

# **Um Esquema de Equalização Turbo Aplicando Decodificação Turbo de Códigos Produto de Paridade Simples Multidimensionais**

ANDRÉ FONSECA DOS SANTOS

Dissertação apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: PROF. DR. DAYAN ADIONEL GUIMARÃES

**Santa Rita do Sapucaí**  
**07/03/2005**

Este trabalho foi realizado com o apoio financeiro da  
Financiadora de Estudos e Projetos - FINEP



Dissertação defendida e aprovada em 07/03/2005, pela comissão julgadora:

---

Prof. Dr. Dayan Adionel Guimarães/INATEL

---

Prof. Dr. Sandro Adriano Fasolo/INATEL

---

Prof. Dra. Maria das Dores dos Santos Miranda/Mackenzie

---

**Prof. Dr. Adonias Costa da Silveira**  
**Coordenador do Curso de Mestrado**

“Algo só é impossível até que alguém duvide e acabe provando o contrário”.

(Albert Einstein)

# Agradecimentos

Agradeço a Deus por guiar o meu caminho. Ao meu pai, minha mãe e minha irmã, também merecedores desta conquista. À minha noiva Paula, pelo carinho e compreensão. Ao Prof. Dayan, pela brilhante orientação e amizade. Ao amigo Angelo, pela valiosa ajuda neste trabalho. A todos os professores e alunos do mestrado do INATEL, pela amizade e incentivo. Ao apoio financeiro proporcionado pelo convênio n° 22.02.0431.00, celebrado entre o Inatel, a Linear Equipamentos Eletrônicos S/A e a FINEP (Financiadora de Estudos e Projetos).

# Índice

<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Tabelas</b>	<b>vii</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>viii</b>
<b>Lista de Símbolos</b>	<b>x</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Equalização Turbo</b>	<b>7</b>
2.1 Modelo Discreto do Canal . . . . .	8
2.2 Princípios da Equalização Turbo . . . . .	11
2.3 Algoritmo BCJR . . . . .	18
2.3.1 Algoritmo BCJR Aplicado ao Bloco Equalizador . . . . .	21
2.3.2 Algoritmo BCJR Aplicado ao Bloco Decodificador . . . . .	22
2.4 Critérios de Parada . . . . .	23
2.4.1 Observando Decisões Abruptas . . . . .	24
2.4.2 Função Risco . . . . .	24
2.5 Sumário e Comentários Adicionais . . . . .	25
<b>3 Equalização Turbo Utilizando Códigos de Paridade Simples Multidimensionais</b>	<b>26</b>
3.1 Códigos Produto de Paridade Simples . . . . .	26

---

3.1.1	Formação dos Códigos Produto . . . . .	27
3.1.2	Decodificação Turbo dos Códigos Produto . . . . .	30
3.2	Equalização Turbo com Decodificação Turbo Utilizando Códigos SPC-TPC . . . . .	35
3.2.1	Equalização Turbo Utilizando Códigos SPC-TPC . . . . .	37
3.2.2	A Proposta de Equalização Turbo com Códigos SPC-TPC	40
3.3	Sumário e Comentários Adicionais . . . . .	44
<b>4</b>	<b>Resultados de Simulação</b>	<b>46</b>
4.1	Desempenho do Equalizador Turbo em Relação ao Número de I- terações . . . . .	48
4.2	Desempenho do Equalizador Turbo Utilizando o Código $C_1$ . . . . .	49
4.3	Desempenho do Equalizador Turbo Utilizando o Código $C_2$ . . . . .	52
4.4	Desempenho do Equalizador Turbo Utilizando o Código $C_3$ . . . . .	55
4.5	Sumário e Comentários Adicionais . . . . .	57
<b>5</b>	<b>Itens Conclusivos</b>	<b>58</b>
5.1	Sumário Geral e Conclusões . . . . .	58
5.2	Principais Contribuições . . . . .	60
5.3	Oportunidades para Futuras Investigações . . . . .	61
5.4	Publicações do Autor . . . . .	62
	<b>Referências Bibliográficas</b>	<b>63</b>

# Lista de Figuras

1.1	Canal de comunicação com múltiplos percursos. . . . .	2
2.1	Sistema de transmissão . . . . .	8
2.2	Modelo discreto do canal com IIS . . . . .	9
2.3	Codificador convolucional de taxa 1/2 . . . . .	9
2.4	Treliça que representa as transições de estados para um codificador convolucional de comprimento 3 ou de um canal com dois múltiplos percursos, onde $S_0 = \{-1, -1\}$ , $S_1 = \{+1, -1\}$ , $S_2 = \{-1, +1\}$ e $S_3 = \{+1, +1\}$ . Para o codificador convolucional o valor $-1$ é trocado por 0 nos valores dos estados. . . . .	10
2.5	Sistema de transmissão com canal sendo o codificador interno . . . . .	11
2.6	Sistema de recepção convencional com decisões tomadas no equalizador e no decodificador de maneira separada. . . . .	12
2.7	Equalizador MAP ( <i>Maximo a Posteriori</i> ). . . . .	13
2.8	Decodificador MAP ( <i>Maximo a Posteriori</i> ). . . . .	14
2.9	Equalizador Turbo. . . . .	15
2.10	Intepertação do Equalizador Turbo ataravés de módulos. . . . .	17
3.1	Formação do código $(6, 5, 2)^3$ . Os bits em azul representam a informação, os bits amarelos as paridades das colunas, os bits vermelhos as paridades das linhas e os bits verdes as paridades da profundidade. . . . .	30



3.2	Sistema de comunicação utilizando decodificação turbo de códigos SPC. . . . .	31
3.3	Entradas e saídas de um decodificador turbo. . . . .	33
3.4	Troca de informações em um decodificador turbo de um código SPC-TPC de três dimensões (adaptada de [Sam04]). . . . .	33
3.5	Sistema de comunicação utilizando equalização turbo. . . . .	37
3.6	Equalização turbo conjunta com decodificação turbo. . . . .	37
3.7	Equalizador Turbo utilizando apenas uma dimensão do código produto no bloco de decodificação. . . . .	40
3.8	Histogramas da evolução da LLR dos bits de informação na saída do bloco decodificador entre a iteração 1 e a iteração 9. . . . .	45
4.1	coeficientes do canal 1 e sua resposta em frequência (magnitude) .	47
4.2	coeficientes do canal 2 e sua resposta em frequência (magnitude) .	47
4.3	Convergência do equalizador turbo utilizando decodificador turbo SPC-TPC. . . . .	49
4.4	Convergência do equalizador turbo utilizando o bloco decodificador de apenas uma dimensão. Estão traçadas de cima para baixo as curvas relativas às iterações 1 à 15. . . . .	50
4.5	Desempenho do equalizador turbo com decodificação parcial SPC-TPC SPC-TPC utilizando o código $C_1$ no canal $h_1$ . . . . .	51
4.6	Desempenho do equalizador turbo com decodificação parcial SPC-TPC SPC-TPC utilizando o código $C_1$ no canal $h_2$ . . . . .	52
4.7	Desempenho do equalizador turbo com decodificação parcial SPC-TPC SPC-TPC utilizando o código $C_2$ no canal $h_1$ . . . . .	53
4.8	Desempenho do equalizador turbo com decodificação parcial SPC-TPC SPC-TPC utilizando o código $C_2$ no canal $h_2$ . . . . .	54
4.9	Desempenho do equalizador turbo com decodificação parcial SPC-TPC SPC-TPC utilizando o código $C_3$ no canal $h_1$ . . . . .	56

# Lista de Tabelas

3.1	Algoritmo de decodificação turbo para códigos SPC. . . . .	35
3.2	Algoritmo da nova configuração proposta de equalização e decodificação turbo. . . . .	42
4.1	Entrelaçadores utilizados nas simulações. . . . .	48
4.2	Ganhos obtidos pelo equalizador turbo proposto em relação aos equalizadores turbo analisados em [Yea02]. A coluna da esquerda apresenta os códigos utilizados em tais equalizadores turbo. Na segunda coluna estão os ganhos obtidos pelo equalizador turbo com decodificação parcial em relação aos equalizadores turbo de taxa 3/4 para uma BER de $10^{-5}$ . . . . .	54
4.3	Ganhos obtidos pelo equalizador turbo proposto em relação aos equalizadores turbo analisados em [Yea02]. A coluna da esquerda apresenta os códigos utilizados nestes equalizadores turbo. Na segunda coluna estão os ganhos obtidos pelo equalizador turbo com decodificação parcial em relação aos equalizadores turbo de [Yea02] de taxa 5/6 para uma BER de $10^{-5}$ . . . . .	56

# Lista de Abreviaturas e Siglas

<b>AWGN</b>	<i>Additive White Gaussian Noise</i> - ruído branco aditivo com função densidade de probabilidade Gaussiana
<b>BER</b>	<i>Bit Error Rate</i> - taxa de erro de bit
<b>BPSK</b>	<i>Binary Phase Shift Keying</i> - modulação binária por chaveamento de fase
<b>BCH</b>	<i>Bose-Chaudhuri-Hocquenghem codes</i> - Códigos Bose-Chaudhuri-Hocquenghem
<b>BCJR</b>	<i>Bahl Cocke Jelinek Raviv</i> - algoritmo que implementa o critério MAP criado por Bahl, Cocke, Jelinek e Raviv
<b>DFE</b>	<i>Decision Feedback Equalizer</i> - equalizador com malha de realimentação
<b>IIS</b>	Interferência Intersimbólica
<b>LMS</b>	<i>Least Mean Square</i> - mínimo médio quadrático
<b>LLR</b>	<i>Log-likelihood Ratio</i> - razão de log-verossimilhança
<b>MAP</b>	<i>Maximum a Posteriori</i> - algoritmo Máximo a Posteriori
<b>MLSE</b>	<i>Maximum Likelihood Sequence Estimation</i> - estimação da sequência de máxima verossimilhança
<b>MMSE</b>	<i>Minimum Mean Square Error</i> - mínimo erro médio quadrático
<b>SISO</b>	<i>Soft-Input Soft-Output</i> - algoritmo com entrada e saída suave
<b>SOVA</b>	<i>Soft Output Viterbi Algorithm</i> - algoritmo de Viterbi com saída suave
<b>SPC</b>	<i>Single Parity Check codes</i> - códigos de cheque de paridade simples
<b>SPC-PC</b>	<i>Single Parity Check codes Product Codes</i> - códigos produto de cheque de paridade simples

**SPC-TPC** *Single Parity Check - Turbo Product Codes* - códigos produto de cheque de paridade simples com decodificação turbo

**ZF** *Zero Forcing*- critério de equalização Força para Zero

# Lista de Símbolos

$b_k$	Bits de informação
$c_n$	Símbolos codificados por um codificador convolucional
$c_n^\pi$	Símbolos codificados por um codificador e embaralhados por um entrelaçador temporal
$g_i$	Coefficiente $i$ do canal com múltiplos percursos
$M$	Número de estados de uma treliça que representa uma cadeia de Markov
$S_i$	Estado $i$ de uma treliça
$\mathbf{v}$	Vetor de amostras com IIS e sem presença de ruído AWGN
$L^E(c_n^\pi)$	Informação a posteriori de $c_n^\pi$
$L_a^E(c_n^\pi)$	Informação a priori de $c_n^\pi$
$L_{ext}^E(c_n^\pi)$	Informação extrínseca de $c_n^\pi$
$L^E(c_n)$	Informação a posteriori de $c_n$
$L_a^E(c_n)$	Informação a priori de $c_n$
$L_{ext}^E(c_n)$	Informação extrínseca de $c_n$
$\mathbf{Z}$	Vetor de probabilidades a priori para o decodificador convolucional
$L^D(c_n)$	Informação a posteriori do decodificador do símbolo $c_n$
$L^D(b_k)$	Informação a posteriori do decodificador do bit de informação $b_k$
$L_{ext}^D(c_n)$	Informação extrínseca do símbolo $c_n^\pi$
$\alpha$	Função densidade de probabilidade no sentido direto da treliça
$\beta$	Função densidade de probabilidade no sentido reverso da treliça
$\gamma$	Função densidade de probabilidade da transição de estados na treliça
$\sigma^2$	Variância (potência) do ruído AWGN
$\mathbf{R}$	Função risco
$L$	Comprimento do bloco de informação
$d$	Número de dimensões
$k$	Comprimento do bloco na entrada do codificador componente SPC
$n$	Comprimento do bloco na saída do codificador componente SPC

---

$h$	Distância de Hamming do código componente SPC
$K$	Comprimento do bloco de entrada do código produto SPC-TPC
$v$	Comprimento do bloco de saída do código produto SPC-TPC
$r$	Taxa de codificação do código componente SPC
$\delta_{min}$	Distância de Hamming do código produto SPC-TPC
$x_n$	Símbolo codificado pelo codificador SPC-TPC no instante $n$
$y_n$	Amostra na saída do filtro casado no instante $n$
$E$	Energia do símbolo na saída do codificador SPC-TPC
$L_C^D(y_n)$	Informação de canal na entrada do decodificador SPC-TPC
$L^D(\hat{x}_n)$	Informação a posteriori do símbolo $x_n$ na saída do decodificador SPC-TPC
$L_a^D(x_n)$	Informação a priori do símbolo $x_n$ na entrada do decodificador SPC-TPC
$L_{ext}^D(\hat{x}_n)$	Informação extrínseca do símbolo $x_n$ na saída do decodificador SPC-TPC
$L_q(\hat{x}_n)$	Informação a posteriori do símbolo $x_n$ na dimensão $q$
$L_{a(q)}(x_n)$	Informação a priori do símbolo $x_n$ na dimensão $q$
$L_{ext(q)}(\hat{x}_n)$	Informação extrínseca do símbolo $x_n$ na dimensão $q$
$x_n^\pi$	Símbolo código entrelaçado
$\hat{b}_k$	bit de informação estimado
$L_a^E(x_n^\pi)$	Informação a priori do símbolo $x_n^\pi$ na entrada do equalizador
$q$	dimensão a ser decodificada no bloco decodificador
$p$	indexador das iterações do equalizador turbo
$h_1$	vetor que contém os coeficientes do canal 1
$B$	número de palavras-código processadas pelo entrelaçador temporal

# Resumo

Combinando blocos já existentes de uma maneira diferente, um novo esquema de equalização turbo é proposto nesta dissertação para canais fixos com múltiplos percursos. Os esquemas existentes na literatura de equalização turbo conjunta com decodificação turbo realizam uma iteração turbo completa no decodificador a cada iteração do equalizador turbo. No esquema proposto nesta dissertação é utilizado apenas um decodificador componente de um código turbo. A cada iteração é realizada uma equalização e uma decodificação de uma dimensão de um código produto de paridade simples. Desta maneira, não é necessário processar todas as dimensões do código produto em cada iteração do equalizador turbo. Os resultados obtidos por simulações revelam que, para altas taxas de codificação, o desempenho deste esquema é superior aos equalizadores turbo apresentados na literatura.

Palavras-chave: Equalização Turbo, decodificação turbo, códigos produto de paridade simples.

# Abstract

Combining existing blocks in a novel way, a turbo equalizer structure is proposed here for fixed multipath channels. The existing combined turbo equalization with turbo decoding make a complete turbo iteration in the turbo decoder at each turbo equalizer iteration. The proposed scheme here uses a single component decoder of a block turbo code. At each iteration is performed an equalization and a decoding of a one dimension of a single parity check product code. Therefore, it is not necessary to process all dimensions of a product code at each turbo equalizer iteration. The results obtained through simulation reveals that, when high code rates are used, the performance of this scheme can be superior to the existing turbo equalizer.

Keywords: Turbo Equalization, Turbo Decoding, single parity check turbo product code.



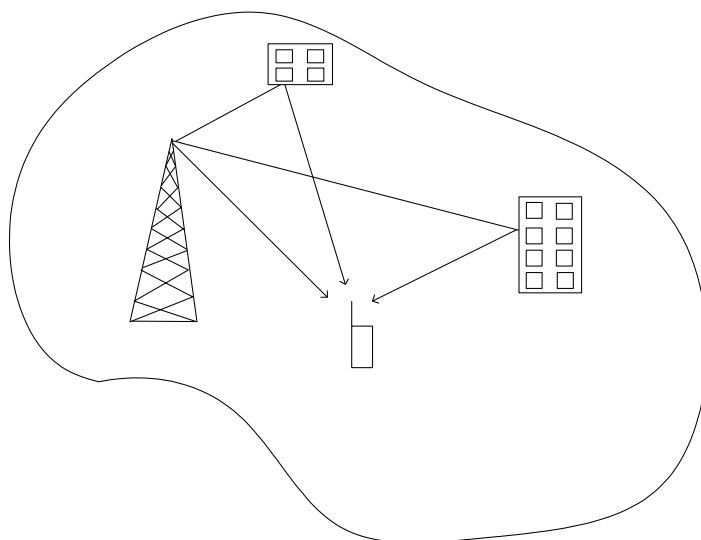
# Capítulo 1

## Introdução

Desde o advento da comunicação digital têm-se empreendido grande esforço em sistemas que permitam maior eficiência nos sistemas de transmissão. O ano de 1948 foi um grande marco na história da comunicação digital, quando Shannon [Sha48] apresenta sua teoria sobre um modelo matemático associado aos limites na transmissão de informação em um canal com a presença de ruído *Additive White Gaussian Noise* (AWGN), com limitações de largura de faixa e na potência de transmissão. Para que sejam aproximados os limites apresentados por Shannon, têm-se utilizado códigos corretores de erro de várias classes. Os códigos corretores de erro inserem bits de redundância na mensagem a ser transmitida, formando uma palavra código. Na recepção, o decodificador utiliza algum algoritmo que permita detectar e corrigir eventuais erros através da redundância inserida na transmissão. A taxa de codificação é definida pelo número de bits de informação, dividido pelo número total de bits na palavra código. Portanto, quanto menor a taxa de codificação, maior é a quantidade de redundância inserida na informação e menor é o aproveitamento da banda de transmissão. Por esta razão, códigos que possuem uma alta capacidade de correção de erros utilizando altas taxas de codificação são um grande atrativo. Nesta dissertação, é de especial interesse os códigos turbo, especialmente os códigos turbo de bloco que alcançam bons resultados a altas taxas de codificação.

Um sinal ao ser transmitido é degradado não só pelo ruído AWGN, mas

também pela Interferência Intersimbólica (IIS). Como exemplo, no canal de comunicação sem fio apresentado na Figura 1.1, o sinal recebido é influenciado por suas réplicas causadas principalmente por difração, reflexão e espalhamento do sinal nos obstáculos que circundam o receptor (Rx) e/ou que se localizam entre transmissor (Tx) e receptor, e também por influência da atmosfera.



**Figura 1.1:** Canal de comunicação com múltiplos percursos.

Os múltiplos percursos (MP) provocam um espalhamento temporal no símbolo transmitido. Desta maneira, o símbolo recebido é composto do símbolo transmitido em questão e de réplicas de símbolos adjacentes no tempo. Para estes canais, o uso de códigos de corretores de erro muitas vezes é insuficiente para viabilizar uma comunicação. Neste caso, para combater as degradações no domínio do tempo e na frequência faz-se uso de equalizadores.

Um método bastante simples de equalização é o critério do *Zero Forcing* (ZF) [Hay01]. Um equalizador ZF é constituído de um filtro digital que procura inverter a função de transferência do canal com MP. Portanto, ao convoluir a resposta do canal com MP e a resposta do equalizador ZF, a resposta em frequência total se encontra livre de distorções. Considerando a dualidade tempo x frequência, o sinal processado pelo equalizador não é mais espalhado no domínio do tempo.

---

Desta maneira, a influência de símbolos vizinhos é forçada para zero, justificando assim a sigla ZF. No entanto, apesar do equalizador ZF combater o efeito da IIS dentro de uma faixa temporal, este apresenta um acréscimo de ruído em sua saída [Hay01].

