José da Silva Barros

Códigos Turbo Híbridos Multiníveis

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica.
Área de concentração: Telecomunicações.

Orientador: Renato Baldini Filho

Campinas, SP
2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

B278c	Barros, José da Silva Códigos turbo híbridos multiníveis / José da Silva Barros. --Campinas, SP: [s.n.], 2007. Orientador: Renato Baldini Filho. Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 1. Teoria da informação. 2. Codificação. 3. Códigos de controle de erros (Teoria da informação). I. Renato Baldini Filho. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Hybrid multilevel turbo codes.
Palavras-chave em Inglês: Information theory, Coding, Error control codes (Information theory).
Área de concentração: Engenharia de Computação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora: Dalton Soares Arantes, Geraldo Gil Ramundo Gomes, Hélio Pires de Almeida, Yuzo Ialo e Renato da Rocha Lopes
Data da defesa: 28/11/2007
Programa de Pós-Graduação: Engenharia Elétrica

José da Silva Barros

Códigos Turbo Híbridos Multiníveis

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica.
Área de concentração: Telecomunicações.

Banca Examinadora:
Prof. Dr. Renato Baldini Filho - FEEC/UNICAMP
Prof. Dr. Hélio Pires de Almeida - CCEN/UFPB
Prof. Dr. Geraldo Gil Ramundo Gomes - INATEL
Prof. Dr. Dalton Soares Arantes - FEEC/UNICAMP
Prof. Dr. Yuzo Iano - FEEC/UNICAMP
Prof. Dr. Renato da Rocha Lopes - FEEC/UNICAMP

Campinas, SP
2007

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: José da Silva Barros

Data da Defesa: 28 de novembro de 2007

Título da Tese: "Códigos Turbo Híbridos Mutiníveis"

Prof. Dr. Renato Baldini Filho (Presidente): ____________________

Prof. Dr. Hélio Pires de Almeida: ____________________

Prof. Dr. Geraldo Gil Ramundo Gomes: ____________________

Prof. Dr. Renato da Rocha Lopes: ____________________

Prof. Dr. Dalton Soares Arantes: ____________________

Prof. Dr. Yuzo Iano: ____________________

Resumo

Neste trabalho apresentamos classes de códigos turbo não binários definidos sobre os campos e anéis de inteiros. Os **códigos turbo multiníveis** convencionais consistem em dois códigos componentes *RSC M-ários*, concatenados via um entrelaçador aleatório de N símbolos e com símbolos codificados transmitidos através da modulação *M-PSK*. Os **códigos turbo híbridos multiníveis** consistem em dois códigos componentes *RSC*, não necessariamente definidos sobre o mesmo alfabeto. Os codificadores componentes são separados por um entrelaçador e os símbolos codificados transmitidos através de um esquema híbrido de modulação *PSK*. O algoritmo de decodificação iterativa de máximo a posteriori, usado para decodificar os códigos concatenados binários, pode ser estendido para a classe dos códigos turbo não binários. Os resultados das simulações mostram que os códigos turbo híbridos multiníveis apresentam melhor desempenho, error floor mais baixo e menor complexidade de codificação e decodificação que os códigos turbo *M-ários* convencionais. Já os códigos turbo multiníveis *M-ários* são mais eficientes que os códigos turbo binários padrão.

Abstract

This work presents classes of non-binary codes defined over rings and fields of integers. The conventional **multilevel turbo codes** consist of two *M-ary RSC* component codes concatenated via a random N-symbol interleaver and with encoded symbols are transmitted using a *M-PSK* modulation. The **hybrid multilevel turbo codes** consist of two *RSC* component codes, defined on different alphabets. The component encoder are separated by a interleaver and the encoder symbols are transmitted utilizing a hybrid *M-PSK* scheme. The iterative binary decoding algorithm is a maximum a posteriori scheme, which can be extended to the class of the non-binary turbo codes. The results of the simulations show that the hybrid multilevel turbo codes present better performance, lower error floor and lower encoding and decoding complexities than the M-ary conventional turbo codes. Moreover, the M-ary multilevel turbo codes are more efficient than the standard binary turbo codes.

A Deus, pela vida e saúde.

A meus pais, meus irmãos e minha esposa Genaldir, por tudo que temos vivido.

Agradecimentos

- A Deus, pelo êxito nessa caminhada.

- A meus pais e meus irmãos pela incomparável lição de vida, amor, trabalho e dedicação.

- A minha esposa Genaldir, pelo seu amor, carinho e compreensão nos momentos em que tive de me dedicar ao trabalho.

- A todos os meus amigos e parentes que, passando pela minha vida, contribuíram de alguma forma para a minha carreira.

- Um agradecimento especial ao meu orientador, Professor Renato Baldini, pela dedicação, paciência e boa vontade desprendidas ao meu trabalho de Doutorado.

- Aos Professores Dalton Soares, Geraldo Gil, Hélio Pires, Yuzo Iano e Renato Lopes, membros da Banca Examinadora.

- Aos amigos da pós graduação, especialmente André, Ariadne, Fábio, Gonzalo, Márzio, Rodrigo e Tarciana, que contribuíram muito para a realização deste trabalho.

- Aos professores e funcionários da FEEC, pelo apoio e dedicação.

- A Fundação de Amparo à Pesquisa do Estado de Alagoas - FAPEAL, pelo apoio financeiro.

Conteúdo

Resumo vii

Abstract vii

Agradecimentos x

Lista de Figuras xvi

Lista de Tabelas xx

1 Introdução 1

1.1 Objetivo 3

1.2 Organização da Tese 4

2 Códigos Turbo M-ários 7

2.1 Introdução 7

2.2 Fundamentos Matemáticos 8

2.3 Codificação Turbo *M-ária* 12

2.3.1 Codificador Convolucional Sistemático Recursivo 13

2.3.2 Entrelaçador 19

2.3.3 Puncionador 23

2.3.4 Modulador 24

2.4 Decodificação Turbo *M-ária* . . . 30
2.4.1 Algoritmo *MAP* . . . 30
2.4.2 Resumo do Algoritmo *MAP* . . . 41
2.5 Decodificação Iterativa Turbo M-ária . . . 42
2.5.1 Um Estágio da Decodificação Turbo . . . 43
2.5.2 Processo de Decodificação Iterativa . . . 47
2.6 Conclusão . . . 53

3 Resultados dos Códigos Turbo Multiníveis 55
3.1 Introdução . . . 55
3.2 Resultados . . . 55
3.2.1 Esquema Turbo Binário com Modulação *B-PSK* . . . 58
3.2.2 Esquema Turbo Ternário com Modulação *3-PSK* . . . 59
3.2.3 Esquema Turbo Quaternário com Modulação *4-PSK* . . . 62
3.2.4 Esquema Turbo 5-ário com Modulação *5-PSK* . . . 63
3.2.5 Discussão . . . 64
3.3 Conclusão . . . 68

4 Códigos Turbo *L-M-ário* com Modulação *Q-L-M-PSK* 71
4.1 Introdução . . . 71
4.2 Codificação Turbo Híbrida Multinível . . . 72
4.2.1 Codificador Convolucional Sistemático Recursivo . . . 73
4.3 Taxa de Codificação Turbo . . . 75
4.3.1 Taxa de Codificação de Códigos Multiníveis . . . 75
4.3.2 Codificador com Múltiplos Alfabetos de Saída . . . 76
4.3.3 Modulador *Q-L-M-PSK* . . . 83
4.4 Algoritmo de Decodificação . . . 85
4.4.1 Decodificação Iterativa Turbo *L-M-ária* . . . 87

4.5 Conclusão 90

5 Resultados dos Códigos Turbo Híbridos Multiníveis 93

5.1 Introdução 93

5.2 Resultados 93

5.2.1 Esquemas Binário, *2-3-ário*, Ternário e *3-5-ário* com Entrada Binária . 96

5.2.2 Esquemas Quaternário, *5-ário*, *5-7-ário* e *7-ário* com Entrada Quaternária 101

5.3 Conclusão 105

6 Conclusões 107

6.1 Resumo da tese 107

6.2 Contribuições da tese 109

6.3 Trabalhos Futuros 110

6.4 Publicações 110

Lista de Figuras

2.1 Sistema de comunicação turbo 7

2.2 Esquema de codificação turbo. 12

2.3 Diagrama do codificador multinível. 13

2.4 Diagrama de um codificador *RSC* quaternário. 18

2.5 Entrelaçador de bloco, $N = 1024$, usado por Berrou-Glavieux. 21

2.6 Entrelaçador s-aleatório de bloco ($N = 1024$). 23

2.7 Constelação *M-PSK* 25

2.8 Variação da constelação *M-PSK*. 27

2.9 Treliça do decodificador *MAP* para o codificador *RSC* ternário com matriz geradora $g(D) = \left[1 \ \frac{1+2D}{1+D}\right]$. 34

2.10 Treliça das expressões recursivas $\bar{\alpha}_k(0)$ e $\bar{\beta}_k(0)$. 36

2.11 Resumo das principais expressões usadas no algoritmo *MAP*. 42

2.12 Decodificação iterativa turbo. 43

2.13 Decodificação iterativa turbo. 48

2.14 Efeito da variação do número de iterações para $N = 100$ símbolos ou $N = 200$ *bits*, taxa de codificação turbo 1/2 e $g(D) = \left[1 \ \frac{2+D+2D^2}{1+D+3D^2}\right]$, sobre a curva da *taxa de erro de bit* $\times E_b/N_0$. 53

3.1 Esquema de codificação turbo. 56

3.2 Decodificação iterativa turbo. 57

3.3 Desempenho dos códigos turbo para taxa de codificação turbo igual a 1/2 e comprimento da seqüência de informação $N = 2000$ bits. 59
3.4 Desempenho dos códigos turbo binários e ternários para taxa de codificação turbo 1/2 e comprimento da seqüência de informação $N = 2000$ bits. 60
3.5 Desempenho dos códigos turbo binários e quaternários para taxa de codificação turbo 1/2 e comprimento da seqüência de informação $N = 2000$ bits. 62
3.6 Desempenho dos códigos turbo binários e 5-ários para taxa de codificação turbo 1/2 e comprimento da seqüência de informação $N = 2000$ bits. 64
3.7 Processo iterativo turbo ternário. 65
3.8 Variação do comprimento da seqüência de informação do sistema ternário. . . 66
3.9 Variação do número de estados. 67

4.1 Esquema de codificação turbo *L-M-ário* com símbolos de entrada do alfabeto *Q-ário*. 73
4.2 Diagrama de um codificador *RSC* definido sobre $\mathbb{Z}_5$ com entrada definida sobre $\mathbb{Z}_4$. 75
4.3 Diagrama do codificador multinível. 76
4.4 Diagrama do codificador multinível com múltiplos alfabetos de saída. 76
4.5 Diagrama do codificador turbo com codificadores componentes definidos sobre $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e símbolos de informação definidos sobre $\mathbb{Z}_4$. 77
4.6 Esquema de codificação turbo *M-ário* com símbolos de entrada do alfabeto *Q-ário*. 78
4.7 Diagrama dos codificadores *RSC* definidos sobre $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e dos respectivos codificadores turbo com entradas quaternárias. 79
4.8 Diagrama do sistema turbo com modulação *Q-L-M-PSK* 84
4.9 Constelações *Q-PSK*, *L-PSK* e *M-PSK* 84
4.10 Decodificação iterativa turbo. 88

5.1 Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas binário com taxas 1/2 e 1/3, *2-3-ário* com taxa 0.43, ternário com taxa 0.3868. 97

5.2 Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas *2-3-ário* com taxa 0.43 e o número de estados assumindo os valores 13, 17 e 25. 99

5.3 Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas binário com taxa 0.333 e *3-5-ário* com taxa 0.335 e 32 estados. 100

5.4 Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas quaternário, *5-ário*, *5-7-ário* e *7-ário* com taxas de codificação iguais a 0.5, 0.46, 0.43 e 0.41, respectivamente. 102

