínseca que será enviada para o decodificador. Esta informação pode ser calculada da seguinte forma

$$L_{ext}^E(c_n^\pi) = L^E(c_n^\pi) - L_a^E(c_n^\pi) \quad (6-2)$$

Esta informação extrínseca pode ser interpretada como o ganho de informação obtido no processo de igualização ou descodificação [25, 28], ou como a LLR à saída do decodificador SISO menos a LLR da entrada.

Este igualizador funciona numa treliça dinâmica discreta descrita pelo canal ótico, com memória $2m + 1$, sendo $2m$ o número de bits que influenciam o bit observado em ambos os lados. Quanto mais estados a treliça tem, maior é a resolução, ou qualidade, do igualizador [1]. Nos cálculos das probabilidades, este assemelha-se ao algoritmo BCJR, exceto no cálculo das probabilidades de transição da treliça, cujas são dadas por [1]

$$p(y_k | y_{(s',s)}) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[-\frac{(y_k - y_{(s',s)})^2}{2\sigma^2} \right] \quad (6-3)$$

onde σ é a variância no domínio elétrico associado a ruído.

Após todos os cálculos o igualizador envia para o decodificador (Figura 53) a informação extrínseca.

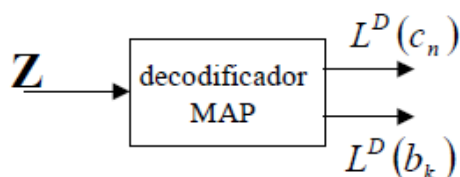


Figura 53 - Decodificador MAP [25]

Este decodificador recebe na entrada o vetor Z composto pela estimativa das probabilidades a priori dos símbolos c_n . Após isto este decodificador serve-se do algoritmo BCJR para calcular a probabilidade a posteriori $L^D(c_n)$.

Um exemplo de um igualizador turbo é representado na Figura 54. Este igualizador tem como princípio a descodificação iterativa série.

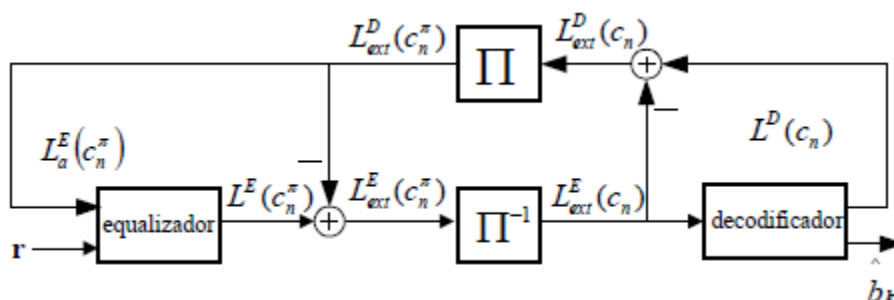


Figura 54 - Igualizador turbo série [25]

Por outro lado como este é um processo iterativo, análogo ao da decodificação turbo, pode-se representar cada iteração por um módulo tal como sugere a Figura 55.

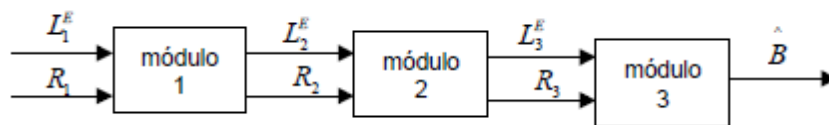


Figura 55 - Igualização iterativa [25]

Na última iteração é calculada a sequência transmitida através de decisões rígidas baseadas na probabilidade a posteriori calculado pelo decodificador.

6.5. Resumo

Neste capítulo foi revista literatura que se foca em igualização turbo por forma a perceber o princípio desta.

Pode-se essencialmente concluir que se necessita de um igualizador e um decodificador, neste caso BCJR, no módulo recetor. Deste modo a igualização é realizada em série e a codificação apenas necessita de codificador RSC.

Outro aspeto importante a concluir é a representação do canal por uma treliça dinâmica que será utilizada no igualizador para combater a ISI.

A próxima secção, ou capítulo, desta dissertação dirige-se à descrição do trabalho realizado, com os correspondentes circuitos realizados, resultados obtidos, uma conclusão sobre estes e uma breve proposta de trabalho futuro.

Capítulo 7

Trabalho realizado

7.1. Introdução

Por forma a atingir o objetivo da igualização turbo num sistema de comunicações óticas, o trabalho foi dividido em pequenos módulos para no fim fazer a junção destes e concretizar o trabalho proposto.

Os parâmetros utilizados para as simulações realizadas são os abaixo apresentados:

Modulador MZM (Mach Zhender):

- $V_{\pi} = 5V$

Fibra monomodo:

- $L=100$ km
- $\alpha= 0,2$ dB/km
- $D = 17.461$ ps/nm.km

Fotodiodo PIN:

- Resistência de carga = 50Ω
- Temperatura = 273 K
- Responsividade = $0,5$ A/W

Parâmetros da simulação:

$$\lambda = 1550\text{nm}$$

$$f_c = 5\text{GHz}$$

$$R_b = 1\text{Gb/s}$$

Ao longo deste trabalho o objetivo passou sempre por iniciar as simulações com um canal de ruído branco Gaussiano e após isto passar para a fibra e tirar conclusões.

Antes de se iniciar a demonstração de resultados é importante referir que em todos os gráficos, quando surge uma curva teórica, esta representa o valor teórico para um canal AWGN.

7.2. Modulação BPSK

Nesta fase inicial foi definido como se processaria a modulação do sistema. Após a geração dos bits de entrada do sistema, estes são convertidos num sinal NRZ (*Non-Return to Zero*), para de seguida modularem a portadora. Após isto o sinal passa num canal, AWGN ou fibra, e depois de desmodulado é realizado um *Integrate and Dump* através de um filtro para ser possível realizar a decisão, tal como sugere a Figura 17.

O débito binário associado ao é de 1Gb/s e é amostrado à frequência de 100GHz. Nesta etapa também se fixou a frequência da portadora em 5GHz.

7.2.1. BPSK sobre um canal AWGN

Nesta fase simulou-se um sistema, equivalente ao da Figura 56, e obteve-se a taxa de erros em função da relação sinal-ruído que pode ser visualizada na Figura 57. Perante os resultados obtidos, verifica-se uma correspondência entre a curva teórica e a curva obtida na simulação, o que era esperado.