Outra alternativa é o uso de equalizadores que minimizam o erro médio quadrático entre o sinal estimado e, por exemplo, uma sequência de treinamento gerada no receptor igual a sequência transmitida. Esta classe de equalizadores é amplamente utilizada porque combate o efeito da IIS sem acréscimo de ruído em sua saída.

Um outro tipo de equalizador de grande importância para esta dissertação, é o equalizador de treliça [Pro95]. Esta classe de equalizadores trabalha com o reconhecimento da treliça que representa o canal com MP.

Os sistemas de equalização e decodificação citados até agora trabalham de maneira separada no combate da IIS e do ruído AWGN, ou seja, o equalizador trabalha de maneira independente do decodificador. No Equalizador Turbo introduzido por [Dou95], equalizador e decodificador trabalham de maneira conjunta. Um sistema de Equalização Turbo nada mais é que um equalizador e um decodificador que trabalham de maneira iterativa como em um decodificador turbo [Ber93].

O primeiro Equalizador Turbo [Dou95] era composto de um decodificador convolucional e um equalizador de treliça. Desde então, vários resultados foram obtidos utilizando esta configuração [Bau97][Chug97][Koe03]. Os resultados obtidos mostram que o desempenho de um sistema composto por um Equalizador Turbo que utiliza um determinado código em uma canal com IIS pode se aproximar do desempenho de um sistema composto apenas por este mesmo código em um canal AWGN.

Desde então, várias configurações de equalizadores turbo vem sendo investigadas. Em [Gla97] o equalizador de treliça é trocado por um *cancelador de*

---

*interferências* e em [Tuc00] é utilizado um equalizador de minimização de erro médio quadrático (MMSE - *Minimum Mean Square Error*). Estes equalizadores apresentam algoritmos sub-ótimos, mas possuem uma baixa complexidade computacional em relação ao equalizador de treliça.

Outra modificação do equalizador turbo proposto em [Dou95] que vem sendo realizada na literatura, é a troca do decodificador convolucional por um código turbo. Os códigos turbo, apresentados em [Ber93], alcançam excelentes desempenhos em canais AWGN, se aproximando do limite de Shannon. Desta maneira, o uso desta classe de decodificadores em equalizadores turbo se torna bastante atrativo.

Em [Rap97] o decodificador convolucional é substituído por um decodificador turbo convolucional. Em [Noo03] é usado um decodificador turbo de bloco BCH (Bose-Chaudhuri-Hocquenghem). Em [Yea02] é realizada uma comparação de desempenho dos equalizadores turbo utilizando os códigos convolucional, convolucional turbo e bloco BCH turbo. Esta última referência será utilizada em grande parte desta dissertação para efeito de comparação de desempenho do equalizador turbo aqui proposto com os demais existentes.

O equalizador turbo proposto aqui é composto de um equalizador de treliça e de um código produto de paridade simples com decodificação turbo (SPC-TPC - *Single Parity Check Turbo Product Codes*). A escolha do código SPC-TPC deve-se ao fato deste possuir um algoritmo de decodificação bastante simples. Além de se utilizar um código que possui um algoritmo simples de decodificação, o equalizador turbo proposto aqui não possui um decodificador turbo completo, mas um decodificador de apenas uma dimensão do código produto. Portanto, o assim chamado *equalizador turbo com decodificação parcial SPC-TPC* possui um algoritmo bastante simples em relação aos apresentados em [Yea02].

Apesar do equalizador turbo com decodificação parcial SPC-TPC possuir uma implementação simples, os resultados obtidos foram superiores aos equalizadores

---

turbo utilizados em [Yea02] para altas taxas de codificação. Os equalizadores turbo utilizados em [Yea02] possuem códigos mais poderosos em canal AWGN do que os códigos SPC-TPC utilizados nesta dissertação. No entanto, o equalizador turbo proposto apresentou uma maior redução de IIS do que os utilizados em [Yea02]. Desta maneira, mesmo com a utilização de códigos inferiores, os desempenhos obtidos foram superiores aos equalizadores turbo considerados em [Yea02], para códigos de taxa alta.

O enfoque dado nesta dissertação é o de trabalhar com alterações no bloco que realiza a decodificação no equalizador turbo. Como dito anteriormente, o bloco que realiza a equalização no equalizador turbo é um equalizador de treliça. O equalizador de treliça possui uma complexidade computacional impraticável para canais com alta dispersão temporal e modulações de alta ordem. No entanto, o equalizador de treliça foi utilizado para que seja possível a comparação com os equalizadores turbo analisados em [Yea02]. Para que o equalizador turbo com decodificação parcial se torne realizável para canais com grande dispersão temporal e modulações de alta ordem, pode-se utilizar no bloco de equalização os equalizadores já citados anteriormente, como o cancelador de interferências e o equalizador MMSE [Gla97][Tuc00].

A dissertação está organizada da seguinte maneira: o Capítulo 2 faz uma introdução sobre equalizadores turbo, tendo como referência um equalizador turbo que utiliza um código convolucional. O algoritmo BCJR (Bahl-Cocke-Jelinek-Raviv) é aplicado ao equalizador de treliça e ao decodificador convolucional. O Capítulo 3 é o principal. Primeiramente é realizada uma descrição do processo de formação dos códigos SPC-TPC, bem como de sua decodificação. Depois é utilizado um decodificador turbo SPC-TPC no bloco de decodificação de um equalizador turbo. Finalmente é apresentado o *equalizador turbo com decodificação parcial SPC-TPC*, onde é utilizado no bloco decodificador um decodificador de apenas uma dimensão do código SPC-TPC. O Capítulo 4 apresenta e faz

uma análise dos resultados obtidos através de simulações computacionais com o equalizador turbo proposto. O equalizador turbo de decodificador de dimensão única SPC-TPC é comparado com o equalizador turbo que possui o decodificador turbo completo SPC-TPC e também com os equalizadores turbo utilizados em [Yea02]. Ao comparar os resultados obtidos com os equalizadores turbo da literatura [Yea02], constatou-se que o equalizador turbo com decodificação parcial obteve um melhor desempenho para altas taxas de codificação. Finalmente no Capítulo 5 são tecidos os comentários finais e são também apontadas as maiores contribuições deste trabalho e os trabalhos futuros que podem ser realizados sobre o assunto.

# Capítulo 2

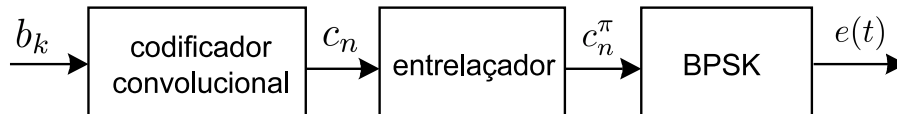
## Equalização Turbo

Este capítulo tem o objetivo de apresentar os conceitos envolvidos na equalização turbo. A Seção 2.1 apresenta o modelo discreto (no tempo) do conjunto formado pelo modulador, o canal com múltiplos percursos e o demodulador. A partir deste modelo discreto pode-se observar que um símbolo transmitido se sobrepõe aos símbolos vizinhos, resultando no efeito da IIS. Nota-se também que este modelo discreto se comporta de forma semelhante a um codificador convolucional [Dou95], o que permite sua caracterização como um codificador interno do sistema. Para fins de simplificação, no texto desta dissertação este modelo composto por modulador, canal e demodulador será citado apenas como modelo discreto do canal. Na Seção 2.2 é apresentado o princípio de funcionamento de um equalizador turbo, no qual as iterações são interpretadas como uma cascata de módulos, como apresentado em [Dou95][Gla97][Lao01]. A Seção 2.3 trata do algoritmo Máximo-a-Posterior (do Latin, Maximum-a-Posteriori, MAP) e sua implementação nos dois blocos do equalizador turbo: o bloco de equalização e o bloco de decodificação [Koe03]. A Seção 2.4 faz uma referência aos critérios de parada apresentados em [Bau98], que determinam o número ótimo de iterações que devem ser realizadas para que o sistema proporcione um desempenho desejado. Finalmente, na Seção 2.5 faz-se uma análise dos resultados obtidos nas publicações mais recentes sobre o assunto e os comentários finais sobre quesitos

como complexidade e desempenho.

## 2.1 Modelo Discreto do Canal

Ao longo deste capítulo o sistema de transmissão utilizado como referência será aquele apresentado na Figura 2.1. Primeiramente, os bits de informação  $b_k$  são codificados através de um codificador convolucional em símbolos  $c_n$ . Os símbolos  $c_n$  são embaralhados por um entrelaçador temporal (*interleaver*), usado para que os erros em rajada se apresentem descorrelacionados após a passagem pelo canal e pelo bloco de desentrelaçamento temporal (*de-interleaver*). Os símbolos  $c_n^\pi$  são mapeados em uma constelação BPSK e transmitidos através de um canal com múltiplos percursos e ruído AWGN.

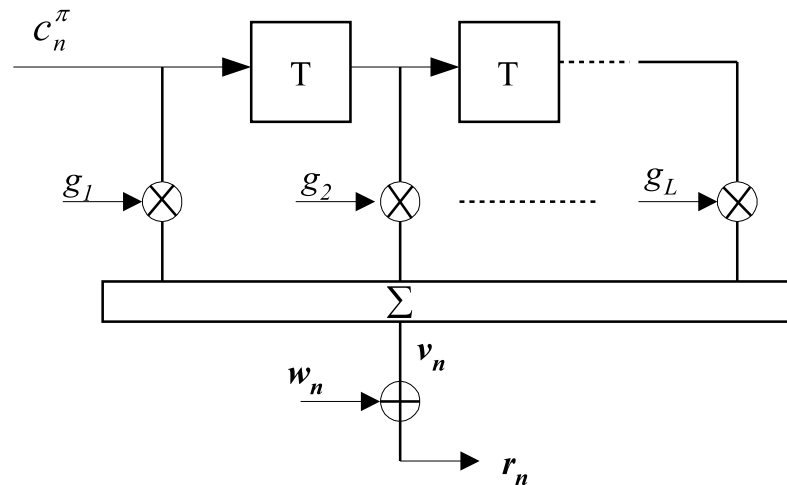


**Figura 2.1:** *Sistema de transmissão*

Em um canal onde há propagação por múltiplos percursos, o sinal recebido é resultado da soma de infinitas réplicas (ou ecos) do sinal transmitido [Hay01]. Como é inviável modelar este canal com infinitos múltiplos percursos, por conta da dificuldade de análise matemática, pode-se admitir como aproximação que ele envolva a soma dos  $N$  percursos mais significativos. O modelo discreto de uma canal com múltiplos percursos pode ser representado por uma linha de atrasos com derivações (*tapped delay line*) [Pro95], como ilustrado na Figura 2.2. Nessa figura, os coeficientes  $g_l$ ,  $l = 1, 2, \dots, L$  representam as variações de intensidade do sinal do  $n$ -ésimo percurso de propagação,  $v_n$  é a saída distorcida pela IIS e  $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_n \ \dots \ r_N]^T$  é o vetor na saída do canal após a adição do ruído AWGN  $w_n$ .

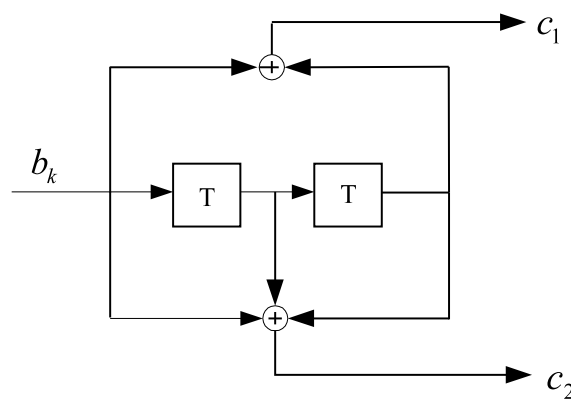
O modelo discreto do canal apresentado na Figura 2.2 pode ser representado





**Figura 2.2:** Modelo discreto do canal com IIS

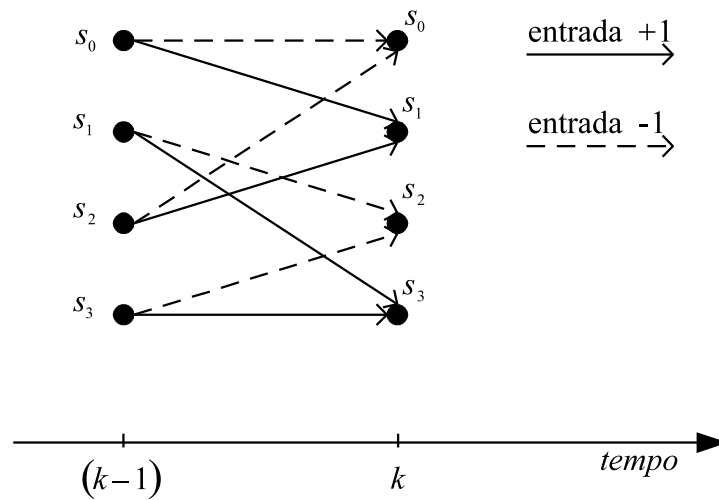
por uma treliça associada a uma cadeia de Markov [Dou95] da mesma maneira que o codificador convolucional da Figura 2.3.



**Figura 2.3:** Codificador convolucional de taxa 1/2

Como exemplo, a treliça da Figura 2.4 representa a transição de estados de um codificador convolucional como o da Figura 2.3 ou de um canal com dois múltiplos percursos (canal com IIS).

Como o canal e o codificador são compostos por dois atrasadores, as treliças do canal e do codificador serão compostas por  $2^M$  estados, onde  $M$  é o número de atrasadores, no caso  $M = 2$  para o canal e para o codificador, o que resulta



**Figura 2.4:** Treliça que representa as transições de estados para um codificador convolucional de comprimento 3 ou de um canal com dois múltiplos percursos, onde  $S_0 = \{-1, -1\}$ ,  $S_1 = \{+1, -1\}$ ,  $S_2 = \{-1, +1\}$  e  $S_3 = \{+1, +1\}$ . Para o codificador convolucional o valor  $-1$  é trocado por 0 nos valores dos estados.

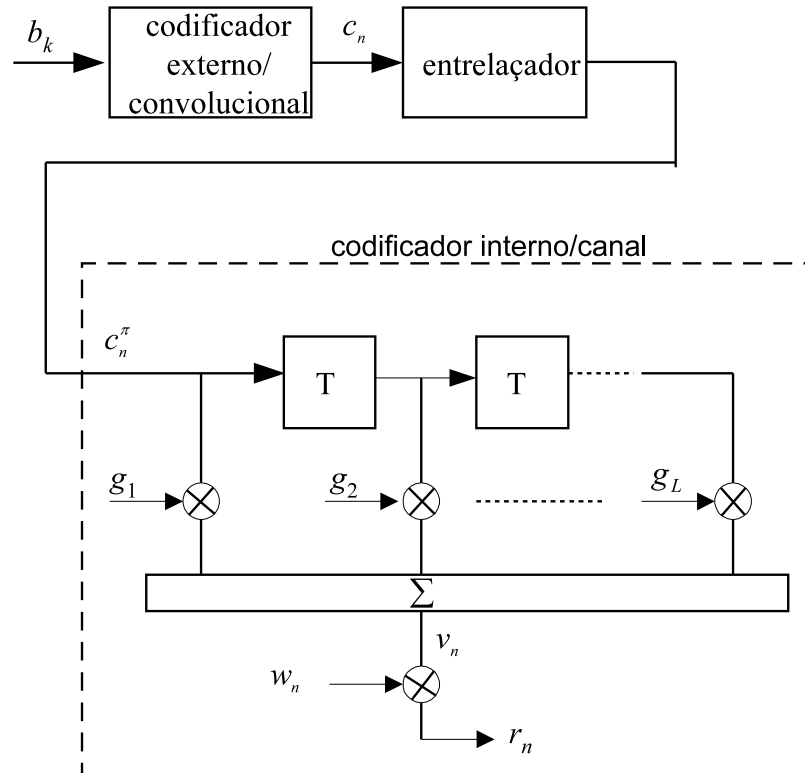
em uma treliça de quatro estados.

Para exemplificar a formação da treliça toma-se a transição de  $S_0$  para  $S_1$ . O estado  $S_0$  é composto pelos símbolos  $\{-1, -1\}$  na saída dos registradores da linha de atraso. Ao transmitir o símbolo  $+1$  pelo canal o conteúdo na saída dos registradores passa a ser  $\{+1, -1\}$ , que corresponde ao estado  $S_1$ . Da mesma maneira são formadas as demais transições.

Partindo-se desse princípio de que o canal com IIS pode ser representado como uma treliça da mesma maneira que um codificador convolucional, o canal será tratado como um codificador componente do sistema de transmissão, num processo de concatenação serial, o que permitirá que a equalização e a decodificação sejam implementadas de maneira iterativa, como em um decodificador turbo.

## 2.2 Princípios da Equalização Turbo

A Figura 2.5 apresenta o canal como um codificador interno do sistema de transmissão e o codificador convolucional como sendo um codificador externo. Nos sistemas convencionais de recepção a recuperação da informação é feita de acordo com a Figura 2.6. Primeiramente é feita a equalização de canal através de, por exemplo, um filtro empregando o algoritmo LMS (*Least Mean Square*) ou um equalizador de treliça empregando o critério MLSE (*Maximum Likelihood Sequence Estimation*) [Pro95]. Em seguida, os símbolos codificados  $c_n^\pi$  na saída do equalizador são reordenados pelo desentrelaçador temporal que fornece os símbolos  $c_n$  na ordem correta para o decodificador convolucional, implementado tipicamente com a versão convencional do algoritmo de Viterbi, versão esta que opera com entradas suaves ou abruptas e fornece decisões abruptas. Finalmente, na saída do decodificador os bits de informação  $\hat{b}_k$  são estimados.



**Figura 2.5:** Sistema de transmissão com canal sendo o codificador interno



**Figura 2.6:** Sistema de recepção convencional com decisões tomadas no equalizador e no decodificador de maneira separada.

Onde a equalização e a decodificação são realizadas separadamente, o equalizador não explora a estrutura do código para realizar suas decisões. Na equalização turbo realiza-se, iterativamente, uma realimentação de informação das decisões do decodificador no equalizador de tal maneira que o equalizador tenha uma medida da confiabilidade de suas decisões e assim atinja melhor desempenho iteração a iteração.

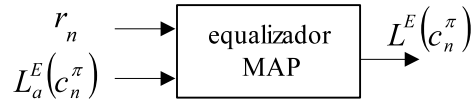
Para que o equalizador e o decodificador possuam uma medida de confiabilidade, suas decisões devem ser suaves. Com este intuito, o equalizador turbo é constituído de dois blocos de entradas e saídas suaves ( *Soft-Input Soft-Output*, SISO). O primeiro bloco é o bloco equalizador apresentado na Figura 2.7. Este bloco é um equalizador que tem sua saída na forma de uma razão de log-verossimilhança ( *Log-likelihood Ratio*, LLR) das probabilidades a posteriori de acordo com:

$$L^E(c_n^\pi) = \ln \frac{P(c_n^\pi = +1 | \mathbf{r})}{P(c_n^\pi = -1 | \mathbf{r})} \quad (2.1)$$

onde  $c_n^\pi$  são os símbolos após passarem pelo entrelaçador e  $\mathbf{r}$  é o vetor de amostras recebidas do canal com IIS. A magnitude de  $L^E(c_n^\pi)$  contém a informação da confiabilidade da decisão tomada pelo equalizador. A partir do sinal de  $L^E(c_n^\pi)$  pode-se tomar uma decisão abrupta sobre os símbolos da seguinte maneira:

$$\text{Se } L^E(c_n^\pi) \geq 0, \text{ então } c_n^\pi = +1$$

$$\text{Se } L^E(c_n^\pi) < 0, \text{ então } c_n^\pi = -1$$



**Figura 2.7:** Equalizador MAP (*Maximo a Posteriori*).

Para o cálculo da saída suave  $L^E(c_n^\pi)$ , o bloco equalizador tem como entrada o vetor de amostras recebidas do canal com IIS,  $\mathbf{r}$ , e a LLR da informação a priori dos símbolos  $c_n^\pi$  dada por:

$$L_a^E(c_n^\pi) = \ln \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \quad (2.2)$$

Aplicando a regra de Bayes em (2.1), a saída do bloco equalizador pode ser reescrita como:

$$\begin{aligned} L^E(c_n^\pi) &= \ln \frac{P(c_n^\pi = +1|\mathbf{r})}{P(c_n^\pi = -1|\mathbf{r})} \\ &= \ln \frac{p(\mathbf{r}|c_n^\pi = +1)}{p(\mathbf{r}|c_n^\pi = -1)} + \ln \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \end{aligned} \quad (2.3)$$

A primeira parcela em (2.3) se refere a *informação extrínseca* na saída do bloco equalizador que será representada por  $L_{ext}^E(c_n^\pi)$ . A segunda parcela se refere a LLR das probabilidades *a priori* dos símbolos  $c_n^\pi$ , que será representada por  $L_a^E(c_n^\pi)$ .