5.5 Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas binários, *3-5-ário* e *5-7-ário* com taxas iguais a 0.33 e 32 estados. 104

Lista de Tabelas

2.1 Adição e multiplicação para o anel de inteiros $\mathbb{Z}_4$. 10

2.2 Adição e multiplicação para o anel de inteiros $\mathbb{Z}_5$. 11

2.3 Escrita e leitura na matriz do entrelaçador ($\rightarrow$ *Escrita*, $\downarrow$ *Leitura*). 20

2.4 Codificadores binários. 29

2.5 Codificadores ternários. 29

2.6 Codificadores quaternários. 29

2.7 Codificadores 5-ários. 30

2.8 Seqüência de informação de comprimento $N = 100$ símbolos. 50

2.9 100 Símbolos decodificados - 23 erros ocorridos. 50

2.10 100 Símbolos decodificados - 16 erros ocorridos. 51

2.11 100 Símbolos decodificados - 15 erros ocorridos. 51

2.12 100 Símbolos decodificados - 11 erros ocorridos. 51

2.13 100 Símbolos decodificados - 1 erro ocorrido. 52

2.14 100 Símbolos decodificados - sem erro. 52

4.1 Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entrada binária e $2 = L \leq M \leq 7$. 81

4.2 Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária e ternária e $3 = L \leq M \leq 7$. 81

4.3 Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária e quaternária e $4 = L \leq M \leq 7$. 81

4.4 Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária, quaternária e *5-ária* e $5 = L \leq M \leq 7$. 82

4.5 Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária, quaternária, *5-ária* e *6-ária* e $6 = L \leq M \leq 7$. 82

4.6 Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária, quaternária, *5-ária*, *6-ária* e *7-ária* e $L = M = 7$. 83

5.1 Codificadores com entrada binária. 94

5.2 Codificadores com entrada quaternária. 95

Capítulo 1

Introdução

O início da teoria da informação e codificação se deu em 1948, com a publicação do artigo "A Mathematical Theory of Communication" pelo matemático e engenheiro Claude E. Shannon [1]. Neste artigo, Shannon estabeleceu os limitantes teóricos para uma comunicação confiável através de um canal ruidoso.

O teorema da codificação de canal garante ser possível realizar transmissões a probabilidades de erro tão baixa quanto se queira, desde que a taxa de transmissão seja menor que a capacidade do canal utilizado, e que o comprimento do código empregado seja suficientemente grande para um canal *AWGN* (*Additive White Gaussian Noise*). O teorema afirma também que existem códigos que permitem a transmissão confiável, mas não mostra como encontrá-los.

A busca por estes códigos motivou vários pesquisadores da área e impulsionou a criação de diversas classes de códigos com aplicação prática mas que, no entanto, não eram capazes de operar próximos ao limitante de Shannon.

Com a introdução do conceito de modulação em treliça por Ungerboeck, em 1982 [9], a distância entre o desempenho do sistema codificado e o limitante de Shannon se estreitou significativamente. Os códigos turbo binários apresentados pela primeira vez por C. Berrou, A. Glavieux e P. Thitimajashima em 1993 [2], representaram o passo final em direção a este limitante. Estes códigos turbo binários apresentam desempenho muito próximo ao limitante de Shannon para o canal com ruído aditivo gaussiano branco (*AWGN*) e sua estrutura com códigos entrelaçados permite um processo de decodificação iterativa por máximo *a posteriori* (*MAP*).

A maioria dos códigos turbo pesquisados utilizam códigos componentes binários concatenados via um entrelaçador aleatório de N bits [2], [3], [7], [10] e [19]. Neste trabalho de tese, apresentamos códigos turbo construídos a partir de códigos componentes não binários, concatenados via um entrelaçador aleatório de N símbolos e cujos símbolos codificados são mapeados em uma constelação apropriada *PSK* (*phase-shift-keying*), para transmissão através de um canal *AWGN*. O processo de decodificação usado para avaliar o desempenho dos códigos turbo apresentados neste trabalho é a decodificação iterativa de máximo *a posteriori* [2], [6], [17] e [27].

Estes códigos não binários foram divididos em duas classes:

- A classe dos **códigos turbo multiníveis convencionais**, cujo esquema de codificação é formado por dois códigos componentes *RSC* (Recursive Systematic Convolutional) *M-ários*, concatenados via um entrelaçador aleatório de N símbolos e com os símbolos codificados transmitidos através da modulação *M-PSK*. O processo de decodificação iterativa utiliza o algoritmo de decodificação *MAP* em cada decodificador componente;

- A classe dos **códigos turbo híbridos multiníveis *L-M-ários*** com símbolos de informação definidos sobre o anel de inteiros $\mathbb{Z}_Q$, onde $\mathbb{Z}_Q \in \{\mathbb{Z}_4, \mathbb{Z}_8, \mathbb{Z}_{16}, \cdots\}$ e $Q \leq L \leq M$. O esquema de codificação *L-M-ário* é formado por um codificador *RSC-1 L-ário*, concatenado via um entrelaçador aleatório de N símbolos com um codificador *RSC-2 M-ário*. O alfabeto de entrada deste codificador é *Q-ário* e a decodificação de máximo *a posteriori* é adaptada para processar os símbolos de informação pertencentes ao anel $\mathbb{Z}_Q$ e os símbolos de paridade pertencentes aos campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$.

1.1 Objetivo

Os códigos turbo binários, com concatenação paralela [2], [3], [7], [10], [11] e [19], fornecem desempenho próximo à capacidade de canal com baixa relação sinal-ruído (*signal-to-noise-ratio-SNR*). Estes códigos apresentam uma complexidade de decodificação significativa devido ao processo iterativo, ao número de estados na treliça do codificador componente e ao comprimento da seqüência de informação.

Nesta tese, apresentamos classes de códigos turbo multiníveis que podem apresentar algumas vantagens sobre os códigos turbo binários padrão, tais como, melhor desempenho associado a menor complexidade de decodificação do sistema turbo.

Na primeira parte desta tese, apresentamos classes de códigos turbo *M-ários* definidos sobre os campos e anéis de números inteiros $\mathbb{Z}_M$. Este esquema proposto permite maior flexibilidade na combinação: complexidade de decodificação, capacidade de correção de erro, cardinalidade do alfabeto e largura de banda de freqüência ocupada. Devido ao casamento natural do tamanho da constelação com o tamanho do alfabeto, é possível encontrar esquemas turbo *M-ários* mais eficientes, com relação aos parâmetros mencionados acima, que os esquemas turbo binários.

Na segunda parte, propomos classes de códigos turbo *L-M-ários* definidos sobre os campos de números inteiros $\mathbb{Z}_L$ e $\mathbb{Z}_M$, que processam símbolos de informação definidos sobre os anéis de inteiros $\mathbb{Z}_Q$. Este esquema produz maior flexibilidade na combinação: complexidade de decodificação, capacidade de correção de erro, taxa de codificação, cardinalidade do alfabeto e largura de banda. Como a fonte de informação é *Q-ária* e o codificador é *L-M-ário* , onde Q é uma potência positiva de 2 e $2 \leq Q \leq L \leq M$, é possível encontrar um esquema *L-M-ário* que proporcione perfeito mapeamento de bits para símbolos e apresente melhor desempenho que o esquema turbo *M-ário* convencionais.

1.2 Organização da Tese

A tese está organizada em 6 capítulos, cujos conteúdos são descritos como a seguir.

No Capítulo 1, apresentamos uma introdução, os objetivos e a organização da tese. No Capítulo 2, descrevemos o esquema de codificação turbo *M-ário*. Este capítulo inicia-se com as definições matemáticas que garantem a existência (construção) dos códigos convolucionais sistemáticos recursivos *M-ário*, e segue descrevendo os componentes do codificador turbo, tais como, o codificador, o entrelaçador e o puncionador. Descrevemos também, de forma breve, o modulador e o canal de comunicação utilizado. Em seguida, apresentamos o algoritmo de decodificação de máximo *a posteriori* (MAP), que fornece a informação *a posteriori* a partir das propriedades estatísticas e os conceitos de código de treliça [2] e [8]. Finalmente, o funcionamento do esquema de decodificação iterativa turbo (a cada iteração) é descrito fazendo uso dos conceitos de informação intrínseca, extrínseca e *a posteriori*.

No Capítulo 3, apresentamos os resultados da simulação para o desempenho dos esquemas turbo *M-ários*, definidos sobre os campos e anéis $\mathbb{Z}_M$, para $M \leq 5$. Começamos com o esquema turbo binário tradicional e seguimos com os esquema ternário, quaternário e *5-ário*, avaliando o desempenho, a complexidade, o mapeamento entre símbolos e bits, a largura de banda, o tamanho do alfabeto e o tempo de decodificação. Em seguida, analisamos o desempenho dos códigos definidos sobre os campos e anéis.

No Capítulo 4, apresentamos o esquema de codificação turbo *L-M-ário* com símbolos *Q-ários* como entrada, e símbolos *Q-ários*, *L-ários* e *M-ários* como saída, onde Q, L e M são números inteiros positivos e $2 \leq Q \leq L \leq M$. Em seguida, construímos um esquema híbrido de modulação *Q-L-M-PSK* e adaptamos o esquema de decodificação iterativa para este esquema híbrido de codificação turbo.

No Capítulo 5, apresentamos as curvas da probabilidade de erro de bits dos esquemas turbo *L-M-ários* definidos sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$. Neste capítulo, avalia-se os aspectos relacionados às simulações, analisando os resultados de forma comparativa com relação ao desempenho, a complexidade, ao mapeamento entre bits e símbolos, a largura de banda, a taxa de codificação, o comprimento do alfabeto e o tempo de decodificação destes esquemas.

No Capítulo 6, apresentamos as conclusões, as contribuições da tese e os trabalhos futuros.

Capítulo 2

Códigos Turbo M-ários

2.1 Introdução

No sistema de comunicação digital mostrado na Figura 2.1, as funções de codificação e modulação são consideradas como operações separadas, enquanto as funções de demodulação e decodificação são realizadas de forma conjuntas. O modulador mapeia os símbolos do alfabeto *M-ário* em elementos do alfabeto da modulação e o demodulador permite a operação reversa. A função do conjunto formado pelo codificador e pelo decodificador é tentar corrigir o máximo dos possíveis padrões de erro que podem ocorrer no canal de comunicação.

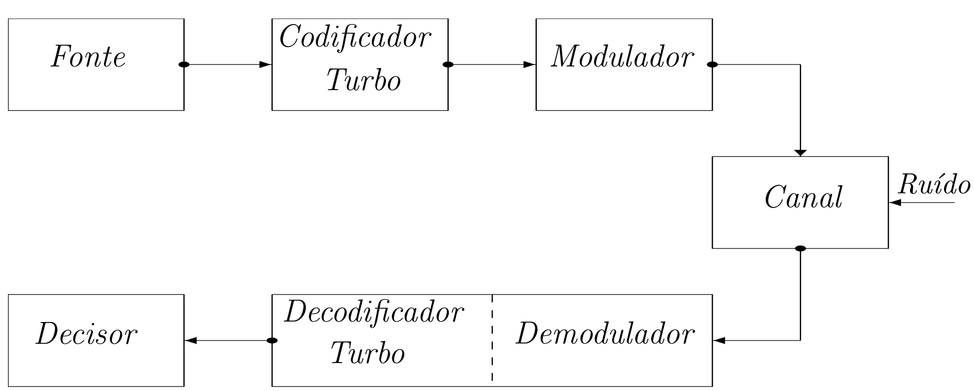


Figura 2.1: Sistema de comunicação turbo

A necessidade de tornar o sistema de comunicação mais confiável, em termos da probabilidade de erro de símbolo (ou bit), implica na introdução de mais redundância, o que tem como conseqüência a diminuição da taxa de informação.

Como o canal de comunicação utilizado neste trabalho possui largura de banda e potência limitadas, precisamos ampliar o conjunto de sinais do sistema de modulação para compensar a perda na taxa de informação produzida no processo de codificação.