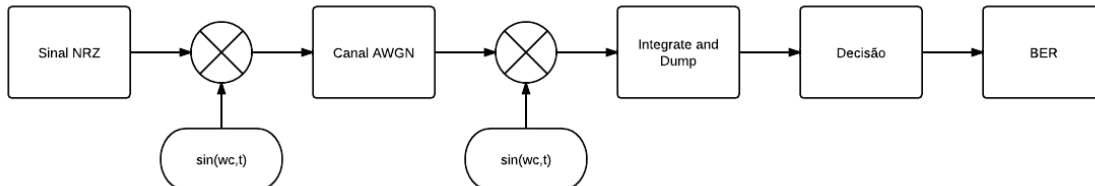


Figura 56 - Sistema com modulação BPSK sobre canal AWGN

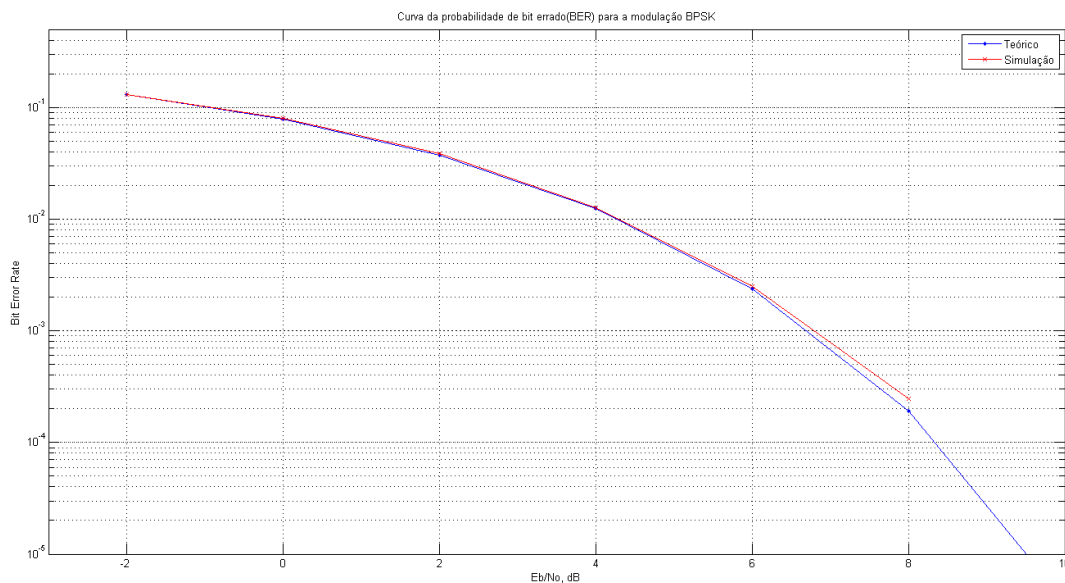


Figura 57 - BER em função do SNR para um canal AWGN com modulação BPSK

7.2.2. BPSK sobre fibra

Com base no sistema da Figura 56, adicionou-se um modulador MZM (Mach-Zhender) [10, 32-34], o canal foi substituído por fibra ótica e por fim colocou-se um fotodiodo PIN. Esta configuração pode ser vista na Figura 58. A fibra ótica a utilizar tem como comprimento $L = 100\text{km}$, parâmetro de dispersão $D = 17,461 \times 10^{-6}$ e atenuação $\alpha = 0,2\text{dB/Km}$. Em relação ao MZM, este é colocado no sistema com o objetivo de modular o sinal e para que este possa ser transmitido pela fibra ótica. Este pode introduzir efeitos não lineares no sistema, mas não significativos neste caso, em comparação com a dispersão introduzida pela fibra. Estes efeitos não lineares evoluem com o aumento do índice de modulação do MZM. O índice de modulação é dado pelo quociente entre a tensão de entrada (V_{IN}) e a tensão de controlo (V_{π}). Neste caso, o MZM está dimensionado por forma a que não introduza efeitos não-lineares no sistema.

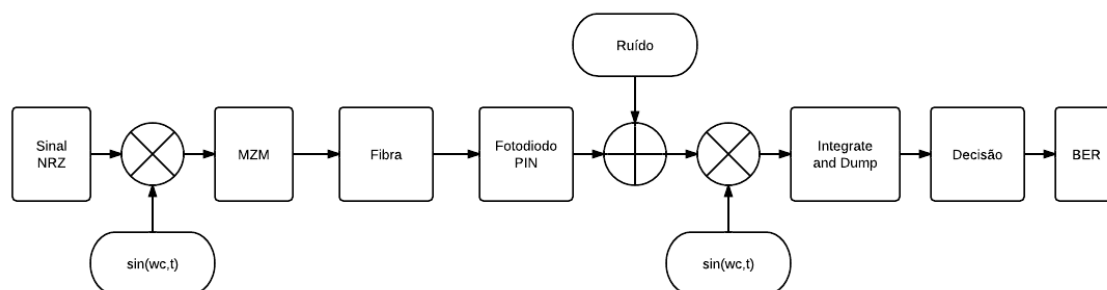


Figura 58 - Modulação BPSK sobre fibra

Através deste sistema, foi possível obter a curva de erros da Figura 59. O desvio das curvas deve-se à introdução da fibra no sistema, a qual introduz dispersão e consequentemente aumenta a taxa de erros para o sistema.

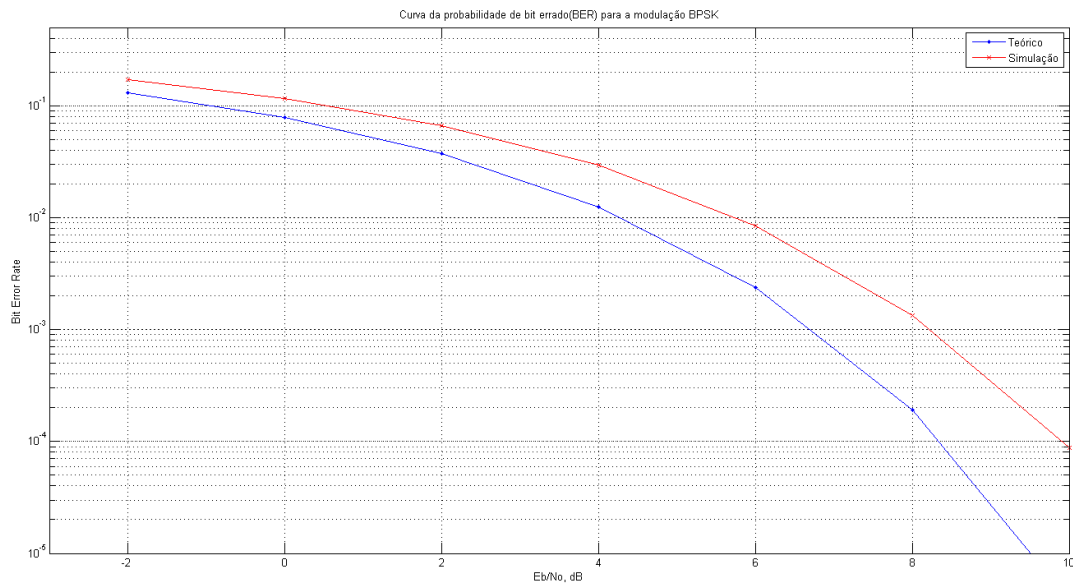


Figura 59 - BER em função de E_b/N_0 para modulação BPSK em fibra

7.3. Algoritmos de decodificação

Para esta etapa selecionou-se um codificador convolucional (2,1,3) com matriz geradora [7 5], tal como o apresentado na Figura 29. Este codificador é colocado no sistema antes do modulador e depois terá, após passagem pelo canal de transmissão e após desmodulação, um decodificador.

7.3.1. Algoritmo de Viterbi

A simulação com o algoritmo de Viterbi serviu apenas para realizar uma comparação entre este e o algoritmo BCJR, tentar perceber quais as diferenças na sua implementação e depois de que forma se repercutem nos resultados práticos.

Nestas simulações usou-se *hard-decision*. Apesar de o trabalho incidir em decodificadores SISO (soft-input soft-output), apenas se pretende uma linha de orientação dos resultados conseguidos por Viterbi. Indispensável será referir que a modulação BPSK se mantém nos próximos sistemas.

AWGN

Após as simulações de um sistema com o codificador convolucional, já referido, e com a decodificação Viterbi *hard-decision* (Figura 60) os resultados obtidos são os da Figura 61. A curva de simulação obtida está um pouco abaixo do que se pretendia, isto comparando com a curva teórica para Viterbi *hard-decision* sobre um canal AWGN.