A *informação extrínseca* pode ser interpretada como o ganho de informação obtido sobre um símbolo no processo de equalização ou decodificação [Dou95] ou como a LLR na saída de um decodificador SISO menos a LLR da entrada [Hay01]. Em [Yea02] a *informação extrínseca* de um bit é definida como a informação gerada pelo equalizador ou decodificador baseada na sequência recebida e na *informação a priori* de todos os outros bits, com exceção do próprio bit recebido e a *informação a priori* relacionada a este bit específico. A *informação a priori*

é definida como a informação conhecida antes do processo de equalização ou decodificação e originada de uma fonte que não seja a sequência recebida ou baseado nas características do código.

O outro bloco SISO do equalizador turbo é o bloco decodificador dado pela Figura 2.8. Este bloco é constituído de uma entrada que recebe o vetor  $\mathbf{Z}$  que tem como elementos uma estimativa das probabilidades *a priori* dos símbolos  $c_n$ :

$$\mathbf{Z} = [P(c_1|\mathbf{r}) \ P(c_2|\mathbf{r}) \ \dots \ P(c_N|\mathbf{r})]^T \quad (2.4)$$

A partir do vetor de entrada  $\mathbf{Z}$  o bloco decodificador calcula as LLR's dos símbolos  $c_n$  e dos bits de informação  $b_k$ .



**Figura 2.8:** Decodificador MAP (*Maximo a Posteriori*).

Da mesma maneira que no bloco equalizador, a saída  $L^D(c_n)$  do bloco decodificador pode ser decomposta em *informação extrínseca* e *informação a priori*:

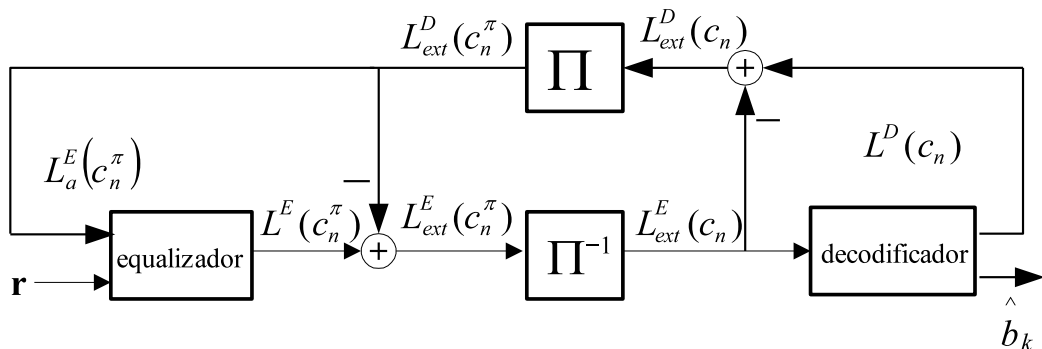
$$\begin{aligned} L^D(c_n) &= \ln \frac{p(\mathbf{Z}|c_n = +1)}{p(\mathbf{Z}|c_n = -1)} \\ &= \ln \frac{p(\mathbf{Z}|c_n = +1)}{p(\mathbf{Z}|c_n = -1)} + \ln \frac{P(c_n = +1)}{P(c_n = -1)} \end{aligned} \quad (2.5)$$

onde o primeiro termo de (2.5) é a *informação extrínseca*  $L_{ext}^D(c_n)$  do símbolo  $c_n$  e o segundo termo é a *informação a priori*,  $L_a^D(c_n)$  desse símbolo.

Na equalização turbo utiliza-se a *informação extrínseca*  $L_{ext}^D(c_n)$  obtida pelo bloco decodificador como uma estimativa da *informação a priori* a ser operada pelo bloco equalizador. De forma análoga, a *informação extrínseca* obtida pelo bloco equalizador é utilizada como *informação a priori* a ser processada pelo bloco

decodificador. Portanto, um equalizador turbo opera como um decodificador turbo, onde um código componente é o codificador convolucional e outro código componente pode ser associado ao canal com múltiplos percursos.

A Figura 2.9 ilustra o processo de equalização turbo. Através do vetor de amostras recebidas do canal,  $\mathbf{r}$ , e da *informação a priori*  $L_a^E(c_n^\pi)$  o bloco equalizador calcula sua saída suave  $L^E(c_n^\pi)$ . De acordo com a equação (2.3), subtrai-se da saída suave total do bloco equalizador  $L^E(c_n^\pi)$  a *informação a priori*  $L_a^E(c_n^\pi)$  para que se transmita somente a *informação extrínseca*  $L_{ext}^E(c_n^\pi)$  para o bloco de decodificação. A *informação extrínseca*  $L_{ext}^E(c_n^\pi)$  passa pelo desentrelaçador temporal e é utilizada como uma estimativa da *informação a priori* para o bloco decodificador,  $L_a^D(c_n)$ . O bloco decodificador calcula sua saída suave  $L^D(c_n)$  e subtrai a parcela da informação a priori  $L_a^D(c_n)$  para o cálculo da *informação extrínseca*  $L_{ext}^D(c_n)$ , de acordo com (2.5). A *informação extrínseca*  $L_{ext}^D(c_n)$  passa pelo entrelaçador temporal e é usada como uma estimativa da *informação a priori*,  $L_a^E(c_n^\pi)$ , para o bloco equalizador. O papel do entrelaçador e do desentrelaçador temporal no equalizador turbo é de organizar as LLR's *a priori* na entrada do bloco equalizador e do bloco decodificador, respectivamente. Outra função do entrelaçamento temporal é a de decorrelacionar as informações trocadas pelos blocos constituintes, o que será explicado com mais detalhes adiante.



**Figura 2.9:** *Equalizador Turbo.*

O processo descrito no parágrafo anterior se repete tantas vezes quanto necessário, até que se atinja a convergência em termos de estabilização dos valores de informação extrínseca, o que leva a uma estabilização no desempenho final da equalização turbo em termos de taxa de erro de bit ou de bloco. A cada repetição (iteração) há um melhoramento da confiabilidade das informações extrínsecas dos blocos equalizador e decodificador, diminuindo a chance de erros nas decisões sobre os bits  $b_k$  ou sobre os símbolos  $c_n$  [Bau97]. Na primeira iteração, como não há *informação extrínseca* disponível vinda do decodificador, a LLR *a priori* operada pelo bloco equalizador,  $L_a^E(c_n^\pi)$ , é feita igual a zero (probabilidades a priori iguais a 1/2) e a cada iteração a estimativa desta LLR é melhorada. Ao final das iterações pode-se realizar uma decisão abrupta sobre os bits de informação  $b_k$  ou sobre os símbolos  $c_n$ , a partir da saída suave do decodificador.

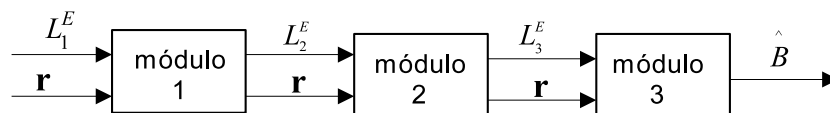
O uso da *informação extrínseca* como estimativa da *informação a priori* dos blocos constituintes do processo de equalização turbo deve garantir que ela seja decorrelacionada em relação aos valores de *informação extrínseca* calculados pelo bloco anterior. Isto é feito para que um bloco não opere com valores que são resultado de decisões tomadas pelo próprio bloco constituinte em questão. Por exemplo, de acordo com (2.3), a saída suave  $L^E(c_n^\pi)$  do equalizador é composta da *informação a priori*  $L_a^E(c_n^\pi)$  e da *informação extrínseca*  $L_{ext}^E(c_n^\pi)$ . Portanto, a saída suave total  $L^E(c_n^\pi)$  do bloco equalizador não deve ser usada como estimativa da *informação a priori* para o bloco decodificador, já que a parcela  $L_a^E(c_n^\pi)$  que compõe a saída  $L^E(c_n^\pi)$  é obtida a partir da saída do bloco decodificador. Do mesmo modo, a saída  $L^D(c_n)$  do bloco decodificador apresenta uma certa correlação com as LLR's de saída obtidas anteriormente pelo bloco equalizador. Para evitar esta redundância é que se utiliza apenas a *informação extrínseca* como estimativa da entrada *a priori* do bloco seguinte.

Para que a correlação remanescente entre os valores de *informação extrínseca* operados pelos blocos equalizador e decodificador seja minimizada, um adequado



projeto do entrelaçador temporal se faz necessário, objetivando também minimizar os efeitos dos erros em rajada causados por canais com memória. Desta maneira, quanto maior é a capacidade do entrelaçador de descorrelacionar os valores por ele operados, maior é o ganho que se tem em termos de redução na taxa de erro de bit a cada iteração do equalizador turbo [Koe03].

O processo de equalização turbo também pode ser representado como uma linha de módulos (*pipeline*) como na Figura 2.10, onde cada módulo representa uma iteração do equalizador turbo. Cada módulo possui duas entradas: o vetor de amostras recebidas do canal,  $\mathbf{r}$ , e a informação a priori do bloco equalizador  $L_a^E(c_n^\pi)$ . Na saída de um módulo tem-se a *informação extrínseca* do bloco decodificador, esta é utilizada como estimativa da *informação a priori* do próximo módulo (próxima iteração). A saída de cada módulo possui uma maior confiabilidade em relação à saída do módulo anterior [Bri01] [Otn02]. Ao final das iterações, a partir da saída suave do último módulo é realizada uma decisão abrupta em relação aos bits de informação  $b_k$  ou em relação aos símbolos  $c_n$ .



**Figura 2.10:** Intepretação do Equalizador Turbo ataravés de módulos.

Neste ponto merecem ser comentadas algumas questões relacionadas ao tempo que o equalizador turbo precisa para obter convergência. Duas medidas de tempo podem ser aqui consideradas: a *latência* e o *atraso*. O atraso é inversamente proporcional à taxa de transmissão e diretamente proporcional ao tamanho do bloco codificado, pois de forma independente do tipo de decodificação, há que se esperar pela recepção de todos os símbolos associados a uma palavra-código para que a estimação dessa palavra código se inicie. Já a latência está associada ao tempo de processamento necessário para que uma palavra de informação seja estimada.

Este tempo depende do número de iterações (número de módulos) no processo de equalização turbo e também da complexidade dos algoritmos utilizados nos blocos componentes (bloco de equalização e bloco de decodificação). Nesse sentido, um dos grandes desafios apresentados ao desenvolvimento de novos processos de equalização turbo está relacionado com o objetivo de redução da latência, assim como ocorre com os códigos turbo.

Os primeiros sistemas de equalização turbo tratados em [Dou95], [Bau97] e [Chug97] são constituídos de blocos equalizador e decodificador que utilizam o algoritmo de decodificação MAP, implementados por algoritmos SISO como o algoritmo sub-ótimo SOVA (*Soft Output Viterbi Algorithm*) [Hag89] e o algoritmo ótimo BCJR [Bah74] (que possui uma complexidade computacional crescente com o número de estados das treliças que representam o canal e o decodificador). Outras alternativas que simplificam o algoritmo BCJR podem ser encontradas em [Bau98], como o algoritmo *Log-MAP* que opera no domínio logarítmico sem perda de desempenho e o algoritmo sub-ótimo *Max-Log-MAP*.

Em [Gla97] é utilizada uma alternativa sub-ótima (um *Cancelador de Interferência* [Ger81] [Mul81]) no bloco de equalização. Esta alternativa apresenta uma grande simplicidade computacional em relação ao algoritmo BCJR. Outra alternativa para o bloco equalizador é apresentada em [Tuc00], onde é usado no equalizador um filtro que procura minimizar o erro médio quadrático (MMSE - *Minimum Mean Square Error*) entre o sinal estimado e as LLR's a priori fornecidas pelo decodificador.

## 2.3 Algoritmo BCJR

O algoritmo BCJR, que implementa o critério de decisão MAP símbolo-a-símbolo, foi proposto em 1974 por Bahl, Cocke, Jelinek e Raviv [Bah74]. Para o entendimento do algoritmo, toma-se os bits transmitidos,  $b_k$ , que passarão por um sistema que é modelado por uma treliça como a da Figura 2.4. Esta treliça pode repre-

sentar um modelo de Markov, como um canal com múltiplos percursos ou um codificador convolucional. O algoritmo BCJR tem como objetivo calcular a LLR:

$$L(b_k) = \ln \frac{P(b_k = +1 | \mathbf{r})}{P(b_k = -1 | \mathbf{r})} \quad (2.6)$$

Aplicando o teorema de Bayes em (2.6), obtém-se:

$$L(b_k) = \ln \frac{p(b_k = +1, \mathbf{r})}{p(b_k = -1, \mathbf{r})} \quad (2.7)$$

A Equação (2.7) pode ser expressa da seguinte maneira [Han02]:

$$L(b_k) = \ln \frac{\sum_{b_k=+1} p(S_{k-1}, S_k, \mathbf{r})}{\sum_{b_k=-1} p(S_{k-1}, S_k, \mathbf{r})} \quad (2.8)$$

onde  $S_k$  se refere ao estado que a treliça se encontra no instante de tempo presente e  $S_{k-1}$  ao estado que a treliça se encontrava no instante de tempo anterior. Note que o numerador de (2.8) representa as probabilidades de ocorrência das transições causadas pelo símbolo  $+1$  (linhas contínuas da Figura 2.4), enquanto o denominador se refere aos símbolos  $-1$ . De [Han02] pode-se demonstrar que [Han02]:

$$\begin{aligned} p(S_{k-1}, S_k, \mathbf{r}) &= p(r_{j>k} | s) p(r_k, s | s') p(s', r_{j<k}) \\ &= \beta_k(S_k) \gamma_k(S_{k-1}, S_k) \alpha_{k-1}(S_{k-1}) \end{aligned} \quad (2.9)$$

onde  $s$  se refere ao estado que a treliça se encontra no instante de tempo presente e  $s'$  no instante de tempo anterior. O termo  $\gamma_k(S_{k-1}, S_k)$  se refere a função densidade de probabilidade de receber  $r_k$  e a treliça se encontrar no estado  $s$ , dado que ela estava em  $s'$ . O termo  $\alpha_{k-1}(S_{k-1})$  se refere a função densidade de probabilidade que a treliça estava no estado anterior  $S_{k-1}$  e recebeu uma sequência passada  $r_{j<k}$ . O valor de  $\alpha$  pode ser calculado através de [Han02]:

$$\alpha_k(S_k) = \sum_{S_{k-1}} \gamma_k(S_{k-1}, S_k) \alpha_{k-1}(S_{k-1}) \quad (2.10)$$

O termo  $\beta_k(S_k)$  se refere a função densidade de probabilidade de receber uma sequência futura  $r_{j>k}$  dado o estado atual  $s$  da treliça. O valor de  $\beta$  pode ser calculado através de:

$$\beta_k(S_k) = \sum_{S_{k+1}} \gamma_k(S_{k-1}, S_k) \beta_{k+1}(S_{k+1}) \quad (2.11)$$

Os valores de  $\alpha$  são obtidos partindo do início para o fim da treliça de acordo com (2.10). Os valores de  $\beta$  são obtidos partindo do fim para o início da treliça através de (2.11).

Caso seja forçado, através de transmissão de zeros, que a treliça inicie e termine no estado  $S_0$ ,  $\alpha$  e  $\beta$  podem ser inicializados da seguinte maneira [Bau98]:

$$\begin{aligned} \alpha_{inicial}(0) &= 1 \text{ e } \alpha_{inicial}(s) = 0 \quad \forall s \neq 0 \\ \beta_{final}(0) &= 1 \text{ e } \beta_{final}(s) = 0 \quad \forall s \neq 0 \end{aligned} \quad (2.12)$$

Quando o estado inicial e o final não forem conhecidos,  $\alpha_{inicial}$  e  $\beta_{final}$  são inicializados como 1 para todos os estados [Koe03]. De posse dos valores de  $\alpha$  e  $\beta$ , resta calcular os valores de  $\gamma$  para se obter o valor dado por (2.9) para todas as transições possíveis e determinar  $L(b_k)$  através de (2.8).

Um receptor ideal seria um receptor que utilizasse o algoritmo BCJR em uma supertreliça que englobaria o codificador, o entrelaçador e o canal [Koe03]. No entanto, o modelamento de tal treliça seria extremamente complexo [Koe03]. O equalizador turbo trata de maneira separada a treliça do canal da treliça do decodificador. Portanto, o algoritmo BCJR deve ser aplicado ao equalizador e ao decodificador. Para ambos, o cálculo de  $\alpha$  e  $\beta$  é executado da mesma maneira. O que irá diferenciar o equalizador do decodificador é a maneira de calcular o valor de  $\gamma$ , como mostrado nas próximas seções.

### 2.3.1 Algoritmo BCJR Aplicado ao Bloco Equalizador

Nesta seção será apresentado o algoritmo BCJR aplicado ao bloco equalizador. Os valores de  $\alpha$  e  $\beta$  serão calculados a partir das equações (2.10) e (2.11) respectivamente. Para o cálculo de (2.10) e (2.11) é necessário calcular o valor de  $\gamma$ . De (2.9) tem-se:

$$\gamma_n(s', s) = p(r_n, s|s') \quad (2.13)$$

Aplicando a regra de Bayes modificada  $P(a, b|c) = P(a|b, c) \cdot P(b|c)$  em (2.13), tem-se:

$$\gamma_n(s', s) = p(r_n|s, s')P(s|s') \quad (2.14)$$

A primeira parcela de (2.14) é a função densidade de probabilidade de receber uma amostra  $r_n$  do canal dado que ocorreu uma transição  $(s', s)$ . A partir do modelo discreto do canal apresentado na Figura 2.2 observa-se que esta função densidade de probabilidade é dada por uma gaussiana com média  $v_n$ , que é o sinal com IIS sem adição de ruído, e com variância  $\sigma^2$  devido a adição do ruído AWGN. Portanto, a primeira parcela de (2.14) é dada por:

$$p(r_n|v_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(r_n - v_n)^2}{2\sigma^2}\right) \quad (2.15)$$

A segunda parcela de (2.14) é a probabilidade da treliça se encontrar no estado  $s$  dado que ela estava no estado  $s'$  anteriormente. Esta probabilidade é governada pela probabilidade a priori do símbolo de entrada  $c_n^\pi$ , já que é o mesmo que irá definir qual transição ocorre na treliça do canal. Substituindo tais valores em (2.14), tem-se:

$$\gamma_n(s', s) = P(c_n^\pi) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(r_n - v_n)^2}{2\sigma^2}\right) \quad (2.16)$$

onde  $P(c_n^\pi)$  é a probabilidade a priori dos símbolos  $c_n^\pi$ . Esta probabilidade é estimada através da *informação extrínseca* do decodificador  $L_{ext}^D(c_n^\pi)$ . Como a *informação extrínseca*  $L_{ext}^D(c_n^\pi)$  é usada como estimativa da informação a priori  $P(c_n^\pi)$ , ela pode ser escrita como:

$$L_{ext}^D(c_n^\pi) \approx \ln \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \quad (2.17)$$

A partir de (2.17) obtém-se a expressão da probabilidade a priori  $P(c_n^\pi)$  em função de  $L_{ext}^D(c_n^\pi)$ :

$$P(c_n^\pi = c) \cong \frac{\exp(c L_{ext}^D(c_n^\pi))}{1 + L_{ext}^D(c_n^\pi)} \quad (2.18)$$

onde  $c \in \{0, 1\}$  caso se deseje calcular  $P(c_n^\pi = +1)$  ou  $P(c_n^\pi = -1)$ , respectivamente. Aplicando (2.18) em (2.16) obtém-se o valor de  $\gamma$  para o bloco equalizador:

$$\gamma_n(s', s) = \frac{\exp(c L_{ext}^D(c_n^\pi))}{1 + L_{ext}^D(c_n^\pi)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(r_n - v_n)^2}{2\sigma^2}\right) \quad (2.19)$$

A partir de (2.19) pode-se notar o ganho de informação do equalizador turbo: na primeira iteração não há *informação extrínseca* do decodificador,  $L_{ext}^D(c_n^\pi)$ , re-alimentada para o equalizador; sendo assim, admite-se que os símbolos  $c_n^\pi$  sejam equiprováveis ou seja,  $P(c_n^\pi)$  é considerada igual 1/2. Nas próximas iterações o valor de  $L_{ext}^D(c_n^\pi)$  é usado para o cálculo de (2.19). A cada iteração, há um aperfeiçoamento na confiabilidade da *informação extrínseca* do decodificador,  $L_{ext}^D(c_n^\pi)$ , e conseqüentemente o valor de  $\gamma$  é aprimorado, ou seja, o valor de  $\gamma$  é aumentado.