As técnicas de codificação apresentadas por Ungerboeck em 1982 permitem encontrar ganhos significativos de codificação convolucional multinível sobre a modulação não codificada, sem sacrificar a taxa de informação, a potência do sinal e a largura de banda.

Neste capítulo, apresentamos os códigos turbo *M-ários*, definidos sobre os campos ou anéis de inteiros módulo-M, $\mathbb{Z}_M$. A estrutura de codificação e decodificação destes códigos proporciona desempenho próximo à capacidade de canal com baixa relação sinal ruído. Na primeira seção apresentamos o esquema de codificação e nas duas seções seguintes descrevemos o algoritmo de decodificação e o processo de decodificação iterativa *M-ária*.

Note que o tratamento matemático dado neste capítulo, para os códigos turbo *M-ários* definidos sobre os anéis de inteiros módulo-M, $\mathbb{Z}_M$, se estende para os códigos turbo *M-ários* definidos sobre os campos de inteiros módulo-M, $\mathbb{Z}_M$.

2.2 Fundamentos Matemáticos

Conjuntos Numéricos

- Naturais: $\mathbb{N} = \{1, 2, 3, 4, 5, \cdots\}$;
- Inteiros: $\mathbb{Z} = \{0, \pm1, \pm2, \pm3, \pm4, \pm5, \cdots\}$, $\mathbb{Z}^* = \{\pm1, \pm2, \pm3, \pm4, \pm5, \cdots\}$;
- Inteiros positivos $\mathbb{Z}_+^* = \{1, 2, 3, 4, 5, \cdots\}$;
- Inteiros não negativos $\mathbb{Z}_+ = \{0, 1, 2, 3, 4, 5, \cdots\} \supseteq \mathbb{N}$;

- Racionais: $\mathbb{Q} = \left\{ \frac{m}{n}, m \in \mathbb{Z}, n \in \mathbb{Z}^* \right\}$;
- Reais: $\mathbb{R} = \mathbb{Q} \cup \mathbb{I}$, onde $\mathbb{I}$ é o conjunto dos números irracionais (decimais não-exatos e não-periódicos).

Seja m um número inteiro positivo. Sobre $\mathbb{Z}$, definimos a relação de congruência, $\underset{m}{\equiv}$, da seguinte maneira: para $a, b \in \mathbb{Z}$, $a \underset{m}{\equiv} b \Longleftrightarrow b - a$ é um múltiplo de m. Em vez de escrever $a \underset{m}{\equiv} b$, escreve-se $a \equiv b \mod (m)$ e diz-se que a é côngruo a b módulo m. Em outras palavras, dois números inteiros a e b serão ditos congruentes módulo m se os restos de a e b por m forem iguais.

Exemplo 2.2.1 $12 \equiv 17 \mod (5)$, *pois* $17 = 3 \cdot 5 + 2$ *e* $12 = 2 \cdot 5 + 2$, *ou ainda* $17 - 12$ *é um múltiplo de 5.*

Definição 2.2.2 *Um grupo A é um conjunto de elementos com uma operação binária chamada adição(+) que satisfaz as seguintes propriedades:*

1. *Fechamento:* $\forall x_1, x_2 \in A \Rightarrow x_1 + x_2 \in A$
2. *Associativa:* $\forall x_1, x_2, x_3 \in A \Rightarrow x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$
3. *Elemento neutro:* $\exists\ 0 \in A$ *tal que* $\forall x \in A,\ x + 0 = 0 + x = x$
4. *Elemento inverso:* $\forall x \in A,\ \exists - x \in A$ *tal que* $x + (-x) = (-x) + x = 0$.

Se um grupo A satisfaz a propriedade comutativa: $(\forall x_1, x_2 \in A \Rightarrow x_1 + x_2 = x_2 + x_1)$ tal grupo é chamado de *grupo abeliano*.

Definição 2.2.3 *Um anel R é um conjunto de elementos com duas operações chamadas de adição(+) e de multiplicação $(\cdot)$ que satisfaz os seguintes axiomas:*

1. *O conjunto R é um grupo abeliano sobre a adição.*
2. *Fechamento:* $\forall x_1, x_2 \in R \Rightarrow x_1 \cdot x_2 \in R$.

3. *Associativa:* $\forall x_1, x_2, x_3 \in R \Rightarrow x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3$.

4. *Distributiva:*

$$\forall x_1, x_2, x_3 \in R \Rightarrow \begin{cases} x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3 \\ (x_2 + x_3) \cdot x_1 = x_2 \cdot x_1 + x_3 \cdot x_1. \end{cases}$$

Um anel R é chamado comutativo com unidade se sua multiplicação é comutativa e existe $1 \in R$, isto é, $\forall x_1, x_2 \in R \Rightarrow x_1 \cdot x_2 = x_2 \cdot x_1$ e $1 \cdot x = x, \forall x \in R$.

Exemplo 2.2.4 *O anel de inteiros módulo* 4 *é definido pelo conjunto* $\{0, 1, 2, 3\}$*, onde as operações adição e multiplicação entre dois elementos deste conjunto são dadas na tabela 2.1.*

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Tabela 2.1: Adição e multiplicação para o anel de inteiros $\mathbb{Z}_4$.

O anel de inteiros módulo 4 é um anel comutativo com o elemento unidade. Anéis que não têm o elemento unidade não serão considerados neste trabalho.

Um campo é um anel comutativo com unidade que obedece a propriedade de existência de inverso, ou seja, dado $x \in R$, $x \neq 0$, existe $x^{-1} \in R$ tal que $x \cdot x^{-1} = 1$. Note que para todo elemento não nulo de um campo existe um único inverso multiplicativo.

Exemplo 2.2.5 *O campo de inteiros módulo* 5 *é definido pelo conjunto* $\{0, 1, 2, 3, 4\}$*, onde as operações adição e multiplicação entre dois elementos deste conjunto são dadas na tabela 2.2.*

Mais geralmente, dado $M \in \mathbb{N}$, o anel dos inteiros módulo M, $\mathbb{Z}_M$, é definido pelo conjunto $\{0, 1, 2, 3, \ldots, M-1\}$, com as seguintes operações de adição e de multiplicação módulo M.

$$\begin{cases} x + y = \text{resto da divisão da soma usual de } x \text{ com } y \text{ por } M \\ x \cdot y = \text{resto da divisão do produto usual de } x \text{ e } y \text{ por } M. \end{cases}$$

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

·	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Tabela 2.2: Adição e multiplicação para o anel de inteiros $\mathbb{Z}_5$.

Definição 2.2.6 *Diz-se que N é um subanel de um anel R se $N \subseteq R$ e N também forma um anel com as operações $+$ e $\cdot$ definidas para R.*

Definição 2.2.7 *Se X e Y pertencem ao conjunto dos números reais $\mathbb{R}$, a distância euclidiana entre X e Y é dada pela seguinte expressão: $D_E(X,\ Y) = |Y - X|$.*

Definição 2.2.8 *Se X e Y são dois pontos do plano bidimensional $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$, a distância euclidiana ao quadrado entre $X = (x_1,\ x_2)$ e $Y = (y_1,\ y_2)$ é dada pela seguinte expressão:*

$$\begin{aligned} D_E^2(X,\ Y) &= \parallel Y - X \parallel^2 \\ &= (y_1 - x_1)^2 + (y_2 - x_2)^2. \end{aligned} \tag{2.1}$$

Definição 2.2.9 *Um polinômio numa variável x sobre um anel R é uma seqüência $(a_0, a_1, \cdots, a_n)$ que pode ser representada pela soma $a_0 + a_1x + \cdots + a_nx^n$ ou $f(x) \doteq a_0 + a_1x + \cdots + a_nx^n$, onde n é o grau do polinômio, $a_i \in R$ para todo índice e $a_n \neq 0$.*

Exemplo 2.2.10 *A seqüência $(2, 1, 3, 5, 0, 4)$ representada por $P(x) = 2 + x + 3x^2 + 5x^3 + 4x^5$ é um polinômio de grau 5 definido sobre o anel $\mathbb{Z}_6$.*

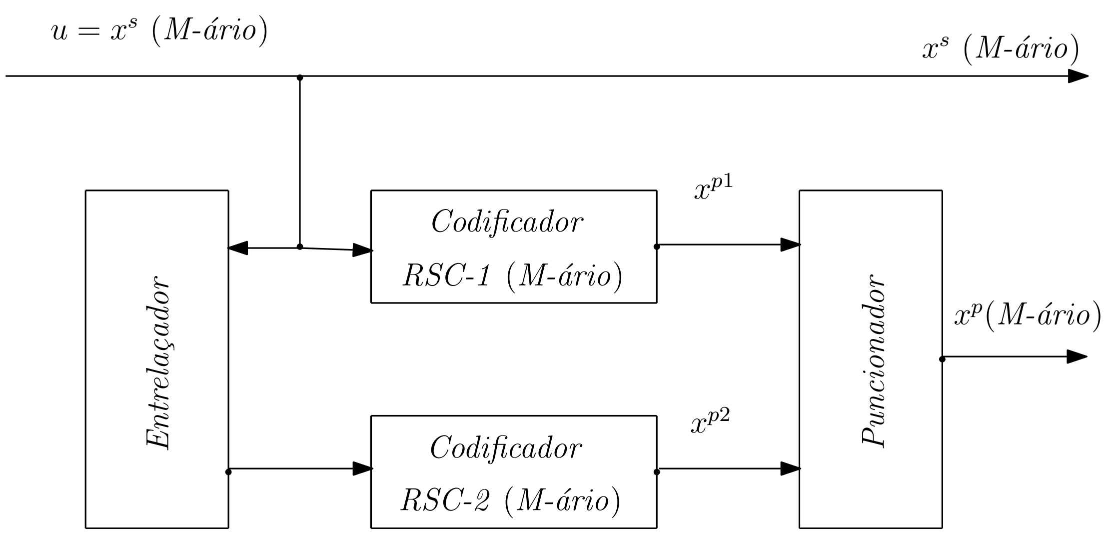
Definição 2.2.11 *O conjunto*

$$R[x] = \left\{ \sum_{i=0}^{n} a_i x^i ; n \in \mathbb{N} \text{ e } a_i \in R \right\}$$

é um anel de polinômios numa variável x sobre o anel R.

2.3 Codificação Turbo *M-ária*

Nesta seção, apresentamos as propriedades matemáticas que garantem a existência e construção dos códigos convolucionais multiníveis. Em seguida, os códigos convolucionais sistemáticos recursivos do codificador turbo *M-ário* da Figura 2.2 são apresentados. Além disso, destacamos os codificadores e entrelaçadores que proporcionam melhor desempenho e menor complexidade na decodificação. Um puncionador pode ser utilizado para aumentar a taxa do código. A modulação *M-PSK* foi escolhida por ter um casamento natural com os símbolos *M-ários* da codificação. O canal de comunicação é o *AWGN*.



$u = x^s$	Seqüência de símbolos de informação ou de símbolos sistemáticos
x^{p1}	Seqüência de símbolos de paridade proveniente do codificador $RSC-1$
x^{p2}	Seqüência de símbolos de paridade proveniente do codificador $RSC-2$
x^p	Seqüência de símbolos de paridade proveniente do puncionador

Figura 2.2: Esquema de codificação turbo.

A descrição do esquema de codificação turbo *M-ário* definido sobre os anéis $\mathbb{Z}_M$ é equivalente à descrição do esquema binário definido sobre o campo $\mathbb{Z}_2$, [2]. Desse modo, o entrelaçador e o puncionador são considerados equivalentes aos do caso binário.

O codificador turbo *M-ário* proposto é constituído por dois codificadores convolucionais idênticos, de taxa 1/2, separados por um entrelaçador de N símbolos, juntamente com um mecanismo opcional de puncionamento. Esses dois codificadores constituintes são codificadores convolucionais sistemáticos recursivos (*RSC-Recursive Systematic Convolutional*), concatenados em paralelo, conforme a Figura 2.2.