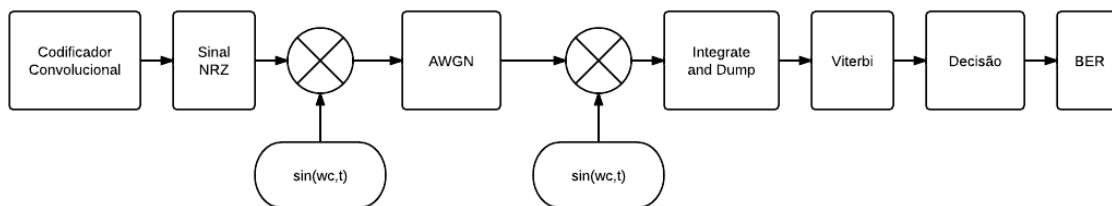


Figura 60 - Descodificação de Viterbi para canal AWGN

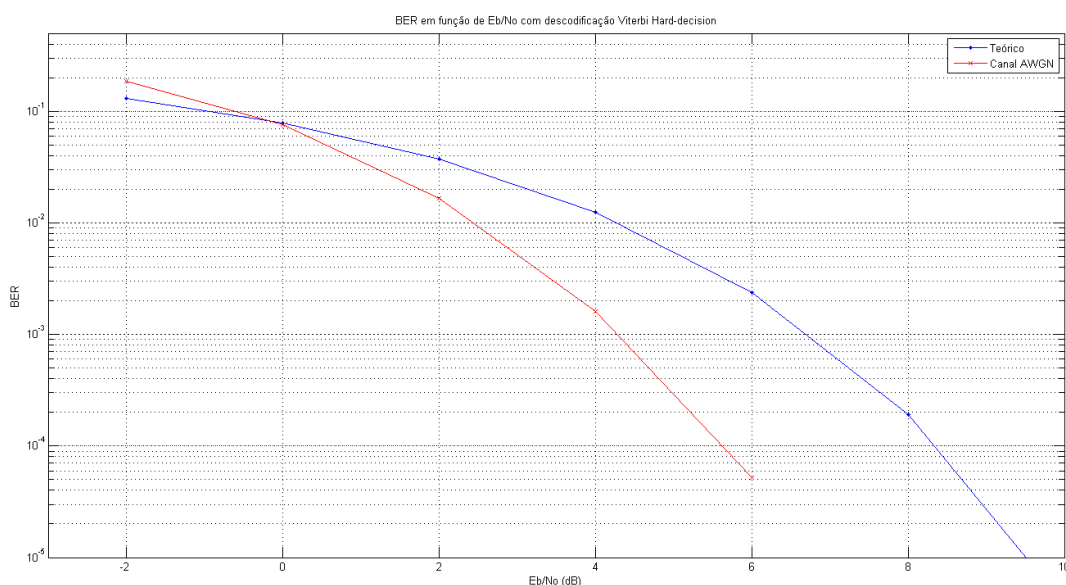


Figura 61 - Descodificação de Viterbi *hard-decision* após passagem por canal AWGN

Fibra

Neste sistema com fibra, e à semelhança da Figura 60 foi necessário introduzir, para além do canal fibra, um modulador mach-zhender e um fotodiodo PIN, tal como sugere a Figura 62. Concluída a simulação obteve-se o gráfico da Figura 63. Deste gráfico conclui-se que até à relação sinal-ruído 4dB o resultado situa-se um pouco abaixo do esperado, mas para 6 e 8dB os resultados já se encontram muito próximo do esperado.

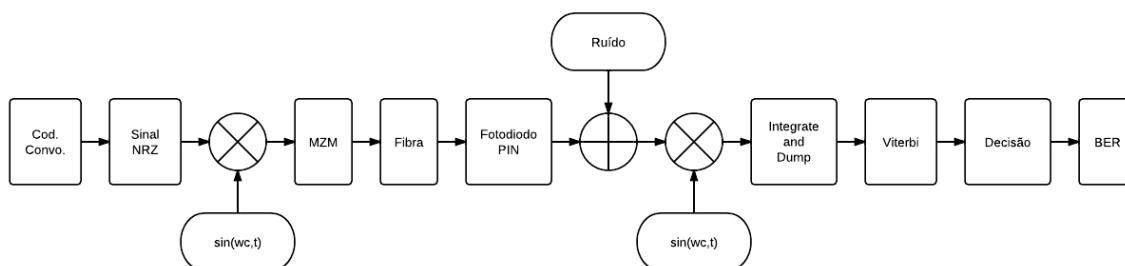


Figura 62 - Circuito para algoritmo de Viterbi com fibra

62 Trabalho realizado

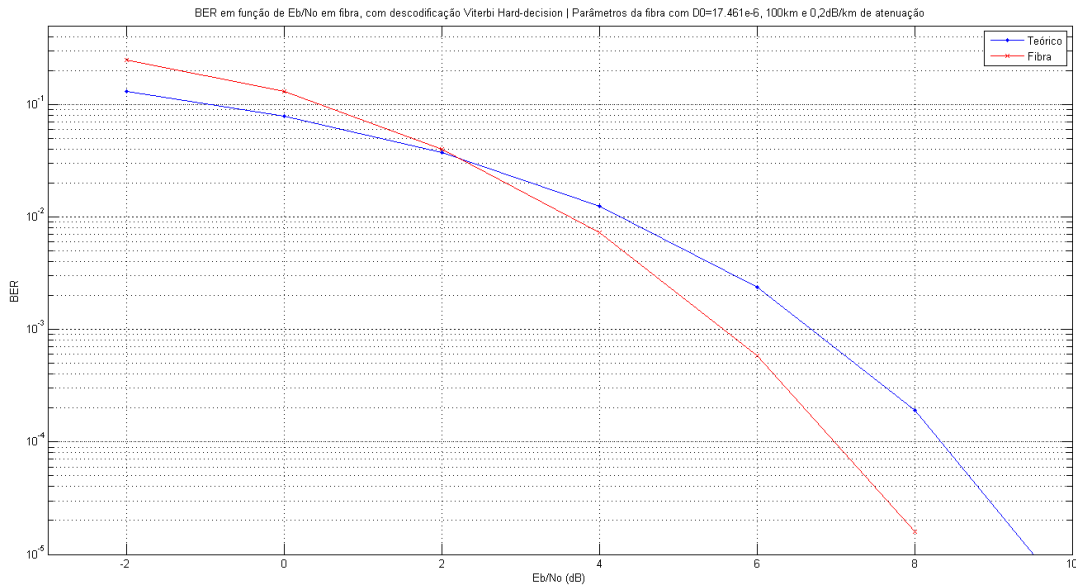


Figura 63 - Decodificação de Viterbi após fibra

7.3.1. Algoritmo BCJR

Este algoritmo é a base da codificação turbo e da igualização turbo, pois é este que é utilizado nos decodificadores de cada um destes princípios. A metodologia utilizada para as simulações foi idêntica à utilizada em Viterbi. Primeiro realiza-se uma simulação para um canal de ruído Gaussiano e por fim para a fibra, mantendo sempre a modulação BPSK. Os sistemas adotados, para o canal AWGN e fibra, para a realização das simulações são apresentados na Figura 64 e Figura 65 respetivamente.

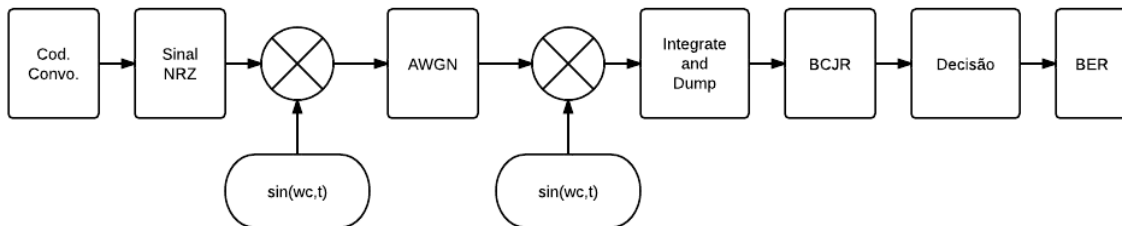


Figura 64 - Circuito para algoritmo BCJR com canal AWGN

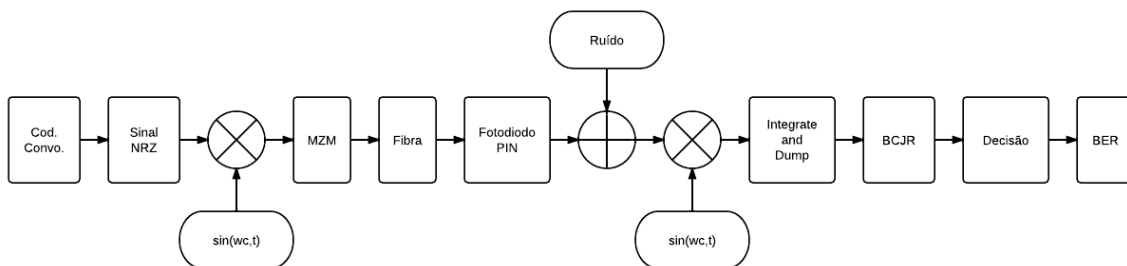


Figura 65 - Circuito para algoritmo BCJR com fibra

AWGN

Após a passagem do sinal pelo canal AWGN, a seguinte curva de erros foi obtida para o decodificador BCJR.