### 2.3.2 Algoritmo BCJR Aplicado ao Bloco Decodificador

Nesta seção é apresentado o algoritmo BCJR aplicado ao bloco de decodificação. Os valores de  $\alpha$  e  $\beta$  são calculados a partir das equações (2.10) e (2.11) da mesma maneira que no bloco equalizador. Para o cálculo de  $\gamma$  é utilizada a equação:

$$\gamma_n(s', s) = p(z_n | s, s') P(s | s') \quad (2.20)$$

onde  $z_n$  são os elementos do vetor  $\mathbf{Z}$  definido em (2.4). A primeira parcela de (2.20) é governada pelos símbolos  $c_n$  de saída do codificador convolucional e a segunda parcela é governada pelos bits de informação  $b_k$  na entrada. Como apresentado na Seção 2.2, o bloco decodificador tem como entrada o vetor  $\mathbf{Z}$  composto pelas probabilidades referentes aos símbolos  $c_n$ . Portanto, o valor de  $\gamma$  para o bloco decodificador é dado pela expressão:

$$\gamma_n(s', s) = P(c_1 = c_1 | \mathbf{Z}) P(c_2 = c_2 | \mathbf{Z}) P(b_k) \quad (2.21)$$

onde  $c_1$  e  $c_2$  são respectivamente o primeiro e o segundo símbolo do codificador convolucional como na Figura 2.3. O termo  $P(b_k)$  é a probabilidade a priori dos bits de informação. Esta informação nunca é disponível, portanto os bits de informação são considerados identicamente distribuídos e o valor de  $P(b_k)$  é feito igual a  $1/2$ . As probabilidades  $P(c_1 = c_1)$  e  $P(c_2 = c_2)$  são retiradas diretamente do vetor  $\mathbf{Z}$ . Este vetor é obtido através da *informação extrínseca* do bloco equalizador  $L_{ext}^E(c_n^\pi)$ :

$$P(c_n = c) \cong \frac{\exp(c L_{ext}^E(c_n))}{1 + L_{ext}^E(c_n)} \quad (2.22)$$

onde  $c \in \{0, 1\}$  para os símbolos 0 e 1 respectivamente.

## 2.4 Critérios de Parada

O número de iterações realizadas para que se obtenha um desempenho desejado na equalização turbo não é sempre o mesmo para todos os blocos de informação. Para alguns blocos, apenas algumas iterações são suficientes para que a taxa de erro apresente um valor aceitável. Para outros blocos não há um melhora-

mento na taxa de erro com o aumento de iterações [Bau97]. Isto se deve as características do equalizador, do decodificador e do canal de comunicação. Os chamados critérios de parada determinam quando o número de iterações em um bloco de informações já é suficiente. Desta forma, o uso desses critérios otimiza o processo de equalização turbo, conseqüentemente diminuindo a latência do sistema. Como exemplo, são apresentados dois critérios de parada apresentado em [Bau97]: o da observação de decisões abruptas e o da função risco.

### 2.4.1 Observando Decisões Abruptas

Este critério consiste em observar, em cada iteração, a saída abrupta (*hard*) dos bits de informação do decodificador. Quando a saída não se alterar ao longo das iterações, o equalizador pára com o processo iterativo. Este critério apesar de alcançar bons resultados, torna necessário que se faça uma iteração a mais para que se observe que a saída não foi alterada.

### 2.4.2 Função Risco

Parte-se do princípio que as LLR's medem a confiabilidade das decisões. Este critério se baseia em observar o valor das LLR's dos bits de informação em cada iteração. Quando as LLR's estiverem com valores altos, o risco da decisão tomada se torna pequeno e o processo iterativo se encerra.

Calculando as probabilidades  $P(b_k|\mathbf{r})$  a partir dos LLR's dos bits  $b_k$ , a função risco é dada por:

$$\mathbf{R} = \frac{1}{L} \left[ \sum_{\substack{k=1 \\ L(u_k > 0)}} P(b_k = -1|\mathbf{r}) + \sum_{\substack{k=1 \\ L(u_k < 0)}} P(b_k = -1|\mathbf{r}) \right] \quad (2.23)$$

onde  $L$  é o tamanho do bloco de informação. A função  $\mathbf{R}$  deve estar abaixo de um determinado limiar para se obter um taxa de erro aceitável. Em [Bau97] é dado um valor prático para este limiar de  $5 \times 10^{-5}$ .



## 2.5 Sumário e Comentários Adicionais

Neste capítulo foi mostrado que a equalização turbo realiza a equalização do canal e a decodificação de maneira iterativa. O equalizador turbo é composto de um bloco equalizador e de um bloco decodificador, ambos com saídas suaves. As saídas suaves do bloco equalizador e do bloco decodificador são dadas na forma de LLR's e possuem a informação da confiabilidade das decisões tomadas. Para evitar uma redundância na realimentação de informação durante as iterações, deve-se subtrair as LLR's da saída de um bloco do equalizador turbo (equalizador ou decodificador) pela LLR da entrada.

Têm-se reportado excelentes resultados na literatura com uso de equalizadores turbo que utilizam o algoritmo BCJR no bloco de equalização e no bloco de decodificação [Bau97][Dou95][Koe03]. Para esta configuração, o equalizador turbo praticamente elimina toda IIS do canal. Isto indica que o desempenho do equalizador turbo possui um limitante dado por um sistema codificado em um canal com a presença de ruído AWGN, sem a presença de IIS. Ou seja, o desempenho de um equalizador turbo se aproxima do desempenho de um sistema de comunicação em um canal AWGN utilizando a mesma codificação utilizada no bloco de decodificação do equalizador turbo. Nos próximos capítulos o bloco decodificador será trocado por um decodificador turbo de bloco, aumentando o desempenho do equalizador turbo e diminuindo sua complexidade.

## Capítulo 3

# Equalização Turbo Utilizando Códigos de Paridade Simples Multidimensionais

Neste capítulo é proposta uma nova estrutura de Equalização Turbo aplicando Códigos Produto de Paridade Simples Multidimensionais (SPC-*Single Parity Product Codes*) [Ran01]. Primeiramente, na Seção 3.1 são introduzidos os códigos produtos de paridade simples e sua decodificação turbo. Na Seção 3.2 os códigos turbo são aplicados em um esquema de equalização turbo e decodificação turbo. São apresentadas duas configurações: a primeira é realizada uma iteração turbo completa no decodificador para cada iteração do equalizador turbo, como em [Rap97], [Yea02] e [Noo03]. Na segunda configuração é proposto um novo esquema conjunto de equalização turbo e decodificação turbo, onde apenas uma dimensão do código produto é decodificada a cada iteração do equalizador turbo. As duas configurações são comparadas em relação a desempenho e complexidade.

### 3.1 Códigos Produto de Paridade Simples

Os códigos turbo foram propostos por Claude Berrou et al. em 1993 [Ber93], onde se utilizava dois códigos convolucionais recursivos como códigos componentes. Os códigos componentes eram concatenados paralelamente via um entrelaçador tem-

poral. O algoritmo de decodificação utilizado nos decodificadores componentes era o BCJR, apresentado no Capítulo anterior, mas com ligeiras modificações. A decodificação era realizada iterativamente de maneira similar ao equalizador turbo apresentado no Capítulo 2: o algoritmo BCJR calcula a *informação a posteriori*, de onde se extrai a *informação extrínseca* que é utilizada como *informação a priori* para o outro decodificador componente.

Os resultados alcançados se aproximaram de apenas 0,7 dB do *limite de Shannon*. No entanto, para alcançar este resultado foi utilizado um bloco de entrada de 65.536 bits com taxa de codificação igual a 1/2. Desde então, têm-se investido grande esforço em novos esquemas de codificação turbo.

O primeiro esquema de decodificação turbo utilizando códigos de bloco foi apresentado em [Pyn98]. Os códigos turbo utilizando códigos de bloco podem apresentar vantagens em relação a códigos convolucionais, tais como: redução de complexidade, aumento de velocidade de decodificação e melhor desempenho para códigos de taxa alta [Gui05]. Em [Ran01] são utilizados códigos de paridade simples como códigos componentes de um código turbo, gerando os chamados códigos produto de paridade simples com decodificação turbo (SPC-TPC, - *Single Parity Check Turbo Product Codes*). A grande vantagem dos códigos SPC-TPC é a sua simplicidade de decodificação, o que justifica a escolha de tais códigos no sistema de equalização e decodificação turbo proposto nesta dissertação.

### 3.1.1 Formação dos Códigos Produto

Um código produto multidimensional pode ser construído de maneira simples, como apresentado em [Ran01]. Os dados a serem transmitidos são arranjados em uma estrutura de dimensão  $d$ , onde cada dimensão da estrutura tem o comprimento definido por  $\{k_1, k_2, \dots, k_q, \dots, k_d\}$ . Em cada dimensão  $q$  os dados são codificados utilizando códigos componentes  $(n_q, k_q, h_q)$ , onde  $n$  é o tamanho da palavra código e  $h$  a distância mínima de Hamming do código na dimensão  $q$ .

O código produto resultante apresenta uma maior capacidade de correção devido principalmente ao aumento da distância mínima. Possui também maior flexibilidade em relação à taxa de codificação e aos tamanhos de bloco de entrada e de saída do codificador [Ran01][Gui05]. Tais características podem ser identificadas abaixo:

- tamanho do bloco de entrada do codificador:

$$K = \prod_{q=1}^d k_q \quad (3.1)$$

- tamanho do bloco de saída do codificador:

$$v = \prod_{q=1}^d n_q \quad (3.2)$$

- taxa de codificação:

$$R = \prod_{q=1}^d r_q \quad (3.3)$$

onde  $r_q = n_q/k_q$ .

- distância mínima:

$$\delta_{min} = \prod_{q=1}^d h_q \quad (3.4)$$

Os códigos produto de interesse para esta dissertação possuem como códigos componentes os códigos de paridade simples,  $(n, n - 1, 2)$ , idênticos para todas as dimensões. Substituindo os valores dos comprimentos de bloco de entrada e saída e da distância mínima dos códigos de paridade simples em (3.1), (3.2), (3.3) e (3.4), têm-se:

- tamanho do bloco de entrada do codificador:

$$K = k^d = (n - 1)^d \quad (3.5)$$

- tamanho do bloco de saída do codificador:

$$v = n^d \quad (3.6)$$

- taxa de codificação:

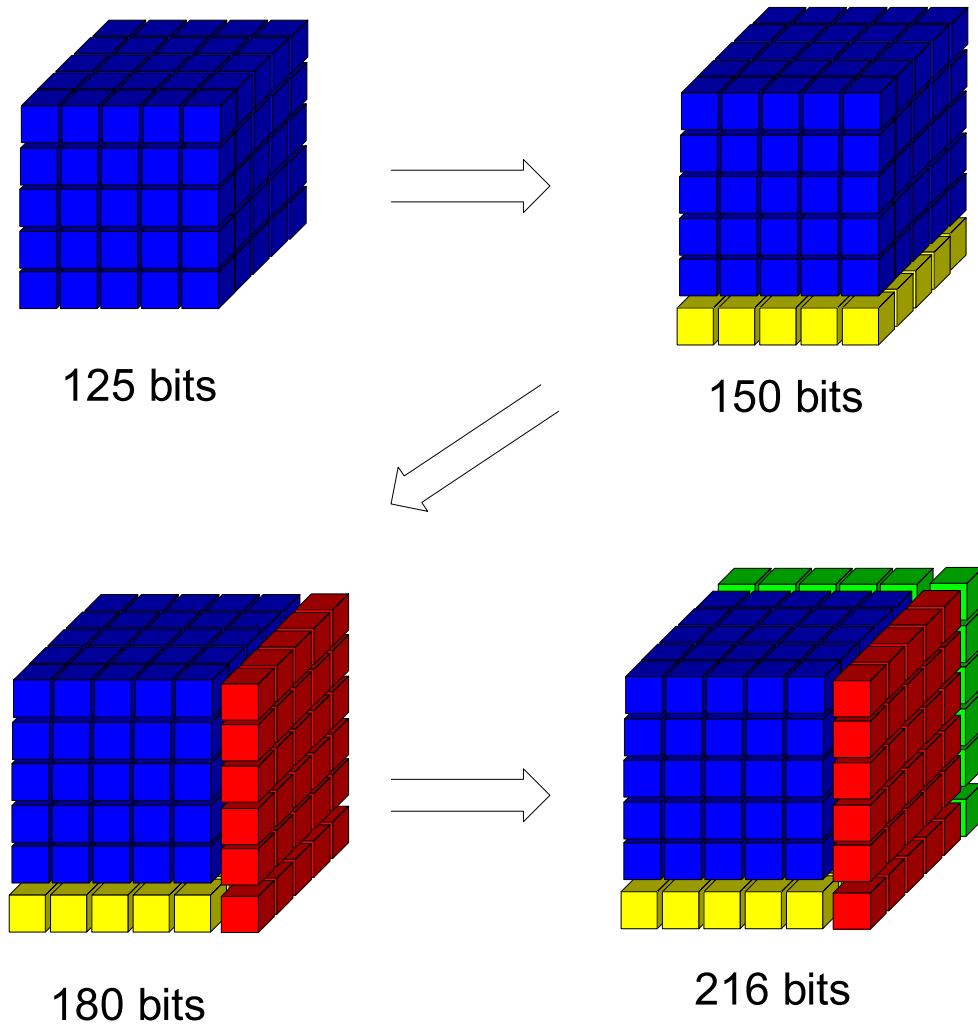
$$R = \left( \frac{n}{n-1} \right)^d \quad (3.7)$$

- distância mínima:

$$\delta_{min} = 2^d \quad (3.8)$$

O processo de formação dos código SPC consiste em adicionar bits de paridade simples em cada dimensão do código. Na Figura 3.1 é apresentado um exemplo da formação de um Código Produto de Paridade Simples. Trata-se de um código de três dimensões formado por códigos componentes  $(6, 5, 2)$  denotado por  $(6, 5, 2)^3$ . Primeiramente são agrupados 125 bits de informação em um cubo de  $5 \times 5 \times 5$  bits (largura  $\times$  altura  $\times$  profundidade). Depois calcula-se as paridades no sentido das colunas, gerando um cubo de dimensões  $5 \times 6 \times 5$  composto por 150 bits. O próximo passo consiste em calcular as paridades no sentido das linhas, o que resulta em um cubo de dimensões  $6 \times 6 \times 5$ , de 180 bits. Finalmente são calculadas as paridades no sentido da profundidade, gerando o código SPC  $(6, 5, 2)^3$ , formado por um cubo de dimensões  $6 \times 6 \times 6$  com 216 bits. O código resultante  $(6, 5, 2)^3$  terá sua distância mínima  $\delta_{min} = 8$  e taxa  $R = 0,58$ . Ao aumentar o número de dimensões do código, há uma melhoria em relação à capacidade de correção do código devido ao aumento da distância mínima do código resultante. O arranjo também possibilita o uso de blocos de entrada e saída maiores.

Uma propriedade importante para o entendimento do sistema de equalização turbo proposto mais adiante é de que todas as dimensões do código produto resultante são compostas por palavras códigos dos códigos componentes [Gui03]. Por exemplo, ao analisar o código produto  $(6, 5, 2)^3$ , todas as palavras formadas



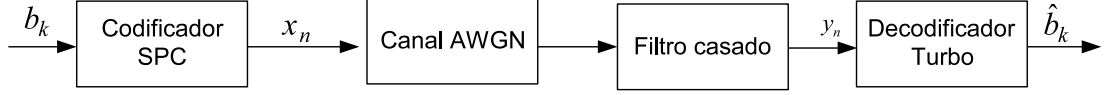
**Figura 3.1:** Formação do código  $(6, 5, 2)^3$ . Os bits em azul representam a informação, os bits amarelos as paridades das colunas, os bits vermelhos as paridades das linhas e os bits verdes as paridades da profundidade.

no sentido das colunas, das linhas e na profundidade são palavras código do código  $(6, 5, 2)$ .

### 3.1.2 Decodificação Turbo dos Códigos Produto

Para o entendimento da decodificação turbo dos códigos SPC será utilizado o sistema de comunicação ilustrado na Figura 3.2. Os bits de informação  $b_k$  são codificados por um codificador, SPC gerando símbolos  $x_n$ . O sinal modulado em BPSK é transmitido através de um canal AWGN utilizando modulação BPSK.

O sinal recebido é processado por um filtro casado e os bits de informação são estimados pelo decodificador turbo.



**Figura 3.2:** Sistema de comunicação utilizando decodificação turbo de códigos SPC.

Sendo os símbolos  $x_n$  de energia  $E$  e  $y_n$  o sinal observado na saída do filtro casado de recepção no instante de tempo discreto  $n$ , a LLR de entrada do decodificador SISO é definida por:

$$L'(\hat{x}_n) = \ln \frac{P(x_n = +\sqrt{E} | y_n)}{P(x_n = -\sqrt{E} | y_n)} \quad (3.9)$$

Aplicando o *teorema de Bayes* em (3.9), tem-se:

$$L'(\hat{x}_n) = \ln \frac{p(y_n | x_n = +\sqrt{E})}{p(y_n | x_n = -\sqrt{E})} + \ln \frac{P(x_n = +\sqrt{E})}{P(x_n = -\sqrt{E})} = L_C^D(y_n) + L_a^D(x_n) \quad (3.10)$$

onde  $L_C^D(y_n)$  é chamada de *informação de estado de canal*, que é uma quantização da confiabilidade da medida tomada pelo filtro casado na recepção [Gui05] e  $L_a^D(x_n)$  é a LLR do valor *a priori* dos símbolos  $x_n$ .

Pode-se demonstrar [Ber93] que para um código sistemático, a saída de um decodificador SISO é formada da seguinte maneira:

$$L^D(\hat{x}_n) = L'(\hat{x}_n) + L_{ext}^D(\hat{x}_n) \quad (3.11)$$

onde  $L_{ext}^D(\hat{x}_n)$  é a *informação extrínseca* do símbolo  $x_n$  na saída do decodificador turbo.