2.3.1 Codificador Convolucional Sistemático Recursivo

Nesta subseção apresentamos os fundamentos matemáticos para a construção dos códigos convolucionais sistemáticos e, a partir destes, construímos os códigos convolucionais sistemáticos recursivos.

A Figura 2.3 mostra o diagrama do codificador convolucional multinível que possui k entradas e n saída com símbolos do alfabeto *M-ário*. A taxa de codificação é dada pela expressão:

$$r = \frac{\log M^k}{\log M^n} = \frac{k \log M}{n \log M} = \frac{k}{n}. \tag{2.2}$$

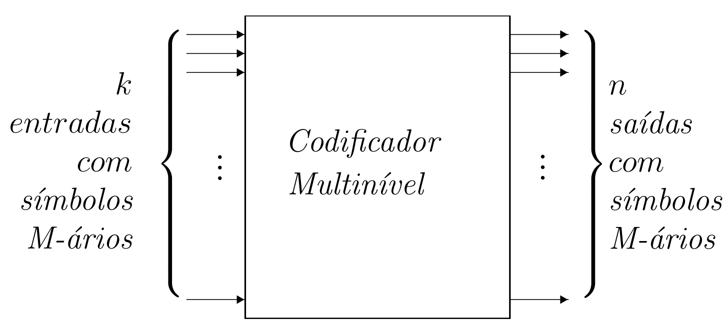


Figura 2.3: Diagrama do codificador multinível.

Seja $\mathbb{Z}_M$ um anel de inteiros comutativo com identidade multiplicativa e seja $L(D)$ o anel dos polinômios com coeficientes em $\mathbb{Z}_M$. Os polinômios de $L(D)$ são expressos pela série de

Laurent [25]

$$f(D) = \sum_{i=0}^{n} f_i D^i = f_0 + f_1 D + f_2 D^2 + \cdots + f_n D^n,$$

onde n é o grau de $f(D)$, D representa uma unidade de atraso e $f_i \in \mathbb{Z}_M$. Se $f(D) \neq 0$ e $f_0 \neq 0$, então f_0 é chamado de coeficiente mínimo de $f(D)$.

Desse modo, chamamos de codificador convolucional com taxa de codificação k/n sobre o anel de funções racionais $L(D)$, o mapeamento linear

$$\begin{array}{ccc} L(D)^k & \longrightarrow & L(D)^n \\ u(D) & \longmapsto & v(D), \end{array}$$

que pode ser expresso como

$$v(D) = u(D)G(D),$$

onde $G(D)$ é uma matriz $k \times n$ (chamada matriz função de transferência ou matriz geradora), com elementos em $L(D)$, cujas linhas são linearmente independentes sobre $L(D)$. E, ainda, $u(D)$ é o polinômio correspondente à seqüência de informação, $v(D)$ é o polinômio correspondente à seqüência código. Então, o conjunto

$$C = \{u(D)G(D) \mid u(D) \in L(D)^k\},$$

é um código convolucional com taxa de codificação k/n sobre $\mathbb{Z}_M$.

Exemplo 2.3.1 *Dada a seqüência de informação em* $\mathbb{Z}_8$

$$u(D) = 2 + 4D + 3D^2 + 7D^3 + D^4,$$

e a matriz geradora, 1×2, do código convolucional com taxa de codificação $1/2$

$$G(D) = \begin{bmatrix} 7 + 3D & 2 + D \end{bmatrix},$$

a palavra código sobre o anel de inteiros $\mathbb{Z}_8$*, é dada pela expressão:*

$$\begin{aligned} v(D) &= u(D)G(D) \\ &= \begin{bmatrix} 6+2D+D^2+2D^3+4D^4+3D^5 & 4+2D+2D^2+D^3+D^4+D^5 \end{bmatrix}. \end{aligned}$$

Agora, seja $L_r(D)$ um subanel de $L(D)$ (chamado anel das funções realizáveis sobre $\mathbb{Z}_M$), consistindo dos elementos que contem uma representação na forma de razão: $\dfrac{q(D)}{p(D)}$, onde $q(D)$ e $p(D) \in L(D)$ e o coeficiente mínimo de $p(D)$ é inversível em $\mathbb{Z}_M$.

Uma matriz geradora convolucional $G(D)$, $k \times n$, é dita sistemática se ela possuir uma submatriz $k \times k$ que é uma matriz identidade.

Teorema 2.3.2 *Um código convolucional* C *sobre o anel* $\mathbb{Z}_M$ *é sistemático se, e somente se, ele possui uma matriz geradora,* $G(D)$*,* $k \times n$*, que possui uma submatriz* $k \times k$ *cujo determinante é igual a unidade em* $L_r(D)$ *- anel das funções realizáveis sobre* $\mathbb{Z}_M$*.*

Prova.

($\Rightarrow$) Se $G(D)$ é sistemática, então $G(D) = \begin{bmatrix} I_k & \bar{G}(D) \end{bmatrix}$. Portanto, o determinante de I_k é igual a unidade em $\mathbb{Z}_M$ (e em $L_r(D)$).

($\Leftarrow$) Se $G(D)$ possui uma submatriz $k \times k$, $A(D)$, cujo determinante é igual à unidade em $L_r(D)$, então, sem perda de generalidade, seja $G(D) = \begin{bmatrix} A(D) & B(D) \end{bmatrix}$. Como $A(D)$ possui determinante unitário em $L_r(D)$, então $A(D)$ possui uma inversa $A^{-1}(D)$ em $L_r(D)$ e

$$G_{sist}(D) = A^{-1}(D)G(D) = \begin{bmatrix} I_k & \bar{B}(D) \end{bmatrix}$$

é uma matriz geradora equivalente para o código C. ■

Um codificador convolucional sistemático recursivo-*RSC (Recursive Systematic Convolutional)*, com taxa de codificação k/n sobre o anel de funções de realizáveis $L_r(D)$, é um mapeamento linear

$$\begin{aligned} L_r(D)^k &\longrightarrow L_r(D)^n \\ u(D) &\longmapsto v(D), \end{aligned}$$

que pode ser expresso como

$$v(D) = u(D)G_r(D),$$

onde $G_r(D)$ é uma matriz $k \times n$ da forma

$$G_r(D) = \begin{bmatrix} I_k & X(D) \end{bmatrix}.$$

Observe que I_k é uma matriz identidade $k \times k$, $X(D)$ é uma matriz $k \times$*(n-k)* com elementos cuja representação é $\dfrac{q(D)}{p(D)}$ em $L_r(D)$ e o coeficiente mínimo de $p(D)$ é inversível em $\mathbb{Z}_M$.

Exemplo 2.3.3 *Transformação da matriz geradora de um codificador convolucional não sistemático numa matriz geradora equivalente de um codificador convolucional sistemático recursivo, definido sobre o anel de inteiros $\mathbb{Z}_6$.*

Considere o codificador convolucional não sistemático sobre $\mathbb{Z}_6$ representado pela matriz geradora

$$G(D) = \begin{bmatrix} 1 & 5+4D & 1 & 3 \\ 0 & 1+3D & 2D & 2+4D \end{bmatrix}.$$

Podemos transformar este codificador, não sistemático, em um codificador RSC equivalente, como se segue. Seja

$$P(D) = \begin{bmatrix} 1 & 5+4D \\ 0 & 1+3D \end{bmatrix},$$

a submatriz k×k (2×2), formada pelas duas primeiras colunas da matriz $G(D)$ e seja

$$P^{-1}(D) = \begin{bmatrix} 1 & \dfrac{1+2D}{1+3D} \\ 0 & \dfrac{1}{1+3D} \end{bmatrix},$$

a inversa à direita da submatriz $P(D)$. Multiplicando $P^{-1}(D)$ por $G(D)$, obtemos a matriz sistemática recursiva

$$G_r(D) = P^{-1}(D)G(D) = \begin{bmatrix} 1 & 0 & \dfrac{1+5D+4D^2}{1+3D} & \dfrac{5+5D+2D^2}{1+3D} \\ 0 & 1 & \dfrac{2D}{1+3D} & \dfrac{2+4D}{1+3D} \end{bmatrix}.$$

Definição 2.3.4 *O conjunto*

$$C = \{u(D)g(D) \mid u(D) \in L_r(D)^k\},$$

é um código RSC com taxa de codificação k/n sobre $\mathbb{Z}_M$, onde $g(D) = G_r(D)$ é a matriz geradora de um codificador RSC com elementos em $L_r(D)$.

Note que temos interesse em um esquema de codificação e decodificação turbo *M-ário* que forneça o melhor desempenho possível, [2], [3], [4], [7], [10], [11] [18], [19] e [22]. Por outro lado, codificadores que proporcionam melhor desempenho possuem matrizes geradoras $\left(g(D) = \begin{bmatrix} 1 & \dfrac{q(D)}{p(D)} \end{bmatrix}\right)$ obedecendo as seguintes restrições:

1. O polinômio $p(D)$ de $\dfrac{q(D)}{p(D)}$ é irredutível em $L(D)$,
2. **O coeficiente mínimo do polinômio** $p(D)$ **é inversível em** $\mathbb{Z}_M$, se $g(D)$ for definida sobre um anel Notheriano [20], [21] e [25].

Note que quando os códigos turbo *M-ários* estão definidos sobre um campo de números inteiros $\mathbb{Z}_M$, a restrição 2 não é necessária, uma vez que todo elemento não nulo do campo $\mathbb{Z}_M$

possui inverso multiplicativo. Por outro lado, quando os códigos turbo *M-ários* estão definidos sobre um anel $\mathbb{Z}_M$, esta restrição 2 é necessária, pois nem todo elemento do anel $\mathbb{Z}_M$ possui inverso multiplicativo.

Exemplo 2.3.5 *A Figura 2.4 mostra o diagrama de um codificador RSC que tem matriz geradora em* $\mathbb{Z}_4$

$$g(D) = \left[\begin{array}{cc} 1 & \dfrac{2 + D + 2D^2}{1 + D + 3D^2} \end{array} \right]. \tag{2.3}$$

Note que D representa uma unidade de atraso, o codificador possui $4^2 = 16$ estados e os coeficientes dos polinômios do numerador e denominador de (2.3) são representados por multiplicadores na alimentação direta e na realimentação, respectivamente.

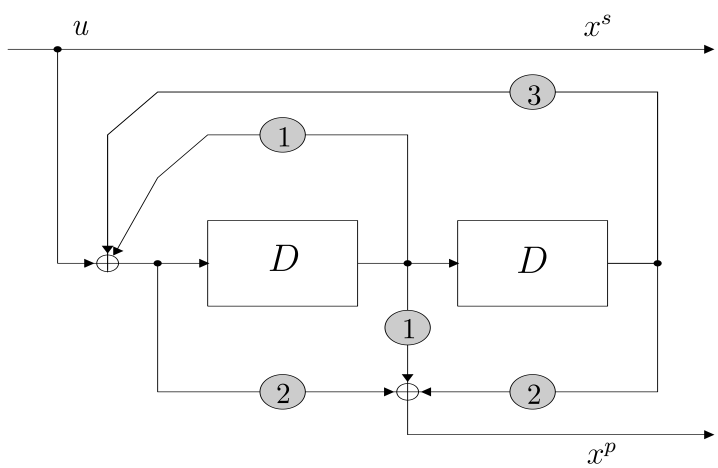


Figura 2.4: Diagrama de um codificador *RSC* quaternário.