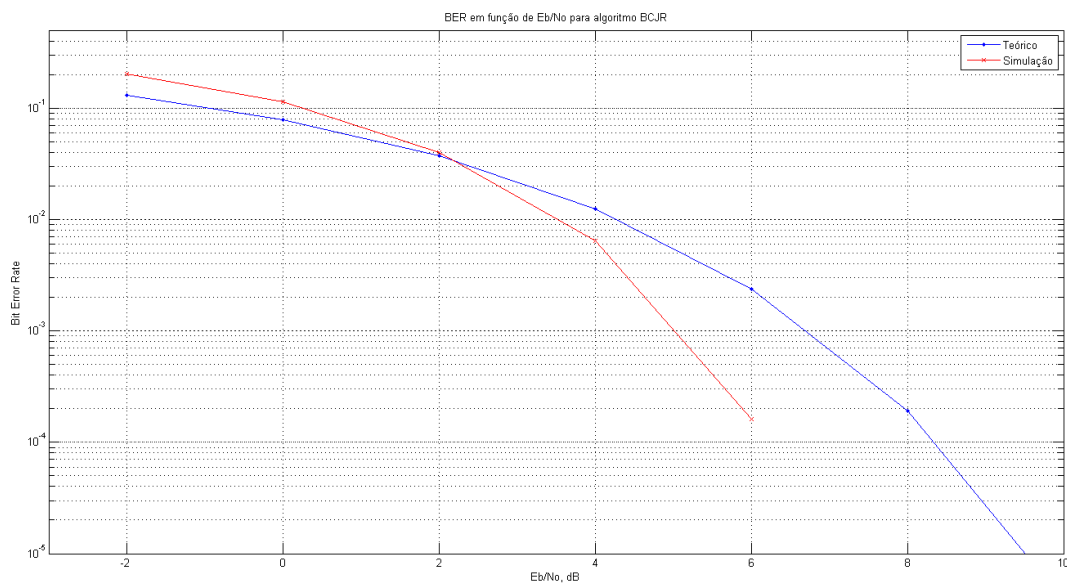


Figura 66 - Decodificação BCJR após canal AWGN

O resultado obtido nesta simulação não está coerente com o pretendido, pois a curva situa-se acima da curva de Viterbi para um canal AWGN, o que não deveria acontecer.

Fibra

Novamente tiveram que ser inseridos os elementos constituintes do sistema (MZM, fibra e PIN) para ser possível a simulação sobre o canal fibra.

A simulação gerou a curva de erros apresentada pela Figura 67. Para esta situação verifica-se que a curva prática está abaixo do pretendido, o que pode estar relacionado com a má implementação do algoritmo BCJR.

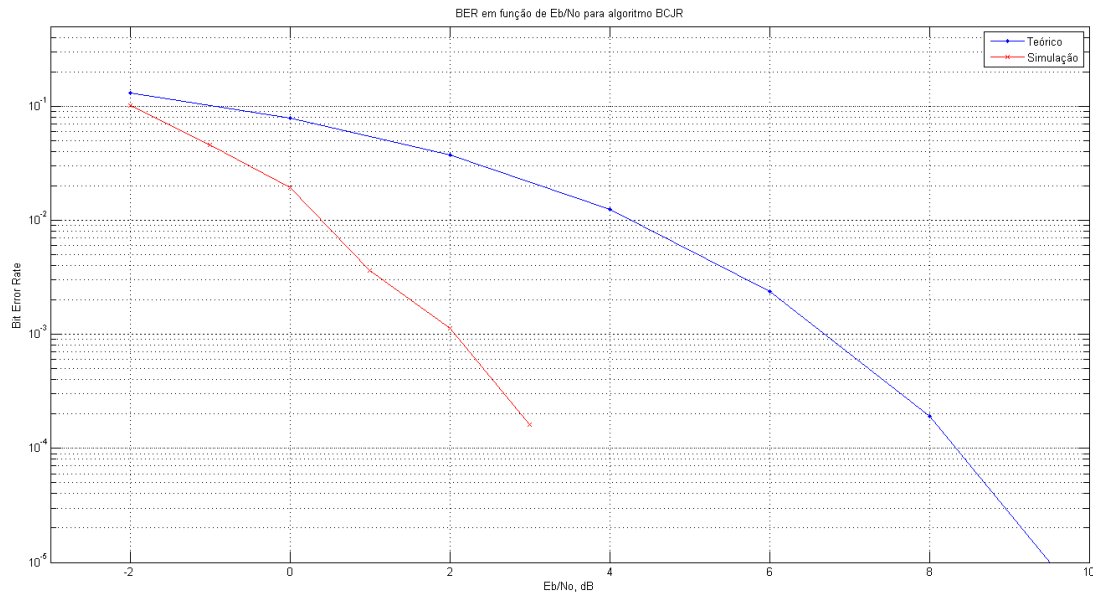


Figura 67 - Probabilidade de erro em função de Eb/No para um decodificador BCJR

7.4. O processamento turbo

Nesta fase é necessário recorrer a um codificador RSC, onde o escolhido foi o da Figura 34, que tem como matriz geradora

$$\bullet \quad G = \left[1 \quad \frac{1+D^2}{1+D+D^2} \right]$$

Após a escolha deste cria-se o módulo responsável pela codificação e que é constituído por dois codificadores RSC deste tipo em paralelo. Sabendo que cada codificador tem taxa associada de 1/2, após a colocação destes dois codificadores em paralelo, o módulo codificador ficará com uma taxa de 1/3. Este módulo é o mostrado pela Figura 43. O facto de a taxa resultante desta concatenação em paralelo ser 1/3 leva à utilização de códigos perfurados [19] para a obtenção de uma taxa pretendida de 1/2.

Em relação a decodificação do sinal, sabe-se que é realizada de forma iterativa entre dois decodificadores. O sistema que representa este módulo de decodificação, introduzido anteriormente, pode ser visto na Figura 44.

Estas simulações vão ter em conta a passagem do sinal pelo canal AWGN ou fibra e o número de iterações realizadas pelo decodificador turbo.

Nesta fase é também necessário de um multiplexador e um demultiplexador para ser possível juntar e separar, respetivamente, o sinal. O sistema correspondente é a baixo apresentado.

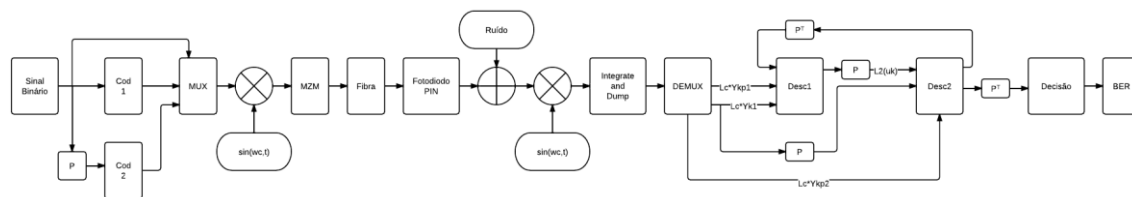


Figura 68 - Sistema para a decodificação turbo iterativa num sistema de comunicações óticas

AWGN

As simulações para um canal AWGN com 1 e 2 iterações são apresentadas na Figura 69 e Figura 70 respetivamente. Estas simulações foram realizadas com o algoritmo BCJR desenvolvido. Os resultados obtidos não são os esperados e tal como nas simulações anteriores (BCJR), o problema aqui pode ser o mesmo.