Substituindo (3.10) em (3.11), tem-se:

$$L^D(\hat{x}_n) = L_C^D(y_n) + L_a^D(x_n) + L_{ext}^D(\hat{x}_n) \quad (3.12)$$

Como a transmissão é realizada em um canal AWGN, o valor de  $L_C^D(y_n)$  é a LLR das funções densidades de probabilidade com distribuição gaussiana de média  $+\sqrt{E}$  e  $-\sqrt{E}$ , respectivamente, e variância  $\sigma^2$  dada pela potência do ruído na saída do filtro casado. Portanto, o valor de  $L_C^D(y_n)$  pode ser escrito como:

$$\begin{aligned} L_C^D(y_n) &= \ln \frac{p(y_n | x_n = +\sqrt{E})}{p(y_n | x_n = -\sqrt{E})} = \ln \left\{ \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{1}{2} \left( \frac{x_n - \sqrt{E}}{\sigma} \right)^2 \right]}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{1}{2} \left( \frac{x_n + \sqrt{E}}{\sigma} \right)^2 \right]} \right\} \\ &= -\frac{1}{2} \left( \frac{x_n - \sqrt{E}}{\sigma} \right)^2 + \frac{1}{2} \left( \frac{x_n + \sqrt{E}}{\sigma} \right)^2 = \frac{2}{\sigma^2} x_n \end{aligned} \quad (3.13)$$

Combinando (3.13) e (3.12), a saída do decodificador turbo pode ser escrita como:

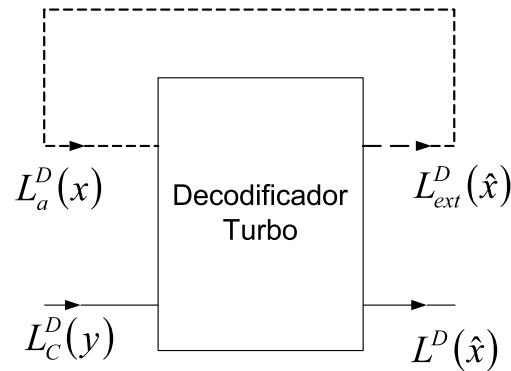
$$L^D(\hat{x}_n) = \frac{2}{\sigma^2} x_n + L_a^D(x_n) + L_{ext}^D(\hat{x}_n) \quad (3.14)$$

Com base em (3.10) e (3.12), o decodificador turbo pode ser representado por duas entradas e duas saídas como na Figura 3.3. Uma entrada é a *informação de estado de canal*,  $L_C^D(y_n)$ , que para um canal AWGN pode ser obtida por (3.13). A outra entrada é a *informação a priori* sobre o símbolo. Uma saída do decodificador SISO é a *informação extrínseca* e a outra é o *informação a posteriori* do símbolo  $x_n$ .

Na primeira iteração do decodificador turbo não há *informação a priori* disponível. Portanto,  $L_a^D(x_n)$  é feita igual a zero (probabilidades *a priori* igual a 1/2). Nas próximas iterações do decodificador turbo, o valor da *informação extrínseca* é usado como uma estimativa da *informação a priori*. Desta maneira, a cada iteração o valor da *informação extrínseca* é melhorado possibilitando que o decodificador tome decisões mais confiáveis.

Um decodificador turbo para códigos SPC-TPC é composto por  $d$  decodificadores SISO como na Figura 3.4. Cada decodificador SISO decodifica um código





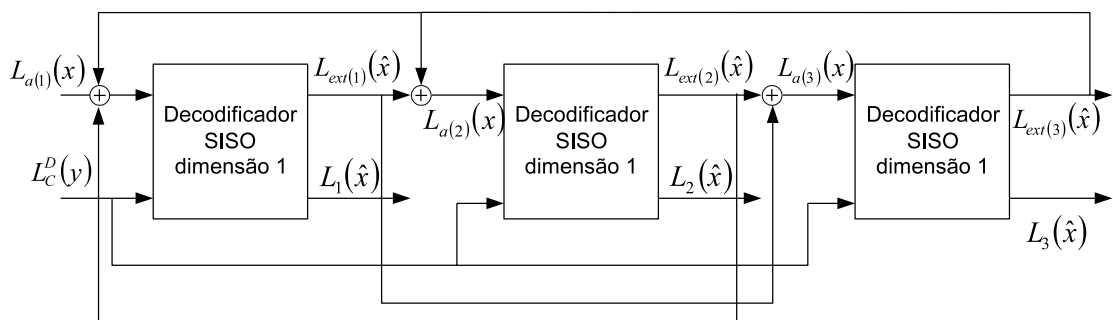
**Figura 3.3:** Entradas e saídas de um decodificador turbo.

componente (uma dimensão) do código SPC-TPC.

A Equação (3.12) pode ser também definida para um decodificador SISO de uma determinada dimensão  $q$  do código produto:

$$L_q(\hat{x}_n) = L_C^D(y_n) + L_{a(q)}(x_n) + L_{ext(q)}(\hat{x}_n) \quad (3.15)$$

onde  $L_q(\hat{x}_n)$ ,  $L_{a(q)}(x_n)$  e  $L_{ext(q)}(\hat{x}_n)$  são, respectivamente, a *informação a posteriori*, a *informação a priori* e a *informação extrínseca* do decodificador SISO na dimensão  $q$ . A informação de estado canal,  $L_C^D(y_n)$ , é a mesma para todas as dimensões.



**Figura 3.4:** Troca de informações em um decodificador turbo de um código SPC-TPC de três dimensões (adaptada de [Sam04]).

A *informação extrínseca*,  $L_{ext(q)}$ , de um determinado bit na dimensão  $q$  em

um sistema pode ser calculada por\* [Ran01]:

$$L_{ext(q)}(\hat{x}_n) = (-1)^{n_q} \times 2 \times \arctan \left[ \prod_{\substack{j=1 \\ j \neq n}}^{n_q} \tanh \left( \frac{L_C^D(y_j) + L_{a(q)}(x_j)}{2} \right) \right] \quad (3.16)$$

onde  $n_q$  é o tamanho é o número de bits codificados pela dimensão  $q$ . O termo  $L_C^D(y_j)$  é a *informação de canal* do bit de posição  $j$  e  $L_{a(q)}(x_j)$  é a *informação a priori* do símbolo na posição  $j$  e na dimensão  $q$ .

A *informação a priori*,  $L_{a(q)}(x)$ , do decodificador SISO em uma determinada dimensão é dada pela soma das informações extrínsecas das demais dimensões [Ran01]:

$$L_{a(q)}(x) = \sum_{\substack{i=1 \\ i \neq q}}^d L_{ext(i)}(\hat{x}_n) \quad (3.17)$$

A decisão abrupta sobre os bits de informação é realizada através da LLR de saída do decodificador SISO da última dimensão na última iteração do decodificador turbo:

$$L_{saida}^D = L_C^D(y) + \sum_{i=1}^q L_{ext(i)}(\hat{x}) \quad (3.18)$$

O algoritmo apresentado na tabela 3.1 descreve o processo de decodificação descrito em [Ran01].

Da mesma maneira que em um equalizador turbo, a cada iteração do decodificador turbo o valor da *informação extrínseca* se torna mais confiável para ser usado como estimativa da *informação a priori*, aumentando a confiabilidade das decisões tomadas sobre os bits de informação ou sobre os bits codificados. É importante notar que no processo de decodificação a *informação a priori* de uma determinada dimensão  $q$  é composta pelas *informações extrínsecas* das demais dimensões, como descrito na Equação (3.17). Isto é feito para que não

---

\*o termo  $(-1)^{n_q}$  não aparece em [Ran01] que utiliza um mapeamento do tipo  $\{0, 1\} \rightarrow \{+1, -1\}$ . Este termo é introduzido em [Sam04] onde se utiliza um mapeamento do tipo  $\{0, 1\} \rightarrow \{-1, +1\}$ .

---



---

(I) *Inicialização*: Calcular o valor da LLR de canal através de (3.13) e fazer a informação extrínseca de todas as dimensões igual a zero.

---



---

(II) *Decodificar todas as dimensões*: Calcular a *informação extrínseca*,  $L_{ext(q)}$ , através de (3.16) de todos os bits na primeira dimensão,  $q = 1$ . Utilizando a *informação extrínseca* das outras dimensões, calcular a *informação a priori* através de (3.17) de cada bit para a segunda dimensão. Repetir este procedimento para todas as outras dimensões do código.

---



---

(III) *Iterações*: Uma iteração do decodificador está completa ao se decodificar todas as dimensões. Repetir estas iterações quantas vezes necessário.

---



---

(IV) *Tomada de decisões*: As decisões pelos bits  $b_k$  são tomadas após a ultima iteração do decodificador turbo. Como o código é sistemático, é realizada uma decisão abrupta em relação à LLR de saída definida em (3.18), nas posições dos bits de informação.

---



---

**Tabela 3.1:** Algoritmo de decodificação turbo para códigos SPC.

haja correlação entre a *informação a priori* e a *informação extrínseca*. Outra característica importante na decodificação turbo de códigos SPC-TPC é que, na maioria das vezes, com  $d + 1$  iterações há convergência no processo de decodificação turbo. Isto quer dizer que não há uma diminuição da taxa de erro de bit (BER, do Inglês- Bit Error Rate) ao se realizar mais iterações.

## 3.2 Equalização Turbo com Decodificação Turbo Utilizando Códigos SPC-TPC

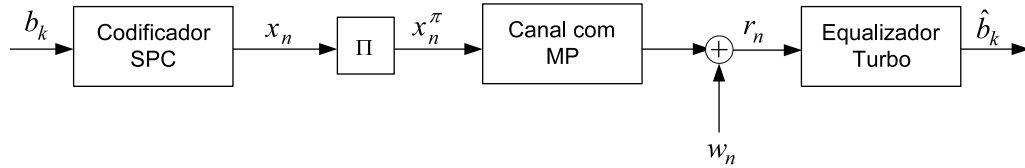
Como visto no Capítulo 2, os primeiros equalizadores turbo utilizavam-se de decodificadores convolucionais no bloco de decodificação [Dou95][Bau97]. Em [Rap97] o bloco decodificador é trocado por um decodificador convolucional turbo. Desta maneira, o equalizador turbo passa a se beneficiar não só das informações trocadas entre bloco equalizador e bloco decodificador, mas também das informações trocadas pelos decodificadores componentes que compõem o bloco decodificador

turbo. Portanto, ao inserir um decodificador turbo no bloco de decodificação, passa-se a utilizar conjuntamente os benefícios da equalização turbo e da decodificação turbo. Tendo em vista as características de códigos turbo de bloco em relação à taxa de codificação e tamanho do bloco de entrada e de saída do codificador, já citadas anteriormente, pode-se também aplicar tais códigos na equalização turbo. Em [Yea02] e [Noo03] é utilizado um decodificador turbo de bloco tendo como códigos componentes códigos BCH. O equalizador turbo apresentado em [Noo03] é composto de um DFE (*Decision Feedback Equalizer*) e de um decodificador turbo de bloco que utiliza o algoritmo descrito em [Pyn98]. Em [Yea02] é utilizado um bloco equalizador Log-MAP (BCJR no domínio das log-verossimilhanças) e são comparados os desempenhos do bloco decodificador utilizando os códigos convolucional, convolucional turbo e bloco turbo. Nas configurações apresentadas em [Rap97][Yea02][Noo03], utilizando decodificadores turbo no bloco de decodificação, uma iteração completa do decodificador turbo é realizada a cada iteração do equalizador turbo. Por exemplo, para um equalizador turbo que utiliza um decodificador turbo de bloco em seu bloco de decodificação, a cada iteração do equalizador turbo, todos os códigos componentes são decodificados. Nesta dissertação é proposto um novo esquema, onde a cada iteração do equalizador turbo apenas uma dimensão de um código produto é decodificada. Desta maneira, a cada iteração do equalizador turbo opera-se com apenas um código componente (uma dimensão), reduzindo assim a complexidade. Outra redução de complexidade introduzida neste esquema é a de utilizar códigos SPC-TPC que possuem um algoritmo de decodificação de baixa complexidade, ao contrário dos algoritmos de decodificação de códigos turbo BCH [Pyn98] e de códigos turbo convolucional [Ber93]. O equalizador turbo proposto é chamado de *equalizador turbo com decodificação parcial SPC-TPC*

### 3.2.1 Equalização Turbo Utilizando Códigos SPC-TPC

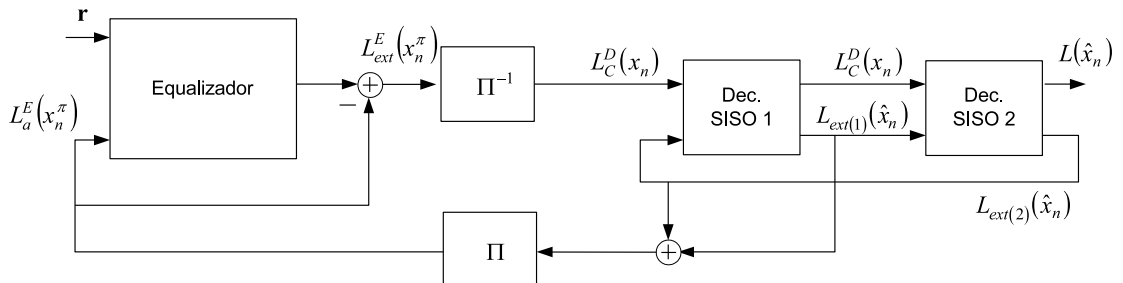
Antes de apresentar o esquema proposto, onde o bloco decodificador é um decodificador de apenas uma dimensão do código produto, será apresentado o esquema de equalização turbo conjunta com decodificação turbo, como em [Yea02] e [Noo03], adaptadas aqui para códigos SPC-TPC.

O sistema de comunicação utilizado é o da Figura 3.5. Os bits de informação  $b_k$  são codificados por um codificador SPC-TPC gerando os símbolos  $x_n$ . Os símbolos  $x_n^\pi$  na saída do entrelaçador são transmitidos por um canal com Múltiplos Percursos (MP) com adição de ruído AWGN,  $w_n$ . Através do sinal recebido,  $r_n$ , os bits de informação são estimados pelo equalizador turbo.



**Figura 3.5:** Sistema de comunicação utilizando equalização turbo.

O equalizador turbo é composto por um bloco de equalização e um bloco de decodificação Turbo. A Figura 3.6 exemplifica um sistema de equalização e decodificação turbo conjunta para um código SPC de duas dimensões.



**Figura 3.6:** Equalização turbo conjunta com decodificação turbo.

Nesta dissertação, o bloco de equalização é um equalizador BCJR para um

canal com resposta conhecida e invariante ao longo do tempo. O equalizador BCJR foi utilizado devido a sua optimabilidade. Desta maneira, o desempenho deste sistema passa ser um limitante do desempenho em relação a outro sistema que utilize outro bloco equalizador. O bloco decodificador é um decodificador turbo que utiliza o algoritmo de Rankin apresentado na Tabela 3.1. É realizada uma iteração no decodificador para cada iteração do equalizador turbo.

A troca de informações entre os blocos equalizador e decodificador é similar ao equalizador turbo convencional apresentado no Capítulo 2. A partir do vetor de amostras recebidas do canal com MP,  $\mathbf{r}$ , e da informação a priori  $L_a^E(x_n^\pi)$ , o bloco equalizador calcula a *informação a posteriori* dos símbolos  $x_n^\pi$ . Como na primeira iteração do equalizador turbo não há *informação a priori* disponível para o bloco equalizador, seu valor é inicializado como 0 (probabilidades a priori iguais a 1/2). Para descorrelacionar as informações trocadas entre os blocos equalizador e decodificador, a saída do bloco equalizador é subtraída de sua entrada *a priori*, transmitindo apenas a LLR  $L_{ext}^E(x_n^\pi)$  para o decodificador.

Neste ponto é necessário um comentário sobre a nomenclatura do valor  $L_{ext}^E(x_n^\pi)$ . Em algumas referências, como em [Han02] e [Yea02], a saída  $L_{ext}^E(x_n^\pi)$  é tratada como *informação extrínseca combinada com informação de canal* (*combined channel and extrinsic information*), ao contrário de outras referências [Koe03] e [Bau98], onde  $L_{ext}^E(x_n^\pi)$  é chamada apenas de *informação extrínseca*. A *informação extrínseca* definida em [Ber93] é a *informação a posteriori* de um componente SISO menos as entradas *informação de canal* e *informação a priori*. No caso do bloco equalizador, o valor da *informação de canal* em sua entrada não é calculado. Isto se deve ao fato do vetor de entrada  $\mathbf{r}$  ser resultado não só da contaminação do ruído AWGN, mas também ser distorcido pela IIS presente no canal. Sendo assim, o cálculo da informação de canal não está evidente como na entrada de um decodificador turbo como na Equação (3.13). Portanto, a *informação a posteriori* na saída do bloco equalizador é subtraída apenas da *informação a priori* em

sua entrada. Fica claro que a saída  $L_{ext}^E(x_n^\pi)$  possui uma parcela de *informação de canal*, justificando a nomenclatura *informação extrínseca combinada com informação de canal*. Por outro lado, o fato do valor de  $L_{ext}^E(x_n^\pi)$  ser um ganho obtido pelo processo de equalização justifica que esta LLR seja também chamada apenas de *informação extrínseca*. Por simplicidade, nesta dissertação é adotada a nomenclatura *informação extrínseca* para o valor de  $L_{ext}^E(x_n^\pi)$ , como foi feito no Capítulo 2.

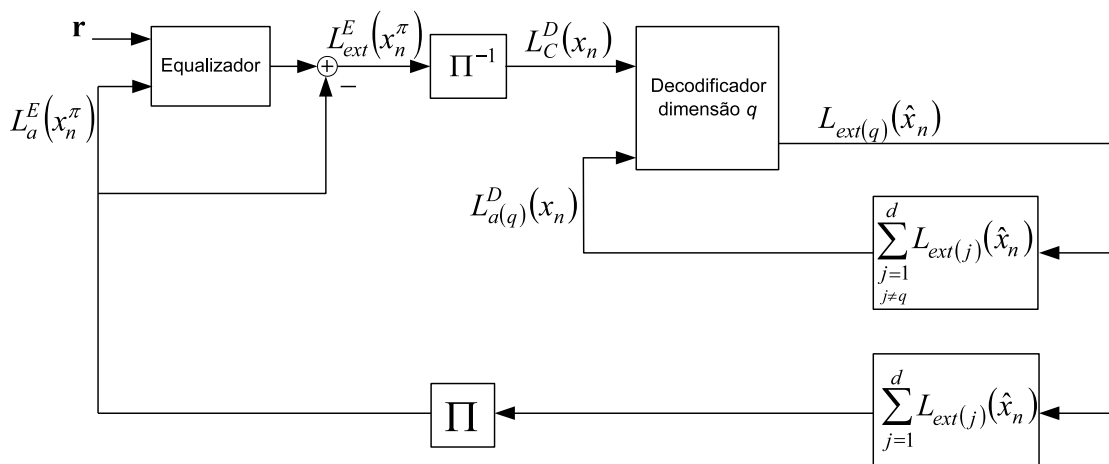
O valor da *informação extrínseca* na saída do equalizador é reordenado pelo desentrelaçador e utilizado como estimativa da *informação de canal* na entrada do bloco decodificador, como em [Rap97], [Yea02] e [Noo03]. Os comentários feitos no Capítulo 2 em relação ao papel do entrelaçador são válidos aqui. O entrelaçador que obteve um melhor desempenho para o sistema proposto é o S-*random* [Dol95]. O entrelaçador S-*random* garante uma separação mínima de S posições entre bits vizinhos. Desta maneira, quanto maior é o valor de S, maior é a decorrelação da sequência na saída do entrelaçador.