A seqüência de informação

$$u = x^s = \quad 3\ 2\ 1\ 0\ 1\ 2\ 3\ 3\ 1\ 0\ 2\ 0\ 0\ 1\ 1\ 0\ 2\ 2\ 3\ 2\ 1\ 0\ 1\ 2\ 1$$

aplicada ao codificador RSC da Figura 2.4 produz a seqüência de saída (seqüência código)

$$x = \left| \begin{array}{l} x^s \quad 3\ 2\ 1\ 0\ 1\ 2\ 3\ 3\ 1\ 0\ 2\ 0\ 0\ 1\ 1\ 0\ 2\ 2\ 3\ 2\ 1\ 0\ 1\ 2\ 1 \\ x^p \quad 2\ 1\ 3\ 0\ 3\ 0\ 3\ 2\ 1\ 1\ 2\ 0\ 3\ 1\ 2\ 2\ 0\ 1\ 2\ 2\ 2\ 1\ 3\ 3\ 2. \end{array} \right.$$

A seqüência código é composta pela seqüência de informação (símbolos sistemáticos) - na 1ª linha, e pela seqüência de paridade - na 2ª linha. Os dois últimos símbolos sistemáticos são os símbolos que fazem este codificador *RSC*, com esta seqüência de informação, voltar ao estado inicial.

2.3.2 Entrelaçador

Os entrelaçadores têm por função espalhar erros que ocorrem em surto (*burst*) geralmente causados pelo ruído impulsivo e pelo desvanecimento (*fading*) seletivo. Entrelaçar os erros introduzidos pelo canal torna-os descorrelacionados, contribuindo assim para que os decodificadores sejam mais eficientes.

Assim, a operação realizada pelo entrelaçador pode ser representada por:

$$\begin{aligned} \pi \; : \; \mathbb{Z} & \longrightarrow \mathbb{Z} \\ i & \longmapsto \pi(i) \quad \forall\, i \in \mathbb{Z}, \end{aligned}$$

ou seja, uma permutação nos inteiros $\mathbb{Z}$. O entrelaçador é portanto uma função inversível, que recebe em sua entrada N símbolos de um dado alfabeto, e produz em sua saída os mesmos N símbolos em uma diferente ordem no tempo. Portanto, sua função no esquema de codificação turbo, Figura 2.2, é tomar cada bloco de N símbolos de informação que chega em sua entrada e rearranjá-lo em uma forma pseudo-aleatória para a codificação pelo codificador *RSC-2*.

Exemplo 2.3.6 *A Tabela 2.3 mostra um entrelaçador de 25 símbolos utilizando uma matriz* P*,* 5×5*. O entrelaçador matricial ordena seus elementos como linhas da matriz* P *e os lê como colunas, ou seja, transforma a seqüência de informação:*

$$x^s = u = \; 3 \;\; 2 \;\; 1 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 3 \;\; 1 \;\; 0 \;\; 2 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 2 \;\; 2 \;\; 3 \;\; 2 \;\; 1 \;\; 0 \;\; 1 \;\; 2 \;\; 1$$

na seqüência de informação entrelaçada:

$$u' = \; 3 \;\; 2 \;\; 2 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 0 \;\; 2 \;\; 0 \;\; 1 \;\; 3 \;\; 0 \;\; 2 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 3 \;\; 2 \;\; 1 \;\; 0 \;\; 1 \;\; 2 \;\; 1$$

que, por sua vez, inserida no codificador RSC-2 da Figura 2.2 (com matriz geradora descrita na expressão (2.3)), produz a seqüência código:

$$x = \left| \begin{array}{ccccccccccccccccccccccccc} x^s & 3 & 2 & 2 & 0 & 1 & 2 & 3 & 0 & 2 & 0 & 1 & 3 & 0 & 2 & 1 & 0 & 1 & 1 & 3 & 2 & 1 & 0 & 1 & 2 & 1 \\ x^p & 2 & 1 & 3 & 0 & 3 & 0 & 3 & 2 & 1 & 1 & 2 & 0 & 3 & 1 & 2 & 2 & 0 & 1 & 2 & 2 & 2 & 1 & 3 & 3 & 2. \end{array} \right.$$

$P =$

	↓	↓	↓	↓	↓
→	1	2	3	4	5
→	6	7	8	9	10
→	11	12	13	14	15
→	16	17	18	19	20
→	21	22	23	24	25

Tabela 2.3: Escrita e leitura na matriz do entrelaçador (→ *Escrita*, ↓ *Leitura*).

A Figura 2.5 mostra o entrelaçador usado por C. Berrou, A. Glavieux e P. Thitimajashima em seu esquema turbo binário [2], [22], cuja representação matemática é a seguinte:

Tome $K = 2^k,\ M = 2^m,\ k, m \in \mathbb{Z}_+^*$ [1] e defina oito números primos, por exemplo:

$$p(1) = 17, \quad p(2) = 37, \quad p(3) = 19, \quad p(4) = 29, \quad p(5) = 41, \quad p(6) = 23, \quad p(7) = 13, \quad p(8) = 7.$$

Então para cada número inteiro i, tal que $0 \leq i \leq K \cdot M$, o entrelaçador é gerado pela

[1] $\mathbb{Z}_+^*$ é o conjunto dos números inteiros positivos

permutação

$$\pi(i) = (c(i) + M) \cdot r(i),$$

onde

$$\begin{aligned} r(i) &= p(l+1) \cdot (c_0 + 1) - 1 \mod (K); \quad c(i) = (M/2 + 1) \cdot (r_0 + c_0) \mod (M); \\ r_0 &= i \mod (M); \quad c_0 = (i - r_0)/M \quad \text{e} \quad l = (r_0 + c_0) \mod (8). \end{aligned}$$

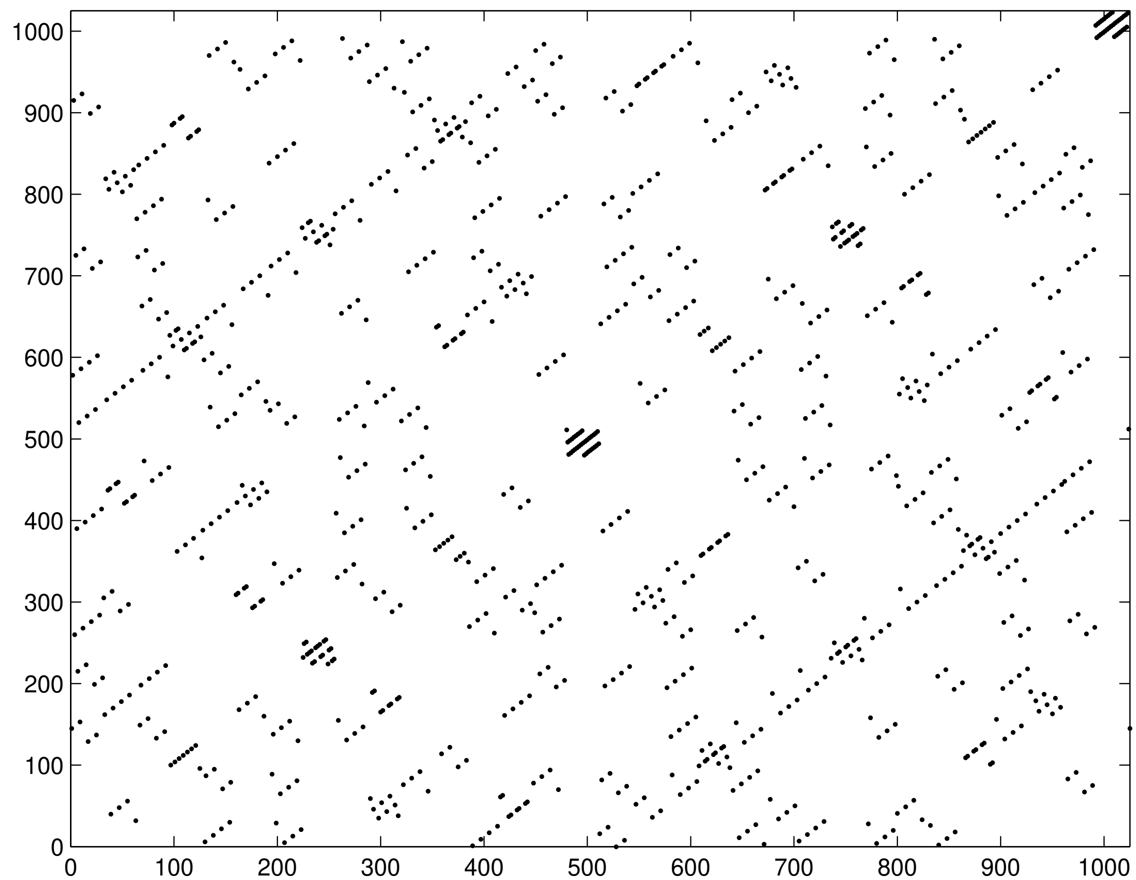


Figura 2.5: Entrelaçador de bloco, $N = 1024$, usado por Berrou-Glavieux.

Este entrelaçador, obtido conforme as expressões matemáticas anteriores para $N = 1024$ símbolos, $K = 8$ e $M = 128$, organiza seus elementos, mostrados na Figura 2.5, por meio de permutações mais deslocamentos cíclicos. Ou seja, o primeiro elemento, após entrelaçado, vai para a posição 128, o segundo vai para posição 577 e assim por diante.

Além do entrelaçador anterior usado por C. Berrou, A. Glavieux e P. Thitimajashima [2], [22] e [23], destacamos também neste trabalho o entrelaçador s-aleatório.

O entrelaçador s-aleatório [22] é gerado por uma permutação aleatória. Ou seja, um vetor ruído de comprimento N é gerado e a permutação que ordena este vetor ruído na ordem sorteada é usada para gerar o entrelaçador.

Este entrelaçador foi projetado considerando dois critérios importantes para o desempenho dos códigos turbo que são: as propriedades do espectro de distância do código e a redução da correlação entre a informação extrínseca e a entrada de dados a cada iteração [24]. Em outras palavras, o entrelaçador deve obedecer às propriedades de dispersão (separação) de seus elementos definidas a seguir.

Considere um entrelaçador de comprimento N tal que, para todo valor inteiro i, existe a permutação $\pi(i)$ que obedece as duas propriedades:

- Para todo i e j se $|i-j| \leq S_1$ então $|\pi(i)-\pi(j)| > S_1$,

- Para todo i e $\pi(i)$ temos $|i-\pi(i)| > S_2$,

onde $S_1 \triangleq \min\left(\rho, \dfrac{N}{\rho+1}\right)$ e $S_2 \triangleq \dfrac{\rho-1}{2}$ são números inteiros positivos pré-determinados e ρ é um número inteiro positivo tal que o mínimo múltiplo comum entre ρ e N é 1 e $\rho-1$ divide N.

A Figura 2.6 mostra o entrelaçador s-aleatório usado em nossas simulações. Este entrelaçador apresenta melhores propriedades sobre o espalhamento dos seus elementos que os outros entrelaçadores encontrados na literatura [23], [24].