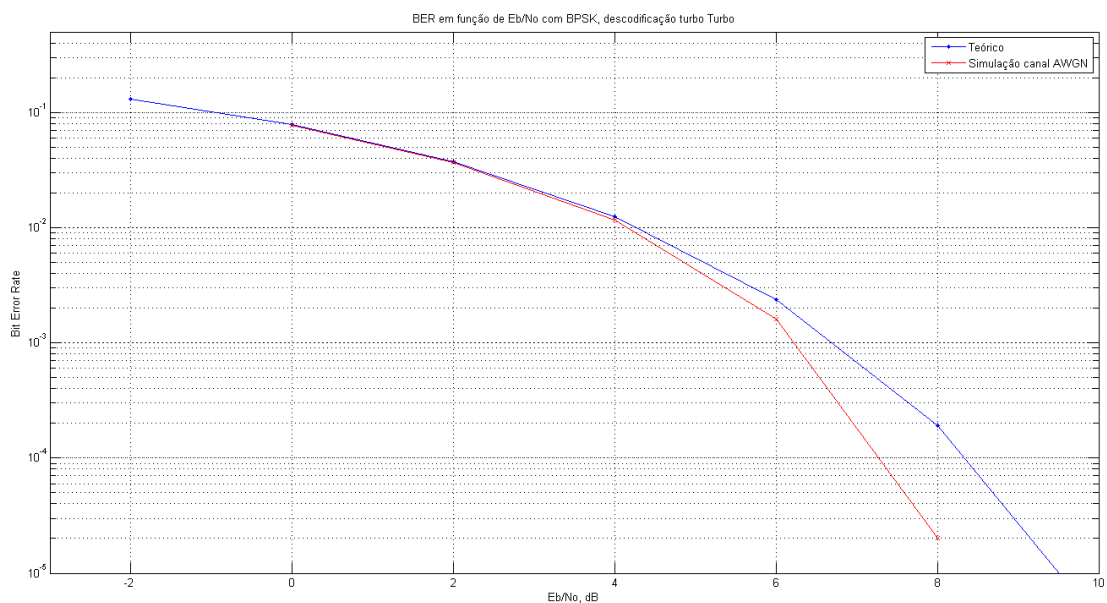


Figura 69 - BER para 1 iteração turbo num canal AWGN

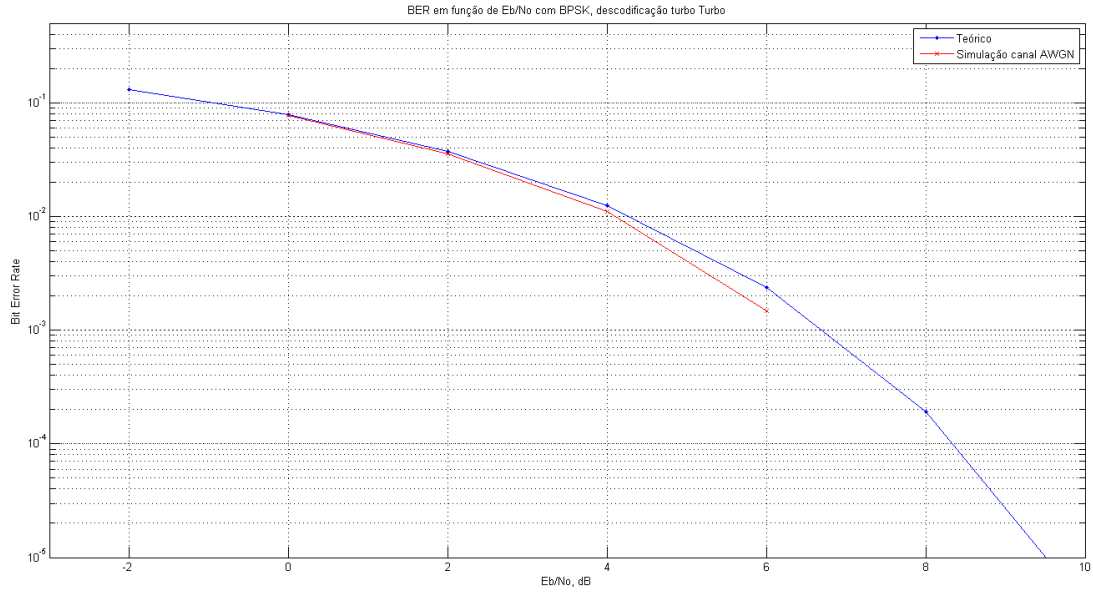


Figura 70 - BER para 2 iterações turbo num canal AWGN

Fibra

No que diz respeito à passagem do sinal pela fibra, os resultados obtidos podem ser observados na Figura 71 para 1 e 2 iterações. Estes resultados estão dentro do pretendido e verifica-se até uma boa progressão do sistema quando se realizam duas iterações. Nestas simulações foram realizadas com um algoritmo Log-MAP desenvolvido, e talvez, por isso se assumam melhores que as do canal AWGN.

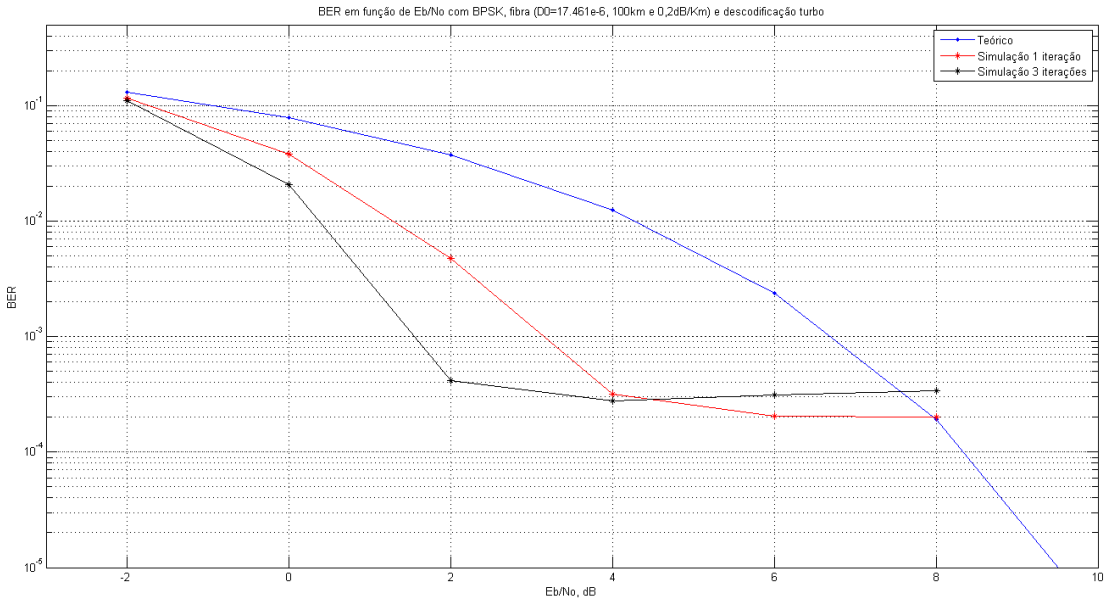


Figura 71 - BER para 1 e 2 iterações turbo em fibra

7.5. A igualização turbo

Na projeção do sistema de igualização turbo foi colocado apenas um codificador RSC, com matriz geradora

$$G = \begin{bmatrix} 1 & \frac{1+D^2}{1+D+D^2} \end{bmatrix}$$

devido ao processo de igualização turbo, no final, ser realizado em série. Desta forma o circuito equivalente para a igualização será o apresentado na Figura 54. Para uma melhor visão de todo o sistema segue a Figura 72, onde agora já só se tem como foco a fibra.

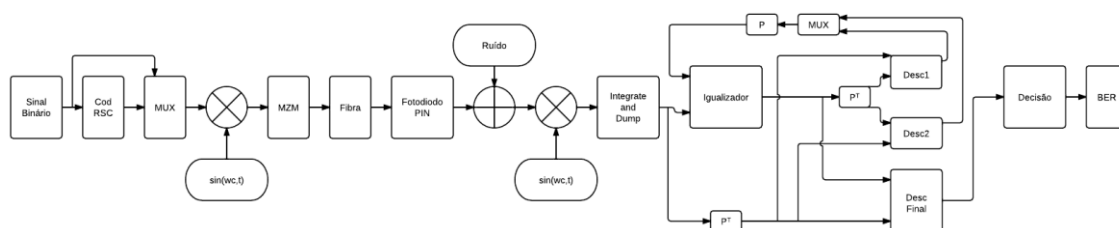


Figura 72 - Circuito para igualização turbo

A igualização turbo, tal como o processamento turbo, é realizada de forma iterativa. Neste trabalho o processo descreve-se da seguinte forma:

- O codificador RSC transforma os bits de entrada em palavras de código, que são entrelaçadas, convertidas a um sinal NRZ por forma a facilitar a posterior modulação BPSK. Após o sinal modulado, este é tratado por um MZM por forma a obter uma forma ótica e ser possível a sua inserção na fibra ótica. Depois disto é recebido por um fotodiodo PIN que converte o sinal ótico em sinal elétrico para ser realizada a desmodulação BPSK. De seguida o sinal passa por um filtro *Integrate and Dump*, que tem como função passar o sinal do domínio dos tempos para o domínio discreto;
 - Nesta fase o sinal já estará na entrada do igualizador turbo e dar-se-á a inicialização do processo de igualização;
 - Na primeira iteração entram as amostras de sinal, provenientes do canal, e respetivas probabilidades a priori (nulas) no igualizador. Este igualizador baseia-se numa treliça dinâmica que representa o canal de transmissão e trata símbolo a símbolo por forma a obter uma probabilidade a posteriori melhor à cerca dos seus símbolos de entrada. Esta ação tem como objetivo a redução da ISI;
 - Se não for a última iteração esta probabilidade à posteriori do igualizador é transformada numa probabilidade extrínseca e cuja será desentrelaçada e entregue a dois decodificadores como sua probabilidade a priori;
 - Estes dois decodificadores, que tendo em conta a treliça do codificador RSC, tratam símbolo a símbolo recebido, desentrelaçado, do canal e converte esta informação em informação a posteriori;
 - De seguida esta informação a posteriori é transformada em informação extrínseca, entrelaçada e enviada para o igualizador de novo;

- Caso seja a última iteração a informação a posteriori do igualizador é inserida no decodificador final e são obtidas as LLR's que serão utilizadas para a decisão rígida final e assim será possível estimar o sinal de entrada.