A partir da *informação de estado de canal* fornecida pelo bloco de equalização e da *informação a priori* do decodificador, a *informação extrínseca* de cada decodificador SISO é calculada por (3.16). O somatório das *informações extrínsecas* das duas dimensões é utilizado como *informação a priori* do bloco equalizador na próxima iteração. Somente a *informação extrínseca* do decodificador SISO 2, calculada na iteração passada do equalizador turbo, é utilizada como *informação a priori* para o decodificador SISO 1 na iteração atual do equalizador turbo. Desta maneira, a cada iteração do equalizador turbo é realizada uma iteração do decodificador turbo. Para códigos SPC com mais de duas dimensões o algoritmo é o mesmo. As *informações extrínsecas* de todas as dimensões são utilizadas como *informação a priori* do bloco equalizador, enquanto a *informação a priori* do decodificador respeita a Equação (3.17). Esta configuração apresentada é a mesma de [Noo03] e [Yea02]. No entanto, nas referências citadas utilizam-se

códigos turbo BCH com o algoritmo de decodificação de Pyndiah, cuja complexidade está fortemente associada à complexidade de implementação do algoritmo de Chase usado na decodificação suave dos códigos componentes. A simplificação proposta até aqui é a de se utilizar códigos SPC-TPC com o algoritmo simples de decodificação de Rankin.

### 3.2.2 A Proposta de Equalização Turbo com Códigos SPC-TPC

Neste ponto é apresentada a mais relevante contribuição desta dissertação. É proposto um novo esquema conjunto de equalização turbo e decodificação turbo. Na configuração apresentada na seção anterior, o bloco de decodificação é composto por um decodificador turbo completo, como em [Noo03] e [Yea02]. O sistema proposto aqui é apresentado na Figura 3.7. O bloco de decodificação é constituído de apenas um decodificador SISO de um único código componente (uma dimensão) do código SPC-TPC. Desta maneira, são necessárias  $d$  iterações do equalizador turbo para se decodificar todas as dimensões do código produto.



**Figura 3.7:** Equalizador Turbo utilizando apenas uma dimensão do código produto no bloco de decodificação.

Sendo  $\{1, 2, p, \dots, P\}$  o conjunto de iterações do equalizador turbo, a dimensão  $q$  a ser decodificada no bloco decodificador na iteração  $p$  do equalizador



turbo é:

$$q = [(p - 1), \text{mod}(d)] + 1 \quad (3.19)$$

A *informação extrínseca* da dimensão  $q$ ,  $L_{ext}(q)$ , é calculada pela Equação (3.16). O equalizador turbo armazena as *informações extrínsecas* de todas as dimensões  $q$  de suas últimas  $d$  iterações. Tendo armazenado estes valores, a informação a priori do bloco equalizador antes de passar pelo entrelaçador é calculada por:

$$L_a^E(x_n) = \sum_{i=1}^d L_{ext}(i)(\hat{x}_n) \quad (3.20)$$

De acordo com o algoritmo de Rankin apresentado na Tabela 3.1, a informação *a priori* de uma determinada dimensão deve ser descorrelacionada das decisões tomadas por esta mesma dimensão anteriormente. Portanto, de posse os valores armazenados de *informação extrínseca* do decodificador em iterações passadas do equalizador turbo, a informação a priori do bloco decodificador é dada por:

$$L_a^D(x_n) = \sum_{\substack{i=1 \\ i \neq q}}^d L_{ext}(i)(\hat{x}_n) \quad (3.21)$$

Esta configuração pode ser entendida também como um decodificador turbo que possui um equalizador SISO em sua malha de realimentação. Como o sinal recebido é distorcido pela IIS, o decodificador não possui uma *informação de canal* confiável. Sendo assim, o equalizador SISO faz uma estimação desta *informação de canal* entre cada decodificador componente. Desta maneira, o decodificador colabora com o equalizador fornecendo *informação a priori* para o mesmo, enquanto o equalizador *fornece informação de canal* para o decodificador. As decisões abruptas sobre os bits de informação são calculadas através da *informação a posteriori* na saída do bloco decodificador.

É importante notar que o fato de se utilizar apenas uma dimensão do código

SPC-TPC a cada iteração do equalizador turbo só é possível devido à propriedade de uma palavra em qualquer dimensão do código produto ser uma palavra-código válida do código componente. Portanto, parece não ser possível utilizar uma configuração similar para códigos convolucionais turbo, inviabilizando a estimação dos símbolos transmitidos com apenas um código componente convolucional.

A Tabela 3.2 apresenta, de maneira resumida, o algoritmo de equalização turbo utilizando decodificação parcial SPC-TPC.

---



---

(I) *Inicialização*: Os valores das *informações extrínsecas* do bloco equalizador e do bloco decodificador são feitos iguais a 0.

---



---

(II) *Equalizar e decodificar*:

- Calcular a *informação a priori* do equalizador através de (3.20).
  - Calcular a *informação extrínseca* do equalizador através do algoritmo BCJR, apresentado na seção 2.3.1.
  - Fazer a *informação de canal* na entrada do bloco decodificador igual a *informação extrínseca* na saída do equalizador.
  - calcular a *informação a priori* do decodificador através de (3.21).
  - calcular a *informação extrínseca* da dimensão  $q$  dada por (3.19), utilizando (3.16).
- 
- 

(III) *Iterações*: Uma iteração do equalizador turbo está completa ao se equalizar e decodificar a dimensão  $q$  atual. Armazena-se as *informações extrínsecas* das dimensões calculadas nas  $d$  iterações passadas do equalizador turbo para que se possa calcular as *informações a priori* dos blocos equalizador e decodificador na próxima iteração. Repetir estas iterações quantas vezes necessário.

---



---

(IV) *Tomada de decisões*: As decisões sobre os bits  $b_k$  são tomadas após a última iteração do Equalizador turbo. É realizada uma decisão abrupta em relação à *informação a posteriori* dos bits de informação na saída do bloco decodificador.

---



---

**Tabela 3.2:** Algoritmo da nova configuração proposta de equalização e decodificação turbo.

O número de iterações que levam o equalizador turbo à convergência depende do código utilizado, do tamanho do entrelaçador e da função de transferência do canal. Portanto, torna-se necessário o estudo de critérios de parada para o esquema sugerido.

Para que se visualize o processo de convergência do equalizador turbo, foram analisados os histogramas da Figura 3.8. A partir de experimentos no MATLAB, são traçados os histogramas da *informação a posteriori* dos bits de informação na saída do bloco decodificador. Nesta figura são apresentados os valores obtidos da iteração 1 até a iteração 9. O eixo horizontal representa os valores suaves da *informação a posteriori*. Nota-se a presença de duas distribuições: a distribuição a direita do zero se refere ao bit "1" e a distribuição a esquerda se refere ao bit "0" (LLR negativa). Quanto mais espaçadas as duas distribuições, maior é a confiabilidade das decisões tomadas. Os histogramas apresentam de maneira clara a evolução da confiabilidade destas decisões. Na primeira iteração, os centros das duas distribuições estão distantes de um valor de aproximadamente 20 e a dispersão da distribuição é grande. Nota-se também que estas distribuições se aproximam do formato de uma distribuição gaussiana. Ao longo das iterações o distanciamento das duas distribuições vai aumentando, o que vem a comprovar o aumento da confiabilidade das decisões tomadas. Na iteração 9, o espaçamento entre as duas distribuições é maior que 600. Um fenômeno que se observa é o surgimento de alguns resíduos laterais ao longo das iterações 6, 7 e 8. Este comportamento já foi observado em [Sam04] para decodificadores SPC-TPC. Tal comportamento merece ser objeto de estudos posteriores.

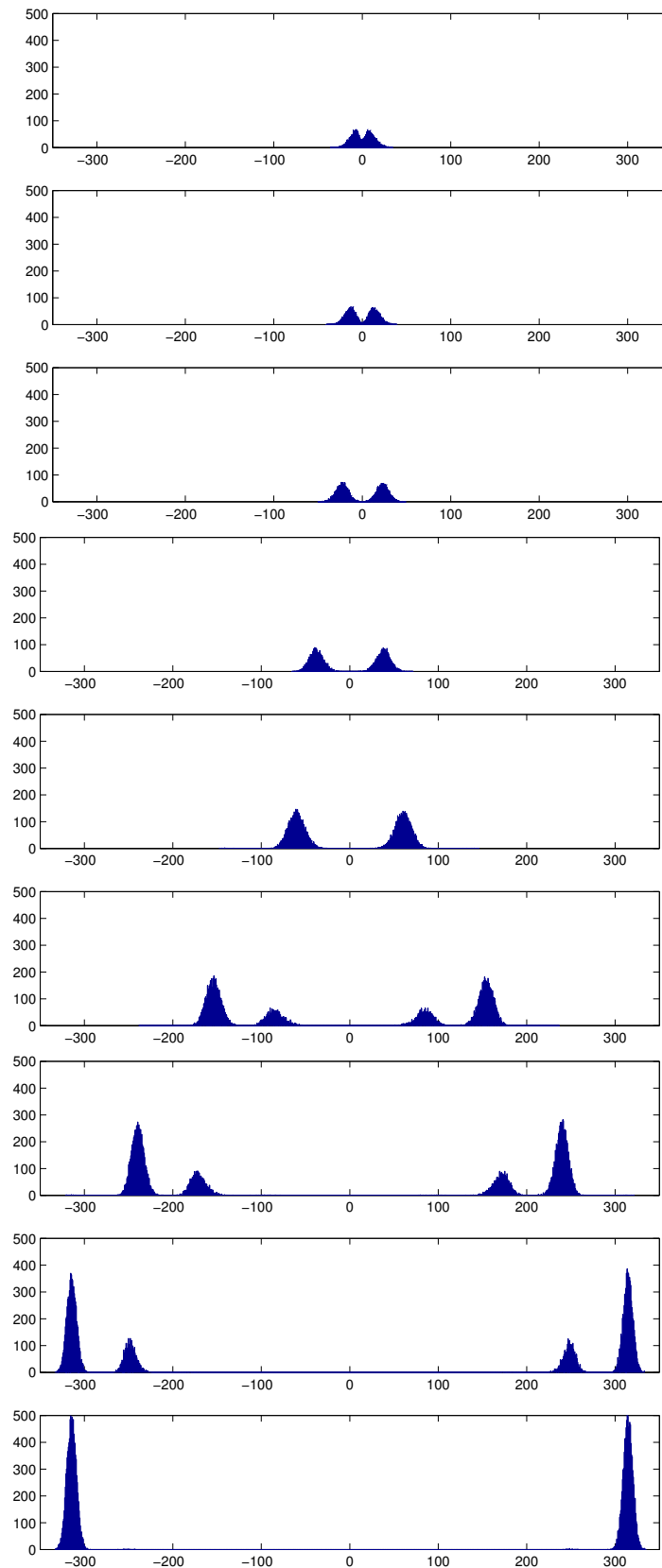
Os resultados obtidos por simulação computacional são apresentados no próximo capítulo. O desempenho do equalizador turbo proposto, onde o bloco decodificador opera em apenas uma dimensão do código produto, superou o desempenho do equalizador turbo que possui um decodificador turbo completo em seu bloco de decodificação. Além do melhor desempenho alcançado, há uma redução na com-

plexidade ao se utilizar a estrutura aqui proposta. A complexidade do sistema convencional (equalizador com decodificador turbo completo) em cada iteração do equalizador turbo é dada pela complexidade do bloco equalizador mais a complexidade do decodificador turbo. A complexidade do sistema proposto aqui é dada pela complexidade do bloco equalizador mais a complexidade de apenas um decodificador componente. Além disso, o algoritmo e decodificação dos códigos SPC-TPC é um algoritmo mais simples que os algoritmos utilizados nos blocos de decodificação de [Yea02] e [Noo03]. Para que se tenha um sistema prático onde o canal é desconhecido e se utilize modulações de alta ordem, deve-se investigar a utilização de outros blocos equalizadores, como o *Cancelador de Interferências* [Gla97] e o *Equalizador MMSE* [Tuc00][Tuc04].

### 3.3 Sumário e Comentários Adicionais

Neste capítulo foi mostrado o novo esquema de equalização turbo proposto nesta dissertação. Foi realizada uma breve descrição da formação e decodificação dos códigos SPC-TPC. Primeiramente um decodificador turbo SPC-TPC foi empregado em um equalizador turbo. A vantagem desta configuração é que o algoritmo utilizado na decodificação do código SPC-PC é mais simples do que os algoritmos dos códigos utilizados nos equalizadores turbo de [Yea02].

Foi proposto um novo equalizador turbo: o equalizador turbo de decodificação parcial SPC-TPC. O equalizador turbo em questão possui um decodificador de apenas uma dimensão de um código SPC-PC. Para cada iteração do equalizador turbo, o equalizador turbo de decodificação parcial SPC-TPC possui uma complexidade menor do que a do equalizador turbo com decodificador turbo completo SPC-TPC.



**Figura 3.8:** Histogramas da evolução da LLR dos bits de informação na saída do bloco decodificador entre a iteração 1 e a iteração 9.

# Capítulo 4

## Resultados de Simulação

Neste capítulo são apresentados os resultados obtidos através de simulações realizadas em MATLAB. O desempenho em termos de taxa de erro de bit do equalizador turbo com decodificação parcial SPC-TPC, será comparado com o desempenho do equalizador turbo com decodificador turbo SPC-TPC e com os sistemas conjuntos de equalização turbo e decodificação turbo existentes, analisados em [Yea02].

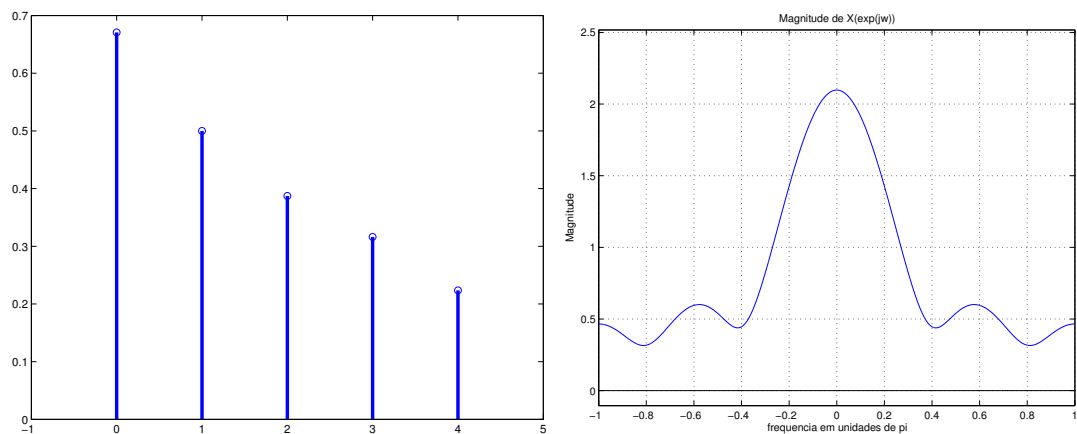
As simulações foram realizadas utilizando dois canais gaussianos com múltiplos percursos, cujos “ganhos” em cada percurso foram definidos de acordo com:

- Canal 1 [Yea02]:  $h_1 = [0,671 \ 0,5 \ 0,378 \ 0,316 \ 0,224]$
- Canal 2 [Pro95]:  $h_2 = [0,407 \ 0,815 \ 0,407]$

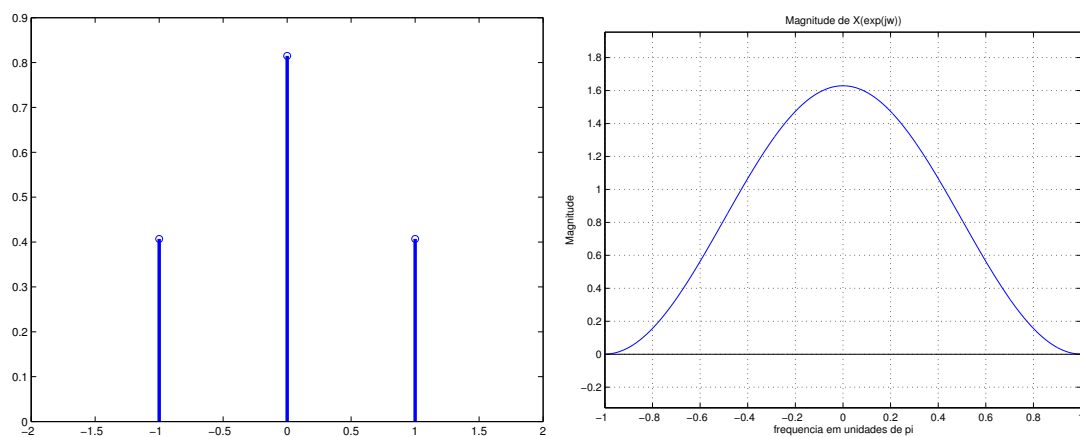
Os coeficientes dos canais estão normalizadas de maneira que a soma dos quadrados dos coeficientes seja igual a 1. Assim, não há modificação na relação sinal/ruído da saída do canal em relação à sua entrada.

As Figuras 4.1 e 4.2 apresentam os coeficientes e respostas em frequência do canal 1 e do canal 2. Analisando visualmente estas respostas, pode-se notar que trata-se de dois canais severos, onde o canal  $h_1$  é mais seletivo em frequência que o canal  $h_2$ .

Os códigos utilizados foram:



**Figura 4.1:** *coeficientes do canal 1 e sua resposta em frequência (magnitude)*



**Figura 4.2:** *coeficientes do canal 2 e sua resposta em frequência (magnitude)*

- Código  $C_1 = (6, 5)^3$ , com taxa  $R \cong 0,578$
- Código  $C_2 = (10, 9)^3$ , com taxa  $R = 0,729$
- Código  $C_3 = (17, 16)^3$ , com taxa  $R \cong 0,834$

A escolha destes códigos foi feita para que as taxas e tamanhos de bloco de entrada e saída se aproximem dos valores utilizados em [Yea02], possibilitando comparações justas.

Para as simulações realizadas foi utilizado o bloco equalizador BCJR, onde os coeficientes dos canais são conhecidos. O entrelaçador utilizado foi o *S-random*. O comprimento dos entrelaçadores e o espaçamento mínimo entre bits,  $S$ , foi escolhido de maneira que fosse possível a comparação com [Yea02]. Para se utilizar

entrelaçadores com comprimentos maiores que o tamanho do bloco de saída do codificador SPC-TPC, o equalizador turbo processava  $B$  blocos codificados. A Tabela 4.1 apresenta o comprimento e os valores de  $S$  e  $B$  dos entrelaçadores de acordo com os códigos  $C_1$ ,  $C_2$  e  $C_3$ .

código	comprimento	$S$	$B$
$C_1$	20736	128	96
$C_2$	20000	128	20
$C_3$	19652	128	4

**Tabela 4.1:** Entrelaçadores utilizados nas simulações.

## 4.1 Desempenho do Equalizador Turbo em Relação ao Número de Iterações

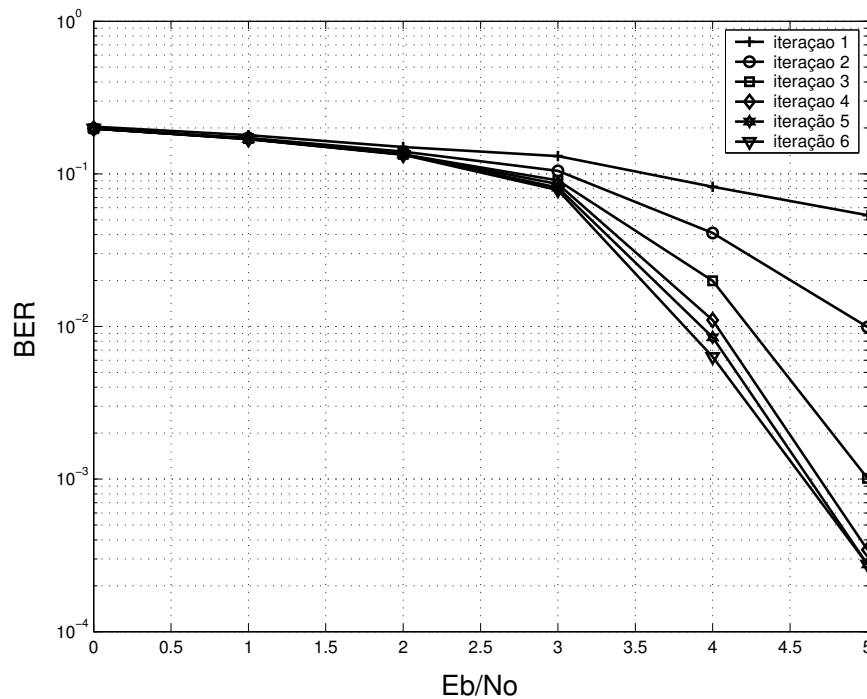
Os gráficos de taxa de erro de bit versus a relação energia de bit por densidade espectral de potência de ruído,  $E_b/N_o$ , que serão apresentados nas próximas seções só apresenta a última iteração do equalizador turbo. Para que se possa vislumbrar esta convergência nos equalizadores turbo tratados aqui, são apresentadas as Figuras 4.3 e 4.4.