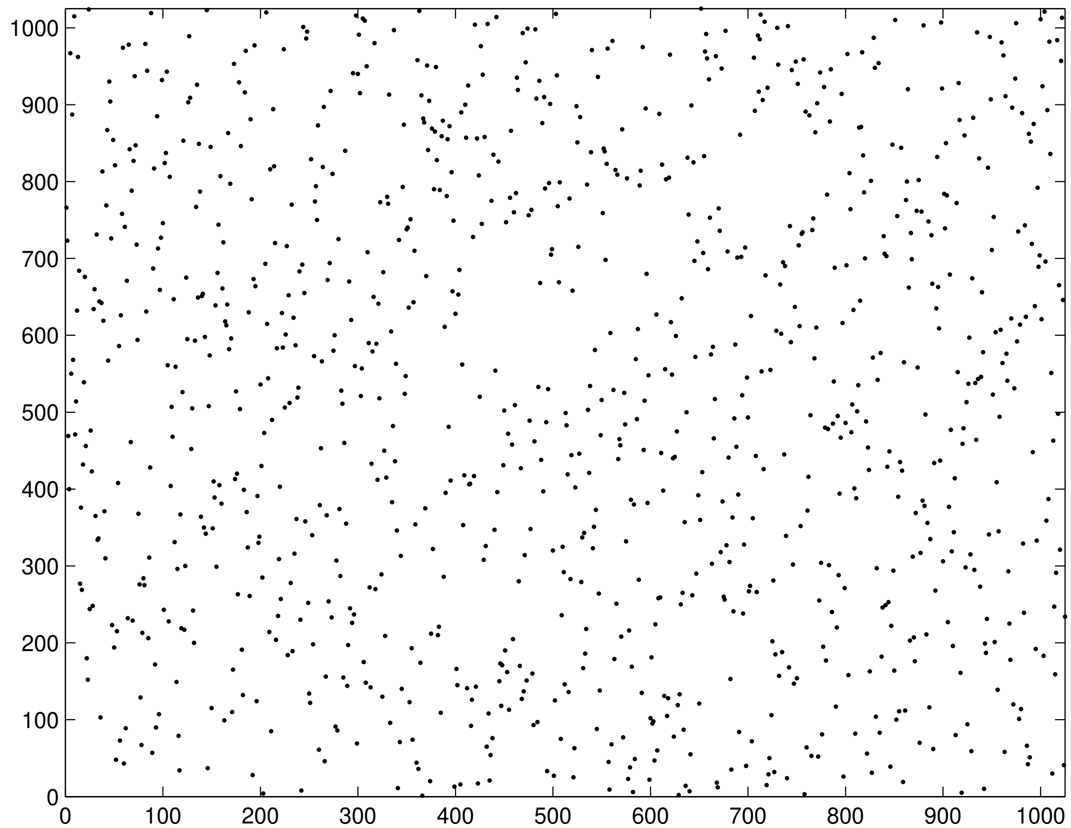


Figura 2.6: Entrelaçador s-aleatório de bloco ($N = 1024$).

2.3.3 Puncionador

O codificador turbo *M-ário* mostrado na Figura 2.2 possui 1 entrada e 3 saídas antes do puncionador e 1 entrada e 2 saídas depois do puncionador.

A função do puncionador é apagar periodicamente símbolos de redundância pré-selecionados dos codificadores, aumentando assim a taxa de codificação.

Exemplo 2.3.7 *Com base nos Exemplos (2.3.5) e (2.3.6), quando não é usado o puncionador, a taxa de codificação turbo é igual a* $1/3$ *e a seqüência código correspondente ao esquema turbo* *é dada por:*

$$x = \left| \begin{array}{cccccccccccccccccccccccccc} x^s & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 3 & 1 & 0 & 2 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 \\ x^{p1} & 2 & 1 & 3 & 0 & 3 & 0 & 3 & 2 & 1 & 1 & 2 & 0 & 3 & 1 & 2 & 2 & 0 & 1 & 2 & 2 & 2 & 1 & 3 & 3 & 2 \\ x^{p2} & 2 & 1 & 1 & 1 & 0 & 0 & 2 & 3 & 0 & 0 & 2 & 0 & 2 & 0 & 1 & 0 & 1 & 3 & 2 & 2 & 2 & 1 & 3 & 3 & 2. \end{array} \right.$$

E, quando é usado o puncionador, podem-se apagar por exemplo, os símbolos de paridade ímpares provenientes do codificador RSC-1 e os símbolos pares provenientes do codificador RSC-2, isto é,

$$x = \left| \begin{array}{cccccccccccccccccccccccccc} x^s & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 3 & 1 & 0 & 2 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 \\ x^{p1} & 2 & & 3 & & 3 & & 3 & & 1 & & 2 & & 3 & & 2 & & 0 & & 2 & & 2 & & 3 & & 2\,, \\ x^{p2} & & 1 & & 1 & & 0 & & 3 & & 0 & & 0 & & 0 & & 0 & & 3 & & 2 & & 1 & & 3 & \end{array} \right.$$

produzindo assim a seguinte seqüência código

$$x = \left| \begin{array}{cccccccccccccccccccccccccc} x^s & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 3 & 1 & 0 & 2 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 1 \\ x^{p} & 2 & 1 & 3 & 1 & 3 & 0 & 3 & 3 & 1 & 0 & 2 & 0 & 3 & 0 & 2 & 0 & 0 & 3 & 2 & 2 & 2 & 1 & 3 & 3 & 2 \end{array} \right.$$

na saída do esquema de codificação turbo da Figura 2.2. Note que este procedimento aumenta a taxa de codificação turbo de $1/3$ *para* $1/2$.

Conforme o exemplo anterior, quando usamos o puncionamento, são mapeados N símbolos de informação em $2N$ símbolos do código, enquanto que, sem o puncionamento, são mapeados N símbolos de informação em $3N$ símbolos do código.

2.3.4 Modulador

A seqüência de entrada do modulador é composta por símbolos pertencentes ao anel de inteiros $\mathbb{Z}_M$. Estes símbolos são transformados através da modulação *M-PSK* em sinais do tipo

$$S_i = A\exp\left(j\frac{2\pi i}{M}\right) = A\left(\cos\left(\frac{2\pi i}{M}\right) + j\mathrm{sen}\left(\frac{2\pi i}{M}\right)\right),$$

onde i é um símbolo do alfabeto *M-ário* $\{0,1,2,3,\cdots,M-1\} \in \mathbb{Z}_M$, $j = \sqrt{-1}$, e A é uma amplitude (raio) do sinal. Neste trabalho consideramos a amplitude do sinal $A = 1$. A Figura 2.7 mostra a constelação de sinais da modulação *M-PSK*.

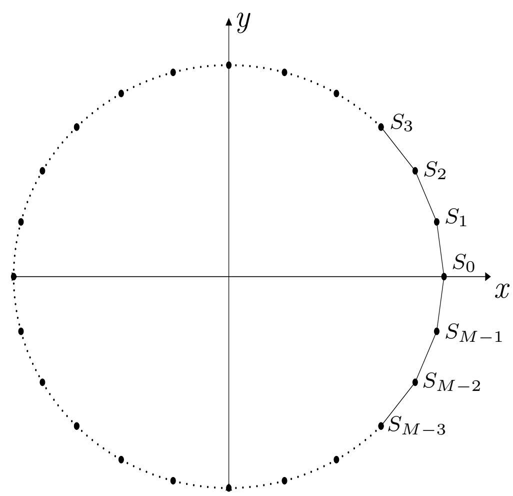


Figura 2.7: Constelação *M-PSK*

A distância euclidiana ao quadrado entre dois sinais modulados S_i e S_r é dada por

$$\begin{aligned} D_E^2(S_i, S_r) &= \|S_i - S_r\|^2 \\ &= \left\|\exp\left(j\frac{2\pi i}{M}\right) - \exp\left(j\frac{2\pi r}{M}\right)\right\|^2 \\ &= \left\|\exp\left(j\frac{2\pi(i-r)}{M}\right) - 1\right\|^2 \\ &= \left\|\exp\left(j\frac{2\pi(i\ominus r)}{M}\right) - 1\right\|^2, \end{aligned}$$

onde $\ominus$ denota a subtração módulo-M. Note que $i \ominus r$ representa uma operação definida pelos elementos i e r do anel de inteiros módulo-M, ou seja, i é adicionado módulo-M ao inverso aditivo de r. Desse modo a distância euclidiana ao quadrado entre dois pontos no espaço de sinais é dada pela expressão:

$$D_E^2(i,r) \triangleq \left\| \exp\left(j\frac{2\pi(i \ominus r)}{M}\right) - 1 \right\|^2.$$

E o peso euclidiano ao quadrado é definido por

$$W_E^2(i) \triangleq D_E^2(i,0) = \left\| \exp\left(j\frac{2\pi i}{M}\right) - 1 \right\|^2$$

Portanto, a distância euclidiana ao quadrado entre dois pontos do espaço de sinais é igual ao peso euclidiano ao quadrado do resultado da subtração módulo-M entre estes dois pontos, isto é,

$$D_E^2(i,r) = W_E^2(i \ominus r).$$

Observe que o índice i é equivalente a $S_i = \exp\left(j\frac{2\pi i}{M}\right)$. Logo, a relação fechada que existe entre o anel de inteiros módulo-M e a modulação *M-PSK*, permite o perfeito casamento dos símbolos do alfabeto *M-ário* com a modulação *M-PSK*.

A Figura 2.8 mostra constelações *M-PSK*, $M \in \{2, 3, \cdots, 7\}$, com mesma energia média, isto é, cada constelação está contida em uma circunferência com centro em $(0, 0)$ e raio igual a 1, no plano cartesiano $\mathbb{R} \times \mathbb{R}$. Nesta figura, os elementos internos a cada circunferência representam as respectivas distâncias euclidiana ao quadrado entre o símbolo *M-ário* 0 (zero) e os demais símbolos de cada constelação. Já os elementos externos a cada circunferência representam os respectivos símbolos *M-ários* das constelações *M-PSK*, $M \in \{2, 3, \cdots, 7\}$. Note que a medida que cresce o tamanho da constelação decresce a distância entre os símbolos não idênticos mais próximos e cresce a quantidade de informação associada a cada símbolo, ou seja, um símbolo *M-ário* corresponde a $\log_2(M)$ bits.

No artigo "Rate-1/2 Component Codes for Nonbinary Turbo Codes" apresentado por Andrew C. Reid e T. Aaron Gulliver [27], a distância euclidiana entre os símbolos não idênticos mais próximos em cada constelação *M-PSK* é a mesma, conseqüentemente a energia média das constelações *M-PSK* cresce com o crescimento da cardinalidade do alfabeto *M-ário*. Para ser justo com nossas comparações, as constelações *M-PSK* tem a mesma energia média e é usada a distância euclidiana em todos os sistemas deste trabalho.

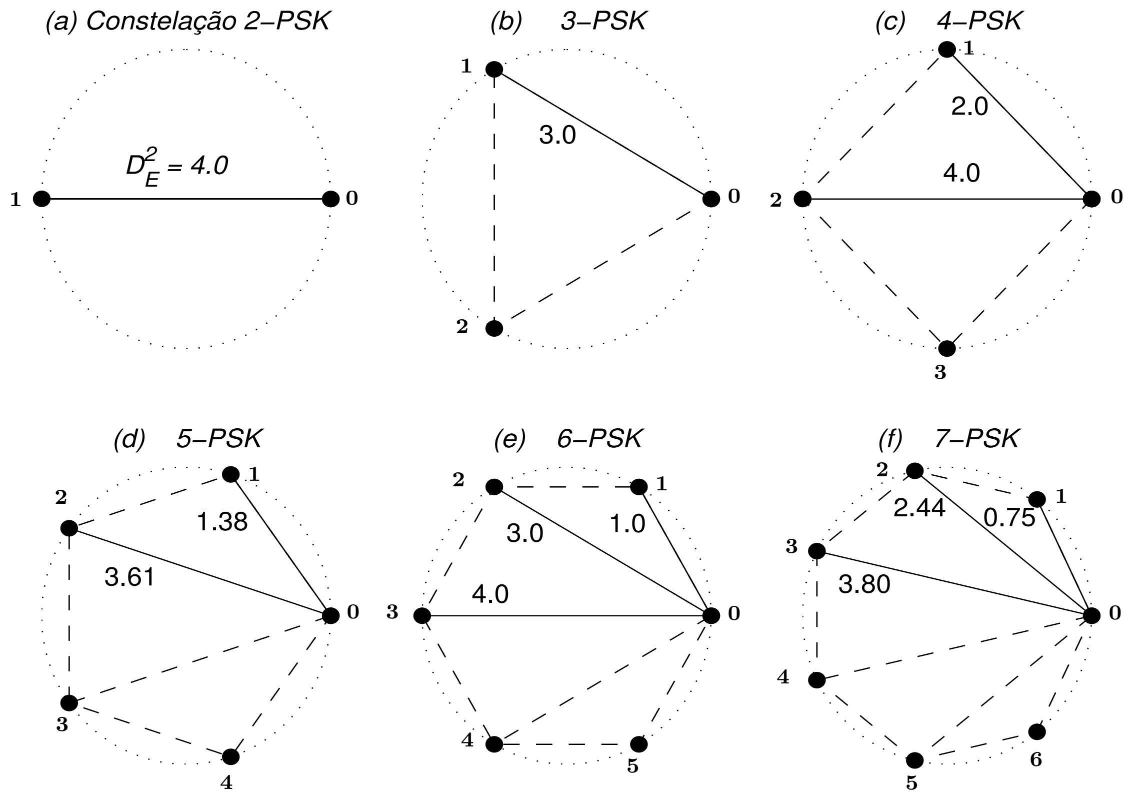


Figura 2.8: Variação da constelação *M-PSK*.