Em todo o processo de igualização turbo, todos os decodificadores utilizam o algoritmo Log-MAP, que como já foi referido tem um melhor desempenho que o BCJR e obtém valores muito próximos deste último.

Neste capítulo foi estudado o efeito do tamanho do entrelaçador no processo de igualização. Os resultados obtidos são apresentados na Figura 73.

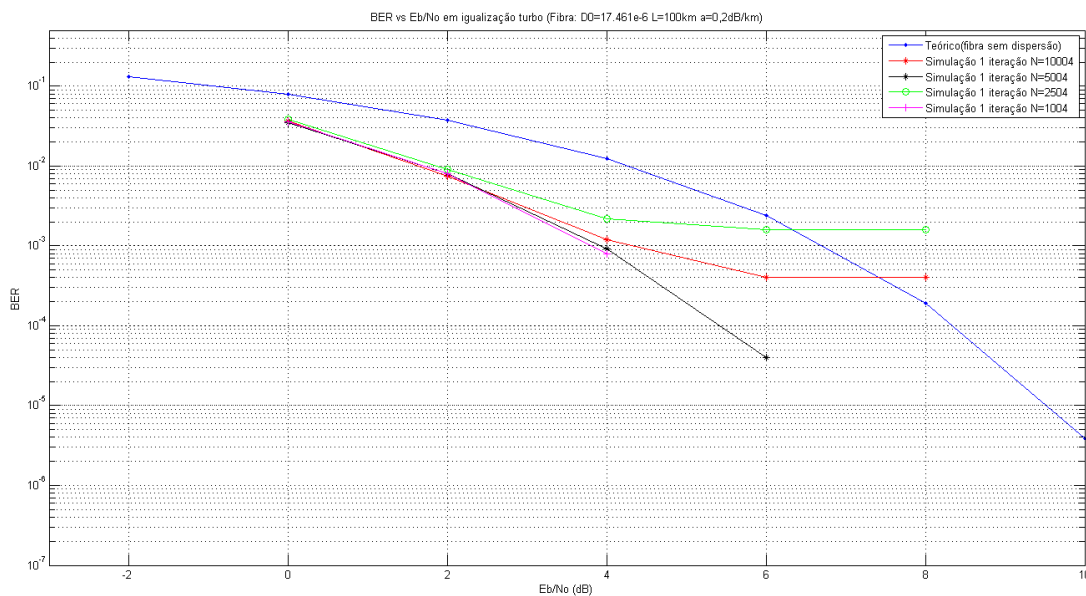


Figura 73 - BER em função de E_b/N_0 para 1 iteração de igualização turbo com vários tamanhos de entrelaçador

Nestas simulações para vários tamanhos de entrelaçadores, foram transmitidos sempre 50000bits, apenas ajustando o número de simulações para compensar o número de bits transmitidos em cada simulação. Nestas simulações, o melhor caso obtido, apesar de todos se situarem muito próximos, foi com o entrelaçador de tamanho $N=1004$. Talvez se o tamanho do entrelaçador fosse aumentado para uma ordem dos 10^5 as simulações obtivessem melhores resultados. Este caso não foi tratado, pois a complexidade associada torna-se impossível e ocorrem falhas de memória no computador.

Agora a igualização turbo utilizará o entrelaçador com $N=10004$, apesar de não ter obtido os resultados pretendidos na simulação anterior, mas como é o de maior tamanho será o escolhido. Serão realizadas várias simulações para comparar as taxas de erros obtidas para várias iterações com um milhão de bits transmitidos. Os resultados obtidos são apresentados a seguir.

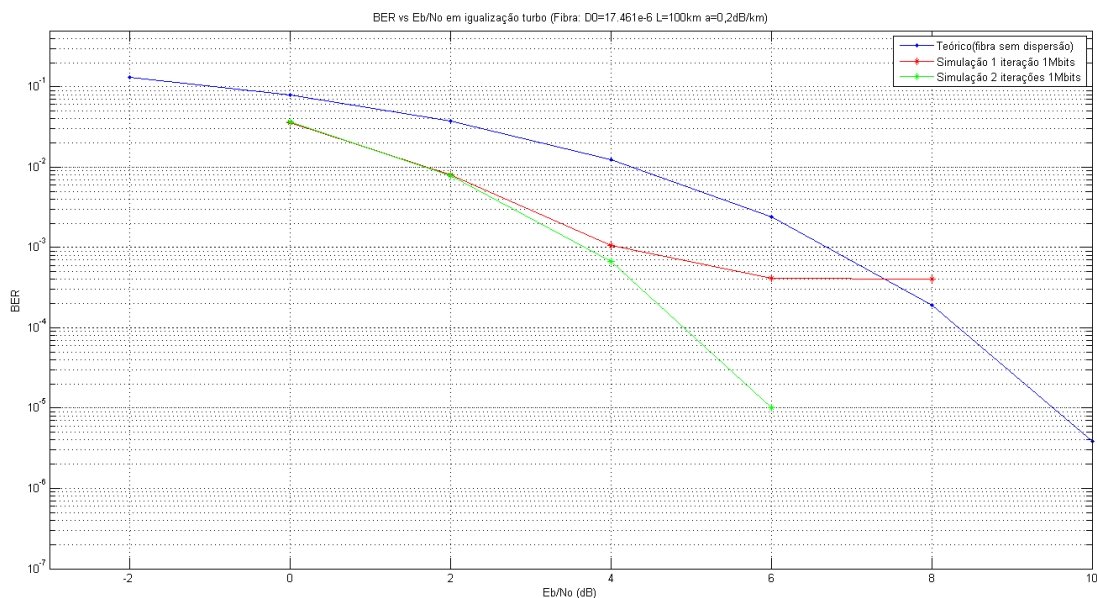


Figura 74 - Igualização turbo com 1 e 2 iterações para 1Mbits transmitidos com matriz Geradora $G=[1\ 0\ 0;0\ 0\ 0]$

Após esta simulação denota-se que para uma iteração o sistema obteve um BER *floor* indesejado, mas para duas iterações já se comportou melhor.

Após estes resultados, decidiu-se aumentar o tamanho do polinómio gerador referente ao canal onde se obteve uma matriz geradora $G=[1\ 0\ 0\ 0;0\ 0\ 0\ 0]$ e os resultados obtidos seguem-se na Figura 75. Estes resultados sofrem uma melhoria em relação à simulação anterior.

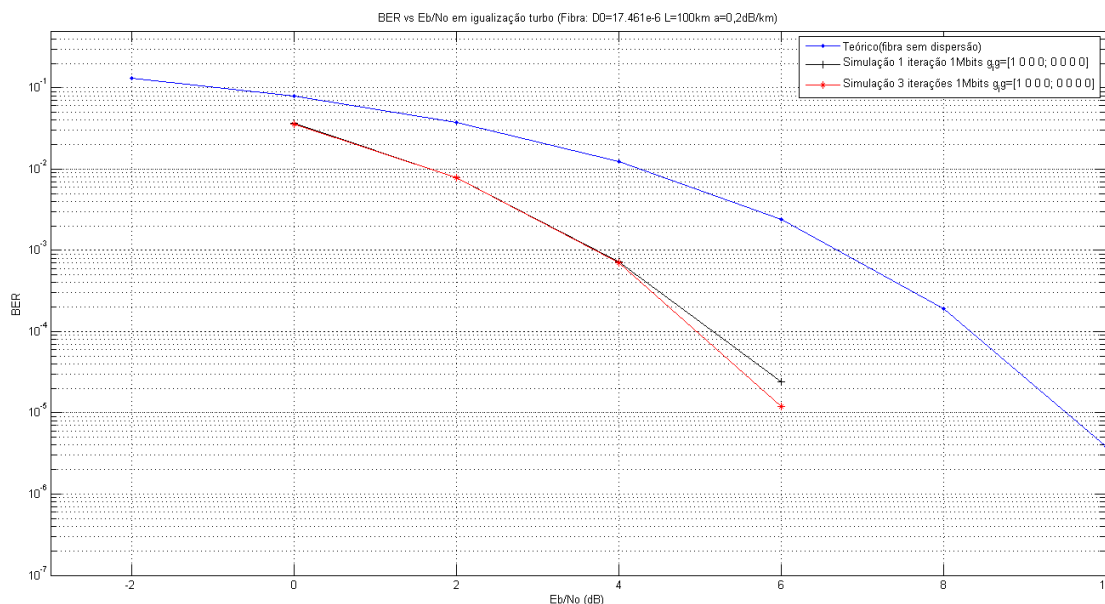


Figura 75 - Igualização turbo com 1 e 3 iterações para 1Mbits transmitidos com matriz Geradora $G=[1\ 0\ 0\ 0;0\ 0\ 0\ 0]$

7.6. Resumo e conclusões

Neste capítulo é descrito o procedimento efetuado ao longo do trabalho, por forma a iniciar um estudo sobre os aspetos mais simples (modulação e codificação) até ao objetivo final, a igualização turbo.