O gráfico da Figura 4.3 apresenta as taxas de erro de bit a cada iteração do equalizador turbo com decodificador turbo SPC-TPC no bloco decodificador. Para a obtenção destes resultados foi utilizado o código  $C_1$  e o canal  $h_2$ . Nesta configuração o equalizador turbo obteve ganho significativo em relação à taxa de erro de bit até a iteração 6.

Para um equalizador turbo com decodificação parcial SPC-TPC foi obtido o resultado da Figura 4.4. Foi utilizado o canal  $h_1$  e o código  $C_2$ . Para esta configuração o equalizador turbo obteve ganhos até a iteração 15. Como o código é tridimensional, o equalizador turbo processou cinco vezes todas as dimensões do código.

Para as configurações de canal, código e entrelaçador utilizados nesta dissertação manteve-se o comportamento dos gráficos 4.3 e 4.4. Na média, o equali-



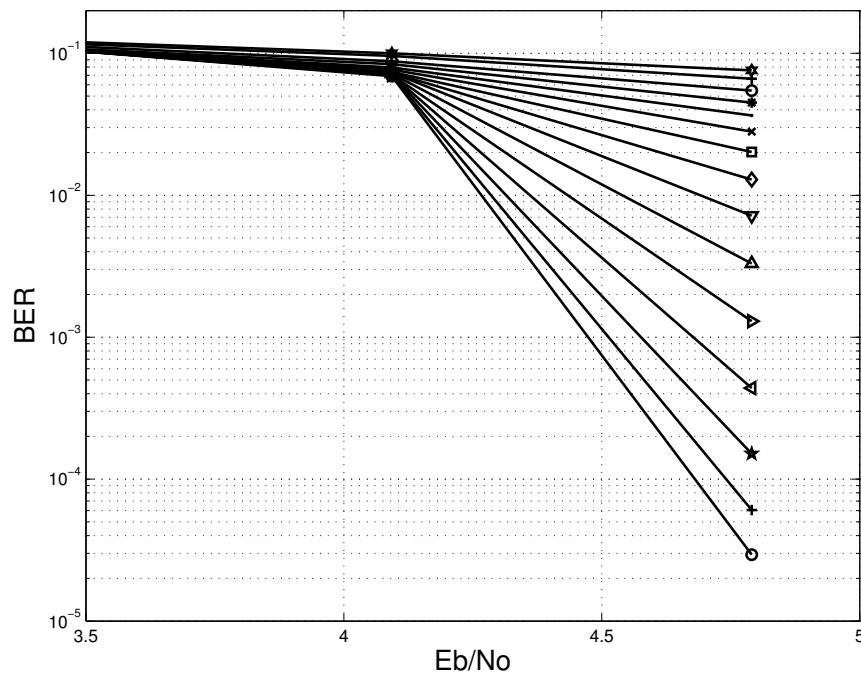


**Figura 4.3:** Convergência do equalizador turbo utilizando decodificador turbo SPC-TPC.

zador turbo com decodificador turbo obteve ganhos até a iteração 6 enquanto o equalizador turbo com decodificação parcial SPC-TPC obteve ganhos até a iteração 15. O maior número de iterações no equalizador turbo com decodificação parcial SPC-TPC permite que esta configuração se aproxime mais do desempenho do código SPC-TPC em canal AWGN, do que o equalizador turbo com decodificador turbo. No entanto, ao realizar um maior número de iterações há um aumento na latência do sistema.

## 4.2 Desempenho do Equalizador Turbo Utilizando o Código $C_1$ .

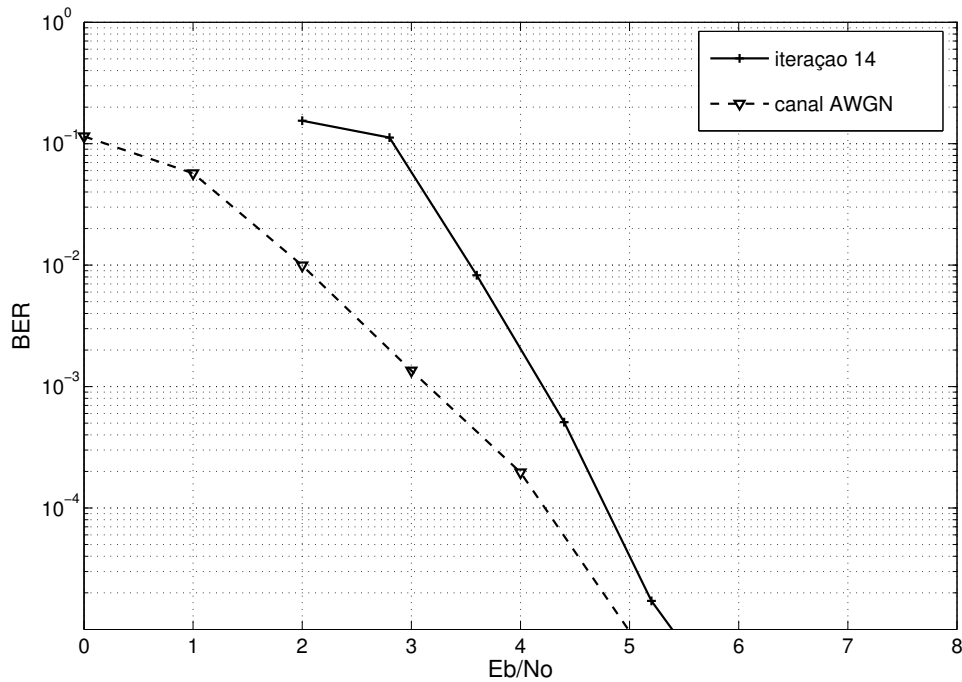
Nesta seção são apresentados e analisados os resultados obtidos utilizando o código  $C_1$ . Primeiramente foi utilizado o equalizador turbo com decodificação parcial SPC-TPC. O canal usado foi o canal  $h_1$ . Este canal foi utilizado em [Yea02], onde são comparados os equalizadores turbo utilizando códigos convolucionais, convolucionais turbo e bloco BCH turbo.



**Figura 4.4:** Convergência do equalizador turbo utilizando o bloco decodificador de apenas uma dimensão. Estão traçadas de cima para baixo as curvas relativas às iterações 1 à 15.

A Figura 4.5 apresenta os resultados obtidos pelo equalizador turbo com decodificação parcial SPC-TPC. Para valores altos de  $E_b/N_o$ , o desempenho do equalizador turbo se aproximou do desempenho do código  $C_1$  em canal AWGN. Por exemplo, para uma taxa de erro de bit de  $10^{-5}$ , o desempenho do equalizador turbo se aproximou de 0,3 dB do desempenho do código  $C_1$  em canal AWGN. Comparando com os resultados obtidos em [Yea02], o desempenho do equalizador turbo com decodificação parcial SPC-TPC foi inferior ao desempenho do equalizador turbo com decodificador turbo de bloco BCH e ao convolucional turbo e se aproximou do equalizador turbo utilizando um código convolucional de taxa 1/2, @  $10^{-5}$  de taxa de erro de bit.

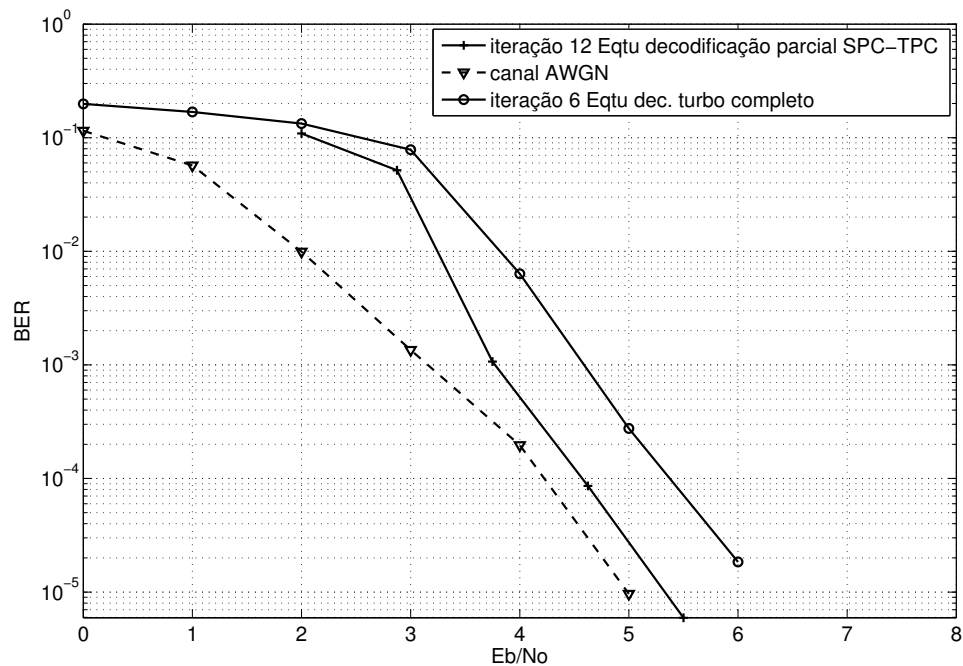
A Figura 4.6 apresenta o desempenho do equalizador turbo com decodificação parcial SPC-TPC utilizando o código  $C_1$  com o canal  $h_2$ . Como  $h_2$  é menos severo do que  $h_1$ , o desempenho do equalizador turbo em valores baixos de  $E_b/N_o$  é melhor para o sistema utilizando  $h_2$ . Por exemplo, para um valor de  $E_b/N_o$  de



**Figura 4.5:** Desempenho do equalizador turbo com decodificação parcial SPC-TPC utilizando o código  $C_1$  no canal  $h_1$ .

4 dB, ao utilizar  $h_2$  o equalizador turbo se aproximou mais do canal AWGN do que ao utilizar  $h_1$ . No entanto, para valores altos de  $E_b/N_o$  o desempenho do equalizador turbo para os canais  $h_1$  e  $h_2$  é praticamente o mesmo.

Na Figura 4.6 também é traçado o desempenho do equalizador turbo utilizando o decodificador turbo SPC-TPC no bloco de decodificação. O desempenho desta configuração foi inferior ao equalizador turbo com decodificação parcial SPC-TPC. O equalizador turbo com decodificador turbo obteve ganhos até a iteração 6, enquanto o equalizador turbo com decodificação parcial SPC-TPC até a iteração 12. O fato do equalizador turbo com decodificação parcial SPC-TPC obter ganhos com um número maior de iterações, permite que o seu desempenho se aproxime mais do desempenho do código em canal AWGN. No entanto, como já mencionado, com um maior número de iterações pode ser que haja um aumento significativo na latência do sistema.

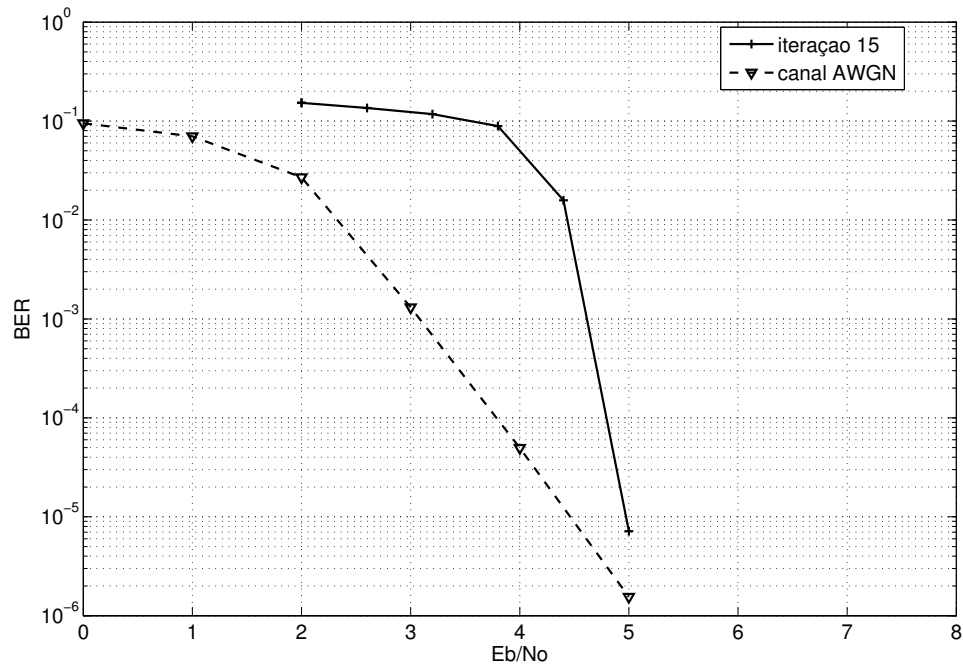


**Figura 4.6:** Desempenho do equalizador turbo com decodificação parcial SPC-TPC utilizando o código  $C_1$  no canal  $h_2$ .

### 4.3 Desempenho do Equalizador Turbo Utilizando o Código $C_2$ .

As Figuras 4.7 e 4.8 apresentam os resultados obtidos utilizando o código  $C_2$  de taxa 0,729. Primeiramente, na Figura 4.7 foi traçado o desempenho do equalizador turbo com decodificação parcial SPC-TPC utilizando o canal  $h_1$ . Para uma taxa de erro de  $10^{-5}$ , o desempenho do equalizador turbo se aproximou de 0,5 dB do desempenho do código  $C_2$  em um canal AWGN.

Diferentemente do desempenho obtido com o código  $C_1$ , o equalizador turbo com decodificação parcial SPC-TPC utilizando o código  $C_2$  superou o desempenho de todos os esquemas de equalização e decodificação turbo analisados em [Yea02]. Ou seja, para uma taxa de codificação de aproximadamente 3/4, o equalizador turbo de decodificação parcial SPC-TPC foi superior aos equalizadores turbo que utilizam código BCH turbo, convolucional turbo e convolucional em seu bloco de

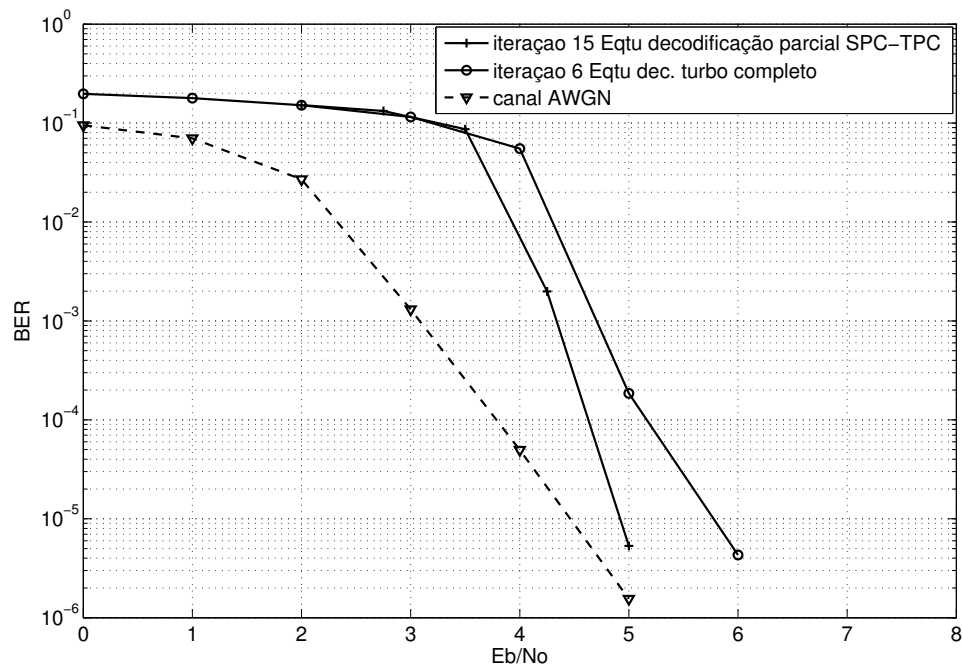


**Figura 4.7:** Desempenho do equalizador turbo com decodificação parcial SPC-TPC utilizando o código  $C_2$  no canal  $h_1$ .

decodificação, para  $10^{-5}$  de BER.

A Tabela 4.2 apresenta os ganhos do equalizador turbo com decodificação parcial SPC-TPC em relação as configurações de equalização e decodificação turbo conjuntas existentes. Na primeira coluna da tabela estão os códigos utilizados no bloco decodificador em [Yea02]. A segunda coluna apresenta os ganhos obtido pelo equalizador turbo aqui proposto em relação aos utilizados em [Yea02]. Foram contabilizados 50 bits em erro para que se estimasse a taxa de erro e os ganhos apresentados nesta tabela. A confiabilidade obtida com este número de erros contabilizados pode ser considerada alta devido ao fato de que, como a curva de taxa de erro de bit é bastante abrupta, uma certa variação na taxa de erro não provoca uma grande variação na correspondente relação  $E_b/N_0$ .

Neste ponto é importante destacar que apesar de se obter resultados melhores com o equalizador turbo com decodificação parcial SPC-TPC, os códigos utilizados em [Yea02] possuem um melhor desempenho em canal AWGN do que o



**Figura 4.8:** Desempenho do equalizador turbo com decodificação parcial SPC-TPC utilizando o código  $C_2$  no canal  $h_2$ .

código	ganho aproximado (dB)
convolucional turbo	0,2
bloco BCH turbo	0,7
convolucional	1

**Tabela 4.2:** Ganhos obtidos pelo equalizador turbo proposto em relação aos equalizadores turbo analisados em [Yea02]. A coluna da esquerda apresenta os códigos utilizados em tais equalizadores turbo. Na segunda coluna estão os ganhos obtidos pelo equalizador turbo com decodificação parcial em relação aos equalizadores turbo de taxa  $3/4$  para uma BER de  $10^{-5}$ .