A distância euclidiana livre do código *RSC*, código linear, é o menor valor da distância euclidiana entre as seqüências v e a seqüência toda nula, isto é, $d_{free} = \min\{D_E(v,0)|u \neq 0\}$, onde v representa todas as seqüências de palavras código produzidas pelo codificador *RSC* com o processamento das seqüências de informação u. Além disso, o caminho gerado na treliça deste codificador com o processamento desta seqüência de informação u diverge e depois converge para estado todo zero.

O peso euclidiano livre do código *RSC* é a distância euclidiana entre a seqüência de informação u e a seqüência toda nula, $W_{free} = \{D_E(u,0)|u \neq 0\}$, onde u está associado a d_{free}.

A distância euclidiana livre D_{free}, do código referente ao codificador turbo com puncionamento, Figura 2.2, é igual a distância d_{free} do código *RSC* com taxa 1/2, onde d_{free} é formada pelo peso euclidiano da seqüência de informação W_{free}, mais a distância euclidiana correspondente à seqüência de paridade do código *RSC*. Já a distância D_{free}, do código referente ao codificador turbo sem puncionamento, é formada pelo peso euclidiano da seqüência de informação W_{free}, mais a distância euclidiana da seqüência de paridade do código *RSC-1*, mais a distância euclidiana da seqüência de paridade do código *RSC-2*.

De acordo com as conjecturas propostas para o sistema binário por Benedetto e Montorsi [26], além da distância euclidiana livre D_{free}, devemos analisar também a distância efetiva do código. Definimos por distância euclidiana efetiva D_{efet}, a distância correspondente ao peso euclidiano efetivo W_{efet}, onde W_{efet} é o menor peso da seqüência de informação u, $W_{efet} = \min(W_{free})$. A seqüência u proporciona a divergência e convergência do estado zero como já foi definido. Estas conjecturas também são utilizadas nos sistemas *M-ários*.

Nas Tabelas 3.1 a 3.4 destacamos os codificadores que possuem máxima distância euclidiana livre e máxima distância euclidiana efetiva.

A Tabela 2.4 mostra os polinômios das matrizes geradoras dos codificadores binários que proporcionam melhor desempenho. Note que $g_1(D)$ e $g_2(D)$ são os polinômios da matriz $g(D) = \left[1\ \frac{g_2(D)}{g_1(D)}\right]$, D_{free} é a distância euclidiana livre, W_{free} é o peso euclidiano livre, D_{efet} é a distância euclidiana efetiva e W_{efet} é o peso euclidiano efetivo. As distâncias de todas as tabelas deste capítulo são referentes ao codificador turbo com puncionamento, isto é, a taxa da codificação turbo é 1/2.

Estados	$g_1(D)$	$g_2(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
4	$1+D+D^2$	$1+D^2$	12	4	10	6
8	$1+D+D^3$	$1+D+D^2+D^3$	16	4	12	8
16	$1+D+D^4$	$1+D+D^3+D^4$	22	6	18	10
16	$1+D+D^4$	$1+D^2+D^3+D^4$	22	6	18	10
16	$1+D^2+D^4$	$1+D+D^3+D^4$	28	4	18	10

Tabela 2.4: Codificadores binários.

A Tabela 2.5 mostra os polinômios das matrizes geradoras dos codificadores ternários que proporcionam melhor desempenho.

Estados	$g_1(D)$	$g_2(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
9	$1+D+D^2$	$1+2D^2$	12.12	3.46	8.66	5.19
9	$1+D+D^2$	$2+D^2$	12.12	3.46	8.66	5.19
27	$1+D+D^2+D^3$	$2+D+2D^3$	13.85	5.19	12.12	6.92
27	$1+2D+D^2+D^3$	$1+2D^2+D^3$	13.85	5.19	12.12	6.92

Tabela 2.5: Codificadores ternários.

A Tabela 2.6 mostra os polinômios da matriz geradora dos codificadores quaternários que proporcionam melhor desempenho.

Estados	$g_1(D)$	$g_2(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
16	$1+D+3D^2$	$2+D+2D^2$	9.65	2.82	8.0	6.0
16	$1+2D+D^2$	$1+D^2$	11.65	2.82	7.65	4.82
16	$1+3D+3D^2$	$2+D+2D^2$	9.65	2.82	8.0	6.0
16	$1+3D+3D^2$	$3+2D^2$	9.65	2.82	7.65	4.24

Tabela 2.6: Codificadores quaternários.

A Tabela 2.7 mostra os polinômios da matriz geradora dos codificadores 5-ários que proporcionam melhor desempenho.

Estados	$g_1(D)$	$g_2(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
25	$1 + D + D^2$	$4 + 4D + 3D^2$	10.4	3.07	7.78	3.52
25	$1 + D + 2D^2$	$3 + D + D^2$	9.6	2.35	8.50	4.25
25	$1 + 4D + 1D^2$	$2 + 4D + D^2$	9.6	2.35	8.50	4.25
25	$1 + 4D + 4D^2$	$2 + 4D + 2D^2$	9.95	2.35	7.78	3.52

Tabela 2.7: Codificadores 5-ários.

2.4 Decodificação Turbo *M-ária*

Nesta seção, apresentamos o algoritmo de decodificação de máximo *a posteriori* - *MAP*, utilizado em cada decodificador componente do esquema de decodificação turbo *M-ários* . Este algoritmo entrega ao usuário final uma seqüência de informação com a menor taxa de erro versus a menor relação sinal ruído possível. O algoritmo calcula o valor do logaritmo da probabilidade *a posteriori*, fazendo uso de propriedades estatísticas e de conceitos de códigos de treliça [2], [3] e [8].

2.4.1 Algoritmo *MAP*

A função do algoritmo *MAP* é fornecer a informação *a posteriori* $L(u_k)$ na saída de cada decodificador. Então, seja

$$L(u_k) = \ln(p(u_k = \theta \mid y)), \tag{2.4}$$

o valor do logaritmo natural da probabilidade *a posteriori* de um símbolo decodificado ser $u_k = \theta \in \{0, 1, 2, ..., M - 1\}$, dado que a seqüência de símbolos recebida é $y = (y_1\ y_2\ \cdots\ y_{k-1}\ y_k\ y_{k+1}\ \cdots\ y_N)$. Note que M é a cardinalidade do alfabeto de entrada e N é o comprimento da seqüência de informação.

A probabilidade *a posteriori*, $p(u_k = \theta \mid y)$, pode ser expressa em função da probabilidade *a priori*, $p(u_k)$, e da probabilidade de transição da métrica do ramo $p(y_k \mid x_k)$, onde x_k é a palavra código transmitida referente ao símbolo de informação u_k e y_k é a palavra código recebida referente a palavra código transmitida x_k.

Proposição 2.4.1 *Se as transições entre o estado prévio,* $S_{k-1} = s'$*, e o estado presente,* $S_k = s$*, são mutuamente exclusivas (isto é, apenas uma delas pode ter ocorrido na treliça referente ao codificador RSC), então*

$$L(u_k) = \ln\left(\frac{\sum_{(s',s)} p(S_{k-1} = s', S_k = s, y)}{p(y)}\right), \tag{2.5}$$

onde (s', s) *é o conjunto de transições do estado prévio,* $S_{k-1} = s'$*, para o estado presente,* $S_k = s$*, que pode ocorrer se o símbolo de entrada* u_k *for igual a* θ*.*

Prova. Usando a regra de *Bayes*, $p(A, B) = p(A \mid B) \cdot p(B)$, e o fato de que, se os estados na treliça são conhecidos, o símbolo de entrada que causa a transição entre o estado prévio e o estado presente será conhecido. Assim, pela expressão (2.4) temos que:

$$\begin{aligned} L(u_k) &= \ln\left(p(u_k = \theta \mid y)\right) \\ &= \ln\left(\sum_{(s',s)} p(S_{k-1} = s', S_k = s \mid y)\right) \\ &= \ln\left(\sum_{(s',s)} \frac{p(S_{k-1} = s', S_k = s, y)}{p(y)}\right). \end{aligned}$$

■

Para simplificar a notação nas expressões, assumimos que

$$p(S_{k-1} = s', S_k = s, y) = p(s', s, y). \tag{2.6}$$

Assim, considerando na expressão (2.5) a expressão de probabilidades do numerador, a seqüência recebida $y = (y_1 \ y_2 \ \cdots \ y_{k-1} \ y_k \ y_{k+1} \ \cdots \ y_N)$ pode ser dividida em três partes:

Parte 1: a seqüência recebida antes da transição presente $y_1^{k-1} = (y_1 \ y_2 \ \cdots \ y_{k-1})$;

Parte 2: a palavra código transmitida associada a transição presente $y_k^k = y_k$;

Parte 3: a seqüência recebida depois da transição presente $y_{k+1}^N = (y_{k+1} \ y_{k+2} \ \cdots \ y_N)$.

Com base na divisão da seqüência y, podemos escrever

$$p(s', s, y) = p(s', s, y_1^{k-1}, y_k, y_{k+1}^N), \tag{2.7}$$

que pode ser expressa como o produto de três probabilidades, $\bar{\alpha}_{k-1}(s')$, $\gamma(s', s)$ e $\bar{\beta}_k(s)$, conforme demonstrado pelo teorema que sucede as definições a seguir:

Definição 2.4.2 *Seja*

$$\bar{\alpha}_{k-1}(s') = p(s', y_1^{k-1}), \tag{2.8}$$

a probabilidade de estar no estado prévio s' no tempo $k-1$ e ter recebido a seqüência de canal y_1^{k-1}.

Definição 2.4.3 *Seja*

$$\gamma_k\,(s, s') = p(\{y_k, s\} \mid s'), \tag{2.9}$$

a probabilidade de receber a seqüência de canal y_k e estar no estado presente s, no tempo k, dado que se estava no estado prévio s', no tempo $k-1$.

Definição 2.4.4 *Seja*

$$\bar{\beta}_k(s) = p(y_{k+1}^N \mid s), \tag{2.10}$$

a probabilidade de receber a seqüência futura de canal y_{k+1}^N, dado que se está no estado presente s, no tempo k.

Teorema 2.4.5 *Se o canal é sem memória e* $p(s', s, y) = p(s', s, y_1^{k-1}, y_k, y_{k+1}^N)$, *então*

$$p(s', s, y) = \bar{\alpha}_{k-1}(s') \cdot \gamma_k(s', s) \cdot \bar{\beta}_k(s). \tag{2.11}$$

Prova. Se o canal é sem memória, então a seqüência recebida y_{k+1}^N depende apenas do estado presente s e não do estado prévio s' ou das seqüências presente, y_k, e prévia, y_1^{k-1}. Desse modo, utilizando a regra de *Bayes*, temos que:

$$\begin{aligned}
p(s', s, y) &= p(s', s, y_1^{k-1}, y_k, y_{k+1}^N) \\
&= p(y_{k+1}^N \mid s', s, y_k, y_1^{k-1}) \cdot p(s', s, y_1^{k-1}, y_k) \\
&= p(y_{k+1}^N \mid s) \cdot p(s', s, y_1^{k-1}, y_k) \\
&= p(s', y_1^{k-1}) \cdot p(\{y_k, s\} \mid s', y_1^{k-1})\} \cdot p(y_{k+1}^N \mid s) \\
&= p(s', y_1^{k-1}) \cdot p(\{y_k, s\} \mid s')\} \cdot p(y_{k+1}^N \mid s) \\
&= \bar{\alpha}_{k-1}(s') \cdot \gamma_k(s', s) \cdot \bar{\beta}_k(s).
\end{aligned}$$

■

Finalmente, substituindo a expressão (2.11) na expressão (2.5), obtemos

$$L(u_k) = \ln \left(\frac{\sum_{(s,s')} \bar{\alpha}_{k-1}(s') \cdot \gamma_k(s', s) \cdot \bar{\beta}_k(s)}{p(y)} \right). \tag{2.12}$$

Na Figura 2.9, as linhas representam as transições entre os estados da treliça geradas pela entrada dos símbolos $\{0, 1, 2\}$, no codificador *RSC* ternário com matriz geradora $g(D) = \begin{bmatrix} 1 & \dfrac{1+2D}{1+D} \end{bmatrix}$.