Nas primeiras simulações, sobre BPSK, nota-se que as curvas de simulação obtidas se aproximam bastante da curva teórica. A curva da fibra situa-se um pouco acima da curva teórica devido ao efeito de dispersão adicionado pela fibra.

Prosseguiu-se o trabalho e o próximo passo foi a junção de codificação à modulação, onde os resultados foram obtidos para os algoritmos de descodificação de Viterbi e BCJR. Como o Viterbi utilizado tem como método de decisão *hard-decision*, se fossem realizadas simulações com *soft-decision*, obter-se-ia um ganho entre 2 a 3dB em relação ao *hard-decision* utilizado [35]. O exemplo de uma curva esperada para Viterbi *hard-decision* foi conseguida através da ferramenta *Bertool* do *Matlab* e é apresentada na Figura 76. Como era de esperar em Viterbi a curva da fibra situa-se um pouco acima da curva do canal com ruído Gaussiano devido ao efeito de dispersão na fibra. Em relação à descodificação através do algoritmo BCJR concluiu-se que não está a funcionar como pretendido. No caso da fibra, este deveria estar a obter taxas de erro mais baixas que Viterbi e tal facto confirma-se através dos resultados gráficos. Mesmo assim, e comparando com o processamento turbo, verifica-se que na fibra a taxa de erros obtida está abaixo, no caso de 1 iteração turbo, o que não é expectável. Na descodificação BCJR para o canal AWGN nota-se que a curva está pior que para a descodificação de Viterbi, o que pode dever-se a qualquer problema no código realizado para o algoritmo BCJR.

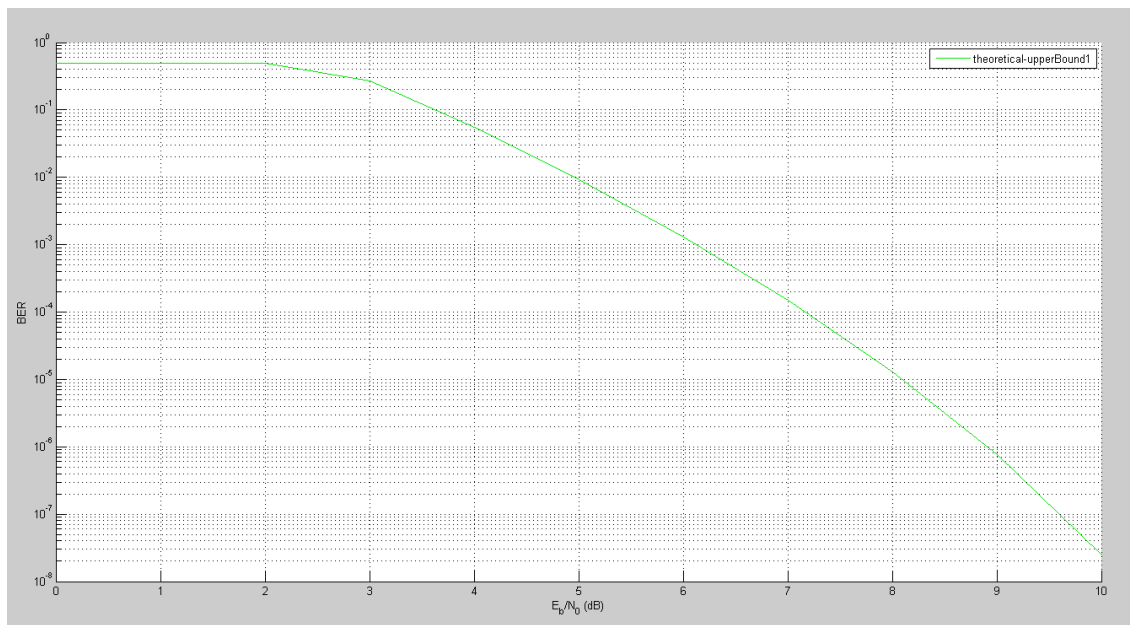


Figura 76 - BER teórico para Viterbi *hard-decision* em canal AWGN

Para processamento turbo num canal AWGN, mais uma vez, os gráficos não estão coerentes com o que se pretendia, de facto existe a sensação de que a descodificação não está a funcionar corretamente. Isto pode dever-se a mais uma vez qualquer problema com o algoritmo BCJR utilizado neste caso. No caso da fibra existe até uma boa aproximação do

esperado para uma iteração e para duas iterações verifica-se uma melhoria na curva, o que aumentado o número de iterações pode melhorar.

Na igualização turbo verificou-se para vários comprimentos de entrelaçadores, qual o que obtém melhor desempenho, isto é, fez-se uma comparação do BER obtido para uma iteração e uma dada relação E_b/N_0 , entre as várias profundidades de entrelaçadores. No que diz respeito ao gráfico da Figura 74, estudou-se o aumento do número de iterações com uma transmissão de um milhão de bits. Nestes resultados vê-se que o BER está aceitável para o número de bits transmitidos e que com o aumento do número de iterações, existe uma prossecução favorável do BER em função do E_b/N_0 . Verifica-se que a curva de duas iterações para uma relação sinal ruído de 8dB já consegue que não existam erros na transmissão. Devido à complexidade temporal associada a cada simulação, não foi possível realizar mais iterações para termo de comparação. Após o aumento dos polinómios geradores referentes ao canal verificou-se uma melhoria para 1 iteração.

Numa visão geral, conseguiu-se obter um circuito igualizador que funciona razoavelmente bem, tendo em conta que para já só se conseguiram estes resultados. Existiram algumas dificuldades na percepção de como seria integrado todo o circuito igualizador, até que se conseguiu a versão apresentada nesta tese e estes respetivos valores. De uma forma geral pode verificar-se que existe uma redução significativa da dispersão, pois, a curva obtida na Figura 75 situa-se abaixo da curva obtida para fibra, apenas com modulação, e com dispersão presente na Figura 59. Através de outros trabalhos já realizados é possível verificar que este sistema se aproxima bem dos mesmos. Este facto pode ver-se na Figura 77.