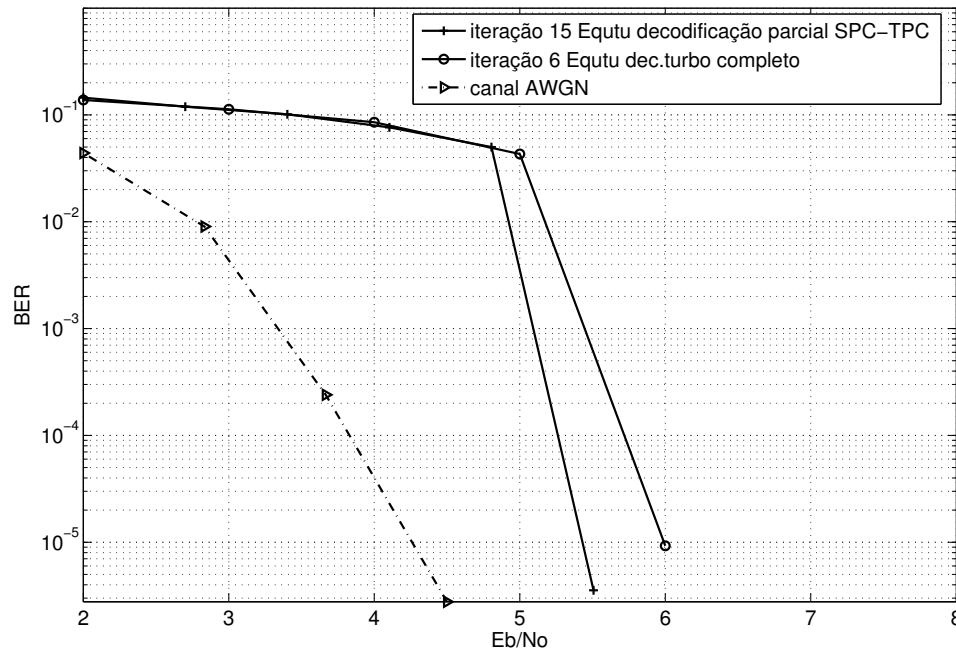
código SPC-TPC  $C_2$ . O que permite que o equalizador turbo proposto aqui alcance melhores desempenhos é a maior capacidade de redução de IIS em relação aos equalizadores analisados em [Yea02]. Por exemplo, o código BCH turbo utilizado em [Yea02] alcança uma taxa de erro de bit de  $10^{-5}$  em 3,5 dB de  $E_b/N_0$ , enquanto o código SPC-TPC  $C_2$  só alcança este desempenho em 4,5 dB. No entanto, a redução de IIS do equalizador turbo com decodificação parcial SPC-TPC é maior do que a do equalizador turbo utilizando decodificador turbo BCH.

Na Figura 4.8 são apresentados resultados com o canal  $h_2$ . O desempenho do equalizador turbo com decodificação parcial SPC-TPC é comparado com o desempenho do equalizador turbo com decodificador turbo SPC-TPC. A partir destes resultados obtidos pode-se observar que o equalizador turbo que utiliza um decodificador de dimensão única SPC-TPC se aproxima mais do desempenho do código em uma canal AWGN do que o que utiliza o decodificador turbo SPC-TPC completo em seu bloco de decodificação. É importante notar também que, quanto maior é o valor de  $E_b/N_o$ , maior é capacidade de redução de IIS do equalizador turbo com decodificação parcial SPC-TPC. O mesmo não acontece com o equalizador turbo com decodificador turbo completo SPC-TPC. A Figura 4.8 sugere que para valores acima de 6 dB de  $E_b/N_o$ , a redução de IIS é total para o equalizador turbo com decodificação parcial SPC-TPC. Esta última afirmação não pôde ser comprovada devido ao longo período de simulação que seria necessário para tal verificação.

## 4.4 Desempenho do Equalizador Turbo Utilizando o Código $C_3$ .

Até este ponto os resultados obtidos pelo equalizador turbo com decodificação parcial SPC-TPC foram comparados com os equalizadores turbo utilizados em [Yea02]. Para o código  $C_1$  ( $R \cong 1/2$ ), o resultado obtido pelo equalizador turbo com decodificação parcial SPC-TPC foi inferior aos resultados obtidos pelos equalizadores turbo utilizados em [Yea02]. No entanto, ao utilizar o código  $C_2$  ( $R \cong 3/4$ ), o equalizador turbo com decodificação parcial SPC-TPC obteve um resultado superior aos utilizados em [Yea02]. Portanto, há um indício de que o equalizador turbo proposto aqui possui um melhor desempenho para altas taxas de codificação. Para confirmar este indício, a Figura 4.9 apresenta o desempenho obtido utilizando o código  $C_3$  ( $R \cong 5/6$ ).

O desempenho obtido com o código  $C_3$  no equalizador turbo com decodificação



**Figura 4.9:** Desempenho do equalizador turbo com decodificação parcial SPC-TPC utilizando o código  $C_3$  no canal  $h_1$ .

parcial SPC-TPC foi superior a todos os equalizadores turbo de [Yea02]. A Tabela 4.3 mostra os ganhos de  $E_b/N_o$  obtidos a uma taxa de erro de bit de  $10^{-5}$  em relação aos equalizadores turbo utilizados em [Yea02].

código	ganho aproximado (dB)
convolucional turbo	0,3
bloco BCH turbo	0,8
convolucional	0,8

**Tabela 4.3:** Ganhos obtidos pelo equalizador turbo proposto em relação aos equalizadores turbo analisados em [Yea02]. A coluna da esquerda apresenta os códigos utilizados nestes equalizadores turbo. Na segunda coluna estão os ganhos obtidos pelo equalizador turbo com decodificação parcial em relação aos equalizadores turbo de [Yea02] de taxa 5/6 para uma BER de  $10^{-5}$ .

O resultado obtido com o código  $C_3$  confirma a suspeita que o equalizador turbo com decodificação parcial SPC-TPC, para altas taxas de codificação, apresenta um melhor desempenho em relação aos equalizadores turbo utilizando código convolucional, convolucional turbo e bloco BCH turbo [Yea02].



A Figura 4.9 também apresenta o desempenho do equalizador turbo utilizando o decodificador turbo completo e o desempenho do código  $C_3$  em canal AWGN. Nota-se que a distância do desempenho do equalizador turbo com decodificação parcial SPC-TPC em relação ao desempenho do código  $C_3$  é de mais de 1 dB. Ainda assim, seu desempenho é superior aos obtidos em [Yea02].

## 4.5 Sumário e Comentários Adicionais

Neste capítulo foram apresentados os resultados obtidos por simulação computacional. O equalizador turbo com decodificação parcial SPC-TPC foi comparado com o equalizador turbo com decodificador turbo SPC-TPC e os equalizadores turbo utilizados em [Yea02]. Os resultados obtidos mostram que o equalizador turbo com decodificação parcial SPC-TPC obteve um melhor desempenho em termos de BER do que o equalizador turbo com decodificador turbo SPC-TPC para todas as taxas de codificação analisadas. Para altas taxas de codificação (3/4 e 5/6), o equalizador turbo com decodificação parcial SPC-TPC obteve melhores resultados do que os equalizadores utilizados em [Yea02].

# Capítulo 5

## Itens Conclusivos

### 5.1 Sumário Geral e Conclusões

Um equalizador turbo é um receptor que realiza a equalização e a decodificação de maneira iterativa. Como um canal com múltiplos percursos pode ser modelado por uma cadeia de Markov, pode-se um equalizador BCJR. O bloco equalizador troca informações com o bloco decodificador de maneira iterativa como nos decodificadores turbo. Desta maneira, o bloco equalizador pode ser visto como um decodificador interno de um sistema de concatenação serial de códigos e o bloco de decodificação como o decodificador externo.

O Capítulo 2 fez uma apresentação sobre os primeiros equalizadores turbo, onde o código utilizado era um código convolucional [Dou95][Bau97][Koe03]. Nesta configuração, o desempenho do equalizador turbo se aproxima do desempenho do código convolucional em um canal AWGN.

Os equalizadores turbo podem se beneficiar da utilização dos decodificadores turbo em seu bloco de decodificação. O Capítulo 3 apresentou a mais relevante contribuição desta dissertação, onde é sugerida uma nova proposta para equalizadores turbo. É utilizado um decodificador de apenas uma dimensão de um código SPC-PC no bloco decodificador. O equalizador turbo de decodificação parcial SPC-TPC é comparado com o equalizador turbo que utiliza um decodificador

turbo completo no bloco de decodificação. O equalizador turbo de decodificação parcial SPC-TPC supera o que utiliza um decodificador turbo completo. Seu desempenho se aproxima do desempenho do código SPC-TPC em um canal AWGN, ou seja, o equalizador turbo praticamente eliminou a IIS. O equalizador turbo com decodificação parcial SPC-TPC foi comparado também com os equalizadores turbo analisados em [Yea02]. Na referência [Yea02], são comparados equalizadores turbo que utilizam decodificadores convolucional, turbo convolucional e bloco BCH turbo. O equalizador turbo de decodificação parcial SPC-PC apresentou um melhor desempenho em relação à taxa de erro de bit para todos estes sistemas para taxas de codificação altas ( $\geq 3/4$ ). Para taxas baixas ( $\cong 1/2$ ) os sistemas utilizados em [Yea02] superaram o equalizador turbo de decodificação parcial SPC-TPC.

O equalizador turbo de decodificação parcial SPC-TPC apresenta uma menor complexidade computacional em relação aos equalizadores turbo que utilizam decodificador turbo para uma iteração. A primeira simplificação é a de utilização de códigos SPC-TPC, que possuem um algoritmo de decodificação turbo bastante simples. Outra simplificação na implementação é que a complexidade do bloco decodificador é a complexidade de apenas um decodificador componente. Apesar da complexidade do equalizador turbo de decodificador de única dimensão ser aparentemente menor que a complexidade de um equalizador turbo conjunto com decodificador turbo em uma única iteração, há necessidade de um número grande de iterações para o equalizador turbo de decodificação parcial apresentar um desempenho satisfatório em relação à taxa de erro de bit. Portanto, há que se verificar a complexidade final do equalizador turbo ao se realizar todas as iterações. Por outro lado, o elevado número de iterações do equalizador turbo com decodificação parcial SPC-TPC permite que a redução da IIS seja quase total.

## 5.2 Principais Contribuições

O Capítulo 2 contribuiu com a introdução dos conceitos de equalização turbo. Com a abordagem utilizada pretende-se que o leitor não necessite de conhecimentos prévios de decodificação turbo para o entendimento do texto.

O Capítulo 3 é o capítulo principal desta dissertação. Suas principais contribuições são:

- A formação dos códigos SPC-TPC e sua decodificação turbo foi apresentada de uma maneira que possibilitasse a adaptação destes códigos em equalizadores turbo.
- Os códigos SPC-TPC foram adaptados para os esquemas convencionais de equalização turbo, onde o bloco decodificador é constituído de um decodificador turbo completo. Desta maneira, utiliza-se o algoritmo de Rankin cuja simplicidade é maior em relação ao algoritmo de Pyndiah utilizado em decodificadores turbo de bloco BCH.
- A maior contribuição é a proposta de um novo equalizador turbo: o equalizador turbo com decodificação parcial SPC-TPC. Trata-se de um sistema que, para taxas altas de codificação, superou o desempenho dos equalizadores turbo existentes em um canal gaussiano com múltiplos percursos. Esta configuração de equalizadores turbo alcança melhor desempenho, em taxas altas, do que equalizadores turbo que utilizam códigos mais potentes do que o SPC-TPC, como o convolucional turbo e o bloco BCH turbo. Outro atrativo deste equalizador turbo é sua baixa complexidade de implementação.
- Um conjunto de simulações de programas criados no MATLAB\* que permite o estudo de futuras modificações a serem investigadas futuramente.

---

\*Estas simulações estão em anexo à versão original desta dissertação na forma de mídia eletrônica.

## 5.3 Oportunidades para Futuras Investigações

Como o equalizador turbo com decodificação parcial SPC-TPC corresponde a de um sistema inédito, cria-se campo para um número grande de investigações a serem realizadas futuramente:

- Para se avaliar a melhor situação nesta dissertação foi utilizado o bloco equalizador BCJR no equalizador turbo. Para tornar o sistema mais atraente para implementações práticas é necessário que se investigue a utilização de *equalizadores MMSE* [Tuc00][Tuc02] e do *cancelador de interferência* [Gla97]. Ao se utilizar estes blocos equalizadores, podem ser analisados canais que sejam desconhecidos, variantes no tempo, com longa dispersão temporal onde se utilizem modulações de mais alta ordem que a BPSK. Pode ser investigado também o uso de estimadores de canal em paralelo com o bloco equalizador.
- Implementar o equalizador turbo de decodificação parcial SPC-TPC utilizando códigos produto de dimensões acima de três. Como o desempenho em relação à taxa de erro de bit dos códigos SPC-TPC aumenta com o acréscimo de dimensões [Ran01], cogita-se que haverá um aumento no desempenho do equalizador turbo utilizando mais dimensões no código.
- Utilizar outros códigos turbo de bloco no equalizador turbo de decodificação parcial. Por exemplo, investigar uma configuração onde o bloco decodificador do equalizador turbo decodifica apenas uma dimensão de um código produto BCH.
- Analisar de maneira quantitativa a complexidade de implementação e a latência dos equalizadores turbo utilizados nesta dissertação e compará-los com outras propostas citadas na literatura.
- Utilizar a ferramenta *Extrinsic Information Transfer Charts* (EXIT charts)

[Otn02] no equalizador turbo de decodificador de dimensão única para obtenção de limitantes teóricos de probabilidade de erro de bit.

- Analisar o efeito do entrelaçador temporal variando seu comprimento e o espaçamento S.

## 5.4 Publicações do Autor

- GUIMARÃES, D. A., SANTOS, A. F. *Novel Block Turbo Equalizer Scheme*. IEE Electronics Letters - em preparação para submissão.
- SANTOS, A. F., and GUIMARÃES, D. A., *Tutorial: Turbo Equalization*. First International Workshop on Telecommunications, IWT 2004, Inatel. Santa Rita do Sapucaí, MG, August, 23-27, 2004.
- SANTOS, A. F., and GUIMARÃES, D. A., *Introdução à Equalização Turbo*. Submetido à Revista Telecomunicações do INATEL.

# Referências Bibliográficas

- [Bau97] G. Bauch, H. Khorram and J. Hagenauer, *Iterative Equalization and Decoding in Mobile Communication Systems*. 2nd EPMCC'97 and 3rd ITG-Fachtagung Telecomm, Mobile Kommunikation, Bonn, Germany, Oct. 1997.
- [Bau98] G. Bauch and V. Franz, *A Comparison of Soft-in/Soft-out Algorithms for Turbo detection*. Intern. Conf. On Telecomm., pp. 259-263, June 1998.
- [Bah74] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, *Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate*. IEEE Transactions on Information Theory, vol. IT-20, pp. 284-287, March, 1974.
- [Ber93] C. Berrou, A. Glavieux and P. Thitimajshima, *Near Shannon Limit Error-Correcting and Decoding: Turbo-Codes*. International Conference on Communication (ICC), pp. 1064-1070, May 1993.
- [Bri01] S. T. Brink, *Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes*. IEEE Transactions on communications, vol. 49, n<sup>o</sup>10, October 2001.
- [Chug97] Anastasopoulos and K. Chugg, *Iterative Equalization/Decoding for TCM for Frequency-Selective Fading Channels*. Conf. Record 31th Asilomar Conf. On Signals, Systems and Comp., vol. 1, pp. 177-181, Nov 1997.
- [Dol95] S. Doullinar and D. Divsalar, *Weight distribution for turbo codes using random and nonrandom permutations*. IEEE Transactions on Communications, vol. 49, n<sup>o</sup>12, December 2001.

- [Dou95] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, *Iterative Correction of Intersymbol Interference: Turbo Equalization*. European Trans. On Telecomm., vol. 6, pp. 507-511, Sep-Oct 1995.
- [Gla97] A. Glavieux, C. Laot, and J. Labat, *Turbo Equalization Over a Frequency Selective Channel*. Proc. Of the Intern. Symposium on Turbo codes, Brest, France, pp. 96-102, September 1997.
- [Ger81] A. Gersho and T. Lim, *Adaptive cancellation of intersymbol interference for data transmission*. Bell Syst. Tech. J., vol. 60, n<sup>o</sup> 11, pp.1997-2021, Nov.1981.
- [Gui03] D. A. Guimarães, *Uma Classe de Códigos Produto e sua Decodificação Turbo Aplicada em um Sistema CDFMA Multiportadora*. Tese de Doutorado: Universidade Estadual de Campinas-Unicamp, Campinas, SP, June2003.
- [Gui05] D. A. Guimarães, *Noções Sobre Decodificação Turbo*. Artigo submetido à Revista Telecomunicações do Instituto Nacional de Telecomunicações.
- [Han02] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo equalization and Space-Time Coding for Transmission Over Fading Channels*. John Wiley and Sons, Inc 2002.
- [Hay01] S. Haykin, *Communication Systems*. 4rd edition: John Wiley and Sons, Inc, New York, 2001.
- [Hag89] J. Hagenauer and P. Hoeher, *A Viterbi Algorithm with Soft Decision Outputs and Its Applications*. IEEE Global Telecomm. Conf., pp. 1680-1686, 1989.
- [Lao01] C. Laot, A. Glavieux and J. Labat, *Turbo Equalization: Adaptive Equalization and Channel Decoding Jointly Optimized*. IEEE Journal on Selected Areas in Communications, vol.19, n<sup>o</sup>9, september,2001.



- [Lin05] M. D. Linda, B. C. Iain, and P. Hoeher, *Joint Map Equalization and Channel Estimation for Frequency-Selective and Frequency-Flat Fast Fading Channels*. The Telecommunications and Data Acquisition progress Report 42-122, pp. 56-65. Jet Propulsion Laboratory, Pasadena, USA, April- June 1995.
- [Mul81] M. S. Muller and J. Salz, *A unified theory of data-aided equalization*. Bell Syst. Tech. J., vol. 60, n<sup>o</sup> 9, pp.2023-2038, Nov.1981.
- [Noo03] M. Noorbakhsh and K. Mohamed-Pour, *Combined turbo equalization and block turbo coded modulation*. IEE Proceedings on Communications, Vol. 150, No. 3, pp 149 - 152, June 2003.
- [Otn02] R. Otnes, T. Micheal , *EXIT Chart Analysis Aplieed To Adaptative Turbo Equalization*. Nordic Signal Processing Symposium, NORSIG, 2002, Hurtigruten Tromso-Trondheim, Norway, October 4-7 2002.
- [Koe03] T. Michael, K. Ralf, A. Singer, *Turbo Equalization*. IEEE Signal Processing Magazine, Feb 2003.
- [Pro95] J. G. Proakis, *Digital Communications*. Third Edition: New York: McGraw-Hill, 1995.
- [Pyn98] R. M. Pyndiah, *Near-Optimum Decoding of Product Codes: Block Turbo Codes*. IEEE Transactions on Communications, pp. 41-46, Vol. 49, n<sup>o</sup> 8, August 1998.
- [Rap97] D. Raphaeli, Y. Zarai, *Combined Turbo Equalization And Turbo Decoding*. Proc. Global Telecommunications Conf., Phoenix , AZ, Nov. 3-8, pp. 639-641, 1997.
- [Ran01] D. M. Rankin and T. A. Gulliver, *Single Parity Check Product Codes*. IEEE Transactions on Communications, vol. 49,no 8, August 2001. Ver

- também: D. M. Rankin, *Single Parity Check Product Codes and iterative decoding*. Ph.D. Thesis, University of Canterbury, New Zeland, May 2001.
- [Sam04] A. P. M. Sâmia, *Códigos de Bloco com Decodificação Turbo Adaptados às Especificações do Padrão DVB-RCT* Dissertação de Mestrado: Instituto Nacional de Telecomunicações, INATEL, Agosto de 2004.
- [Sha48] C. E. Shannon, *A mathematical theory of communication*. Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
- [Skl97] B. Sklar, *A Primer on Turbo Concepts*. IEEE Communication Magazine, pp. 94-101, December 1997.
- [Tuc00] M. Tuchler, R. Koetter, and A. Singer, *Iterative Correction of ISI via Equalization and Decoding using Priors*. in Proc. Intern. Symp. on Information Theory, Sorrento, Italy, p. 194, July 2000.
- [Tuc02] M. Tuchler, R. Koetter, and A. Singer, *Turbo Equalization: Principles and New Results*. IEEE Transactions on Communications, vol. 50, pp. 754-767, May 2002.
- [Tuc04] M. Tuchler, *Turbo Equalization*. Ph.D. Thesis, University of Munich, Germany. Esta tese foi cedida por Michael Tüchler apesar de não ter sido defendida ainda.
- [Yea02] 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*. IEEE Transactions on Wireless Communications, vol. 1, N<sup>o</sup>2, April 2002.