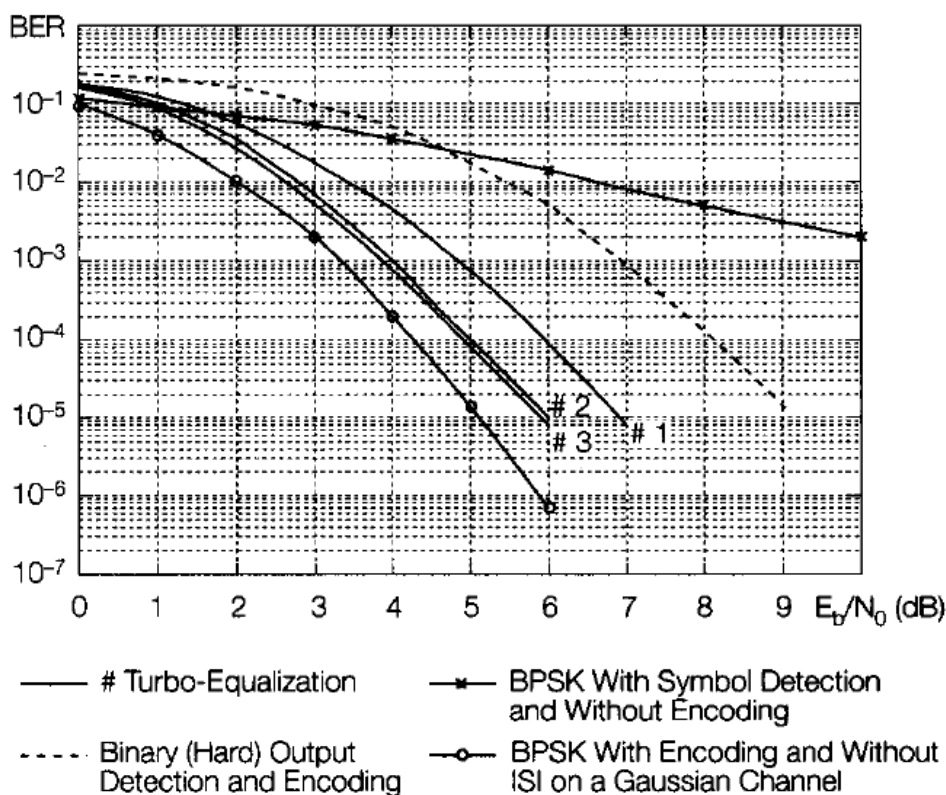


Figura 77 - Igualização turbo para 1, 2 e 3 iterações[28]

7.7. Trabalho futuro

Numa próxima abordagem deste tema, seria possível o estudo de:

- Como se processa a troca de informações extrínsecas entre equalizador e decodificador, sem ser necessário a inserção dos símbolos de informação proveniente do canal de transmissão. Ou seja, neste caso o decodificador seria alimentado apenas com a informação extrínseca da saída do igualizador e na sua entrada de informação a priori seria alimentado com informação nula (vetor de zeros).
- A possibilidade de existir apenas igualizador e decodificador final, sem que seja necessário recorrer aos decodificadores intermédios, que se devem às dimensões de saída do decodificador final serem diferentes das dimensões de entrada do igualizador.

Referências

- [1] I. Djordjevic, W. Ryan, and B. Vasic, *Coding for optical channels*, 2010.
- [2] A. Aloisio, F. Ameli, A. D'Amico, R. Giordano, G. Giovanetti, and V. Izzo, "Performance Analysis of a DWDM Optical Transmission System," *Nuclear Science, IEEE Transactions on*, vol. 59, pp. 251-255, 2012.
- [3] H. M. d. C. F. Salgado. *Apontamentos da Disciplina de Comunicações Ópticas*. Available: <http://paginas.fe.up.pt/~hsalgado/co/>
- [4] P. D. C. C. Freudenrich, "How Fiber Optics Work," *HowStuffWorks.com*, 06 March 2001.
- [5] R. Salem, M. A. Foster, A. C. Turner, D. F. Geraghty, M. Lipson, and A. L. Gaeta, "Signal regeneration using low-power four-wave mixing on silicon chip," *Nat Photon*, vol. 2, pp. 35-38, 2008.
- [6] L.-N. L. A. R. H. F.-W. S. M. Eroz, "Application and Standardization of Turbo Codes in Third-Generation High-Speed Wireless Data Services," *Vehicular Technology, IEEE Transactions*, vol. 49, November 2000.
- [7] C. T. P. C. B. Honary, "Turbo Code Application to Image Transmissão," pp. 17/1 - 17/6 1999.
- [8] MATLAB. Available: <http://www.mathworks.com/products/matlab/>
- [9] G. Keiser, *Optical fiber communications*: McGraw-Hill, 1983.
- [10] H. F. P. d. S. Reboredo, "Processamento de Sinal Turbo em Sistemas de Comunicações Ópticas," 2009.
- [11] B. Xu and M. Brandt-Pearce, "Modified Volterra Series Transfer Function Method and Applications to Fiber-Optic Communications."
- [12] S. R. Abdollahi, H. S. Al-Raweshidy, S. M. Fakhraie, and R. Nilavalan, "Digital Radio over Fibre for Future Broadband Wireless Access Network Solution," presented at the Proceedings of the 2010 6th International Conference on Wireless and Mobile Communications, 2010.
- [13] D. Opatich, "Radio over Fiber Technology for Wireless Access," pp. 202-227, 1995.
- [14] A. Ng'oma, "Radio-over-Fibre Technology for Broadband Wireless Communication Systems," Technische Universiteit Eindhoven, 2005.

- [15] S. A. Abrantes. *Modulações Digitais* 2. Available: http://paginas.fe.up.pt/~sam/Tele2/apontamentos/Modul_2.pdf
- [16] P. M. Bonfim. *Técnicas de Modulação*. Available: <http://www.eletr.ufpr.br/marlio/te241/aula4.pdf>
- [17] V. M. d. O. Fialho. (2011). *Apontamentos sobre Modulações Digitais*. Available: <http://www.deetc.isel.ipl.pt/sistematele/cm/Bibliografia/MaterialDeApoio/ModulacoesDigitais.pdf>
- [18] S. A. Abrantes. *Modulações digitais lineares com detecção coerente*. Available: http://paginas.fe.up.pt/~sam/Tele2/apontamentos/Modul_3_1.pdf
- [19] S. A. Abrantes, *Códigos Correctores de Erros em Comunicações Digitais* vol. 1ª edição, 2010.
- [20] S. A. Abrantes. *Códigos de blocos*. Available: http://joaopizani.hopto.org/graduacao/disciplinas/mtm5512/Trabalho%20Final/cod_blocos_web.pdf
- [21] S. A. Abrantes. *Códigos Convolucionais*. Available: <http://paginas.fe.up.pt/~sam/TL/apontamentos/apontamentos.html>
- [22] S. A. Abrantes. (2004). *Do algoritmo BCJR à descodificação turbo*. Available: <http://paginas.fe.up.pt/~sam/textos/De%20BCJR%20a%20turbo.pdf17.pdf>
- [23] S. A. Abrantes. (2005). *Turbo everywhere*. Available: <http://paginas.fe.up.pt/~sam/textos/turbo%20everywhere.pdf>
- [24] M. Taskaldiran, R. C. S. Morling, and I. Kale, "The Modified Max-Log-MAP Turbo Decoding Algorithm by Extrinsic Information Scaling for Wireless Applications."
- [25] A. F. d. Santos and D. A. Guimarães, "Introdução à Equalização Turbo."
- [26] M. C. V. B. D. Woerner, "Variable Latency Turbo Codes for Wireless Multimedia Applications," 1997.
- [27] B. L. Yeap, T. H. Liew, J. Hamorsky, and L. Hanzo, "Comparative study of turbo equalization schemes using convolutional, convolutional turbo, and block-turbo codes," *Trans. Wireless. Comm.*, vol. 1, pp. 266-273, 2002.
- [28] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, pp. 507-511, 1995.
- [29] R. Koetter, A. C. Singer, and M. Tuchler, "Turbo equalization," *Signal Processing Magazine, IEEE*, vol. 21, pp. 67-80, 2004.
- [30] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, 1993, pp. 1064-1070 vol.2.
- [31] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *Communications, IEEE Transactions on*, vol. 50, pp. 754-767, 2002.
- [32] L. Chun-Ting, P. Wei-Ren, P. Peng-Chun, J. Chen, P. Cheng-Feng, C. Bi-Shiou, and S. Chi, "Simultaneous Generation of Baseband and Radio Signals Using Only One Single-Electrode Mach–Zehnder Modulator With Enhanced Linearity," *Photonics Technology Letters, IEEE*, vol. 18, pp. 2481-2483, 2006.

- [33] J. Mora, A. Ortigosa-Blanch, D. Pastor, and J. Capmany, "Tunable Microwave Photonic Filter Free from Carrier Suppression Effect and Baseband Response not Requiring Single Sideband Modulation," in *Optical Communications, 2006. ECOC 2006. European Conference on*, 2006, pp. 1-2.
- [34] J. Mora, A. Ortigosa-Blanch, D. Pastor, and J. Capmany, "Tunable microwave photonic filter free from baseband and carrier suppression effect not requiring single sideband modulation using a Mach-Zenhder configuration," *Opt. Express*, vol. 14, pp. 7960-7965, 2006.
- [35] F. I. Alajaji and N. C. Phamdo, "Soft-decision COVQ for Rayleigh-fading channels," *Communications Letters, IEEE*, vol. 2, pp. 162-164, 1998.