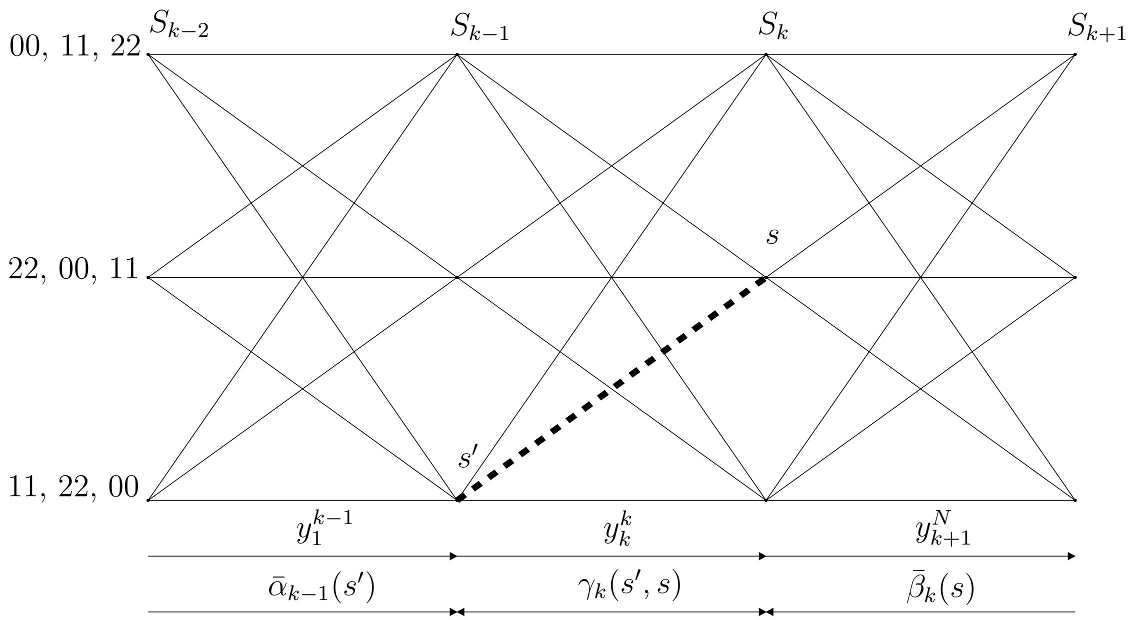


Figura 2.9: Treliça do decodificador *MAP* para o codificador *RSC* ternário com matriz geradora $g(D) = \left[1 \; \dfrac{1+2D}{1+D}\right]$.

Esta figura mostra a divisão da seqüência y e o significado das probabilidades $\bar{\alpha}_{k-1}(s')$, $\gamma_k\left(s', s\right)$ e $\bar{\beta}_k(s)$, para a transição na treliça do estado prévio, $S_{k-1} = s'$, para o estado presente, $S_k = s$, conforme é mostrado pela linha em destaque.

Expressão da Probabilidade Recursiva Direta $\bar{\alpha}_k(s)$

Proposição 2.4.6 *Se o canal é sem memória e* $\bar{\alpha}_{k-1}(s') = p(s', y_1^{k-1})$, *então*

$$\bar{\alpha}_k(s) = \sum_{todo\ s'} \bar{\alpha}_{k-1}(s') \cdot \gamma_k\left(s', s\right). \tag{2.13}$$

Prova. Usando as hipóteses dadas e a regra de *Bayes*, temos que

$$\begin{aligned}
\bar{\alpha}_k(s) &= p(s, y_1^k) \\
&= \sum_{todo\ s'} p(s, s', y_1^{k-1}, y_k) \\
&= \sum_{todo\ s'} p(\{s, y_k\} \mid \{s', y_1^{k-1}\}).p\left(s', y_1^{k-1}\right) \\
&= \sum_{todo\ s'} p(\{s, y_k\} \mid s').p\left(s', y_1^{k-1}\right) \\
&= \sum_{todo\ s'} \bar{\alpha}_{k-1}(s') \cdot \gamma_k\left(s', s\right).
\end{aligned}$$

■

Supondo que a treliça começa no estado inicial $S_0 = 0$, a condição inicial para esta recursão é:

$$\bar{\alpha}_0(S_0 = s) = \begin{cases} 1 & \text{para } s = 0 \\ 0 & \text{para } s \neq 0. \end{cases} \tag{2.14}$$

A Figura 2.10 mostra todas as transições que chegam e partem do estado $S_k = 0$, na treliça do codificador *RSC* ternário, cuja matriz geradora é $g(D) = \begin{bmatrix} 1 & \dfrac{1+2D}{1+D} \end{bmatrix}$. Diante disso, o cálculo da expressão recursiva direta $\bar{\alpha}_k(s)$, usando $\bar{\alpha}_{k-1}(s')$ e $\gamma_k\left(s', s\right)$, é:

$$\begin{aligned}
\bar{\alpha}_k(0) &= \sum_{s'=0}^{2} \bar{\alpha}_{k-1}(s') \cdot \gamma_k\left(s', 0\right) \\
&= \bar{\alpha}_{k-1}(0) \cdot \gamma_k\left(0, 0\right) + \bar{\alpha}_{k-1}(1) \cdot \gamma_k\left(1, 0\right) + \bar{\alpha}_{k-1}(2) \cdot \gamma_k\left(2, 0\right).
\end{aligned}$$

Expressão da Probabilidade Recursiva Reversa $\bar{\beta}_k(s)$

Proposição 2.4.7 *Se o canal é sem memória e $\bar{\beta}_k(s') = p(y_{k+1}^N \mid s')$, então*

$$\bar{\beta}_{k-1}(s') = \sum_{todo\ s} \bar{\beta}_k(s) \cdot \gamma_k\left(s', s\right). \tag{2.15}$$

Prova. Usando uma derivação similar a demonstração da expressão de $\bar{\alpha}_k(s)$, (2.13), temos que:

$$
\begin{aligned}
\bar{\beta}_{k-1}(s') &= p\left(y_k^N \mid s'\right) \\
&= \sum_{todo\ s} \bar{\beta}_k(s) \cdot \gamma_k\left(s', s\right).
\end{aligned}
$$

■

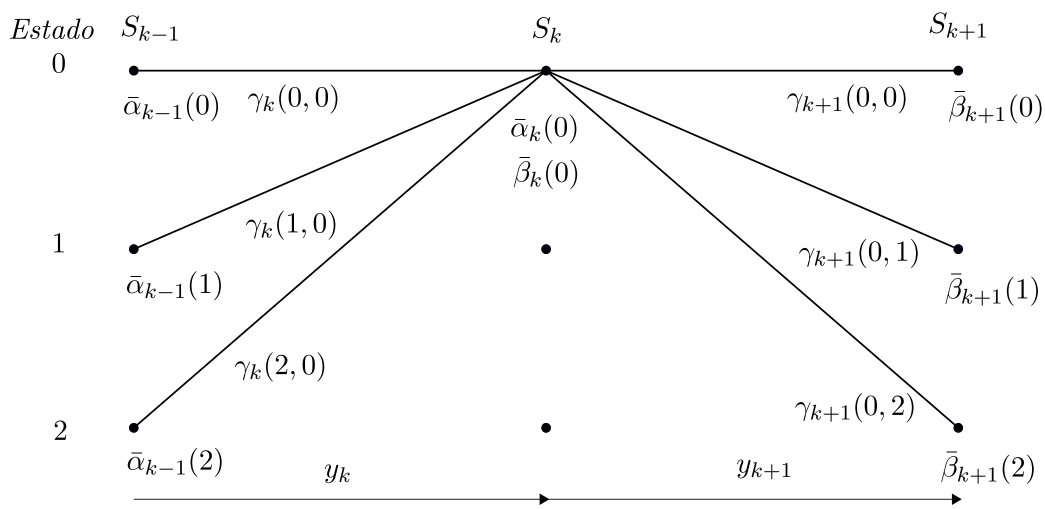


Figura 2.10: Treliça das expressões recursivas $\bar{\alpha}_k(0)$ e $\bar{\beta}_k(0)$.

Supondo que o codificador sempre volta ao estado $S_N = 0$, a condição inicial para esta recursão é:

$$
\bar{\beta}_N(S_N = s) = \begin{cases} 1 & \text{para } s = 0 \\ 0 & \text{para } s \neq 0. \end{cases} \tag{2.16}
$$

Com base na Figura 2.10, calculamos também a expressão recursiva reversa $\bar{\beta}_k(0)$, a partir de $\bar{\beta}_{k+1}(s)$ e $\gamma_{k+1}(0,s)$, do seguinte modo:

$$\begin{aligned}
\bar{\beta}_k(0) &= \sum_{s=0}^{2} \bar{\beta}_{k+1}(s) \cdot \gamma_{k+1}(0,s) \\
&= \bar{\beta}_{k+1}(0) \cdot \gamma_{k+1}(0,0) + \bar{\beta}_{k+1}(1) \cdot \gamma_{k+1}(0,1) + \bar{\beta}_{k+1}(2) \cdot \gamma_{k+1}(0,2) .
\end{aligned}$$

Expressões Recursivas Finais para $\alpha_k(s)$ e $\beta_k(s)$

Voltando à expressão (2.12), verificamos que, se utilizarmos o divisor $p(y)$, caminhamos para um algoritmo que não converge para o limitante de *Shannon* [11] e [19].

Assim, para construir o processo de decodificação iterativo turbo [2],[11] e [19], usamos o artifício de retirar da seqüência recebida y o elemento de ordem k, caso contrário, haverá sobreposição da informação extrínseca com a informação *a priori* . Então, substituímos $p(y)$ por $\dfrac{p(y)}{p(y_k)}$, dando origem ao seguinte corolário.

Corolário 2.4.8 *Se o canal é sem memória e* $p(y) = p\left(y_1^k, y_{k+1}^N\right)$ *então*

$$\frac{p(y)}{p(y_k)} = p\left(y_1^{k-1}\right) \cdot p\left(y_{k+1}^N \mid y_1^k\right) . \tag{2.17}$$

Prova. Usando a regra de *Bayes* temos que:

$$\begin{aligned}
\frac{p(y)}{p(y_k)} &= \frac{p\left(y_1^k, y_{k+1}^N\right)}{p(y_k)} \\
&= \frac{p\left(y_{k+1}^N \mid y_1^k\right) \cdot p\left(y_1^k\right)}{p(y_k)} \\
&= p\left(y_1^{k-1}\right) \cdot p\left(y_{k+1}^N \mid y_1^k\right) .
\end{aligned}$$

■

Teorema 2.4.9 *Se substituirmos* $p(y)$ *por* $\dfrac{p(y)}{p(y_k)}$, *então*

$$L(u_k) = \ln\left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s)\right), \tag{2.18}$$

onde $\alpha_k(s)$ *e* $\beta_k(s)$ *são as novas probabilidades modificadas definidas como:*

$$\begin{aligned} \alpha_k(s) &\triangleq \frac{\bar{\alpha}_k(s)}{p\left(y_1^k\right)}; \\ \beta_k(s) &\triangleq \frac{\bar{\beta}_k(s)}{p\left(y_{k+1}^N \mid y_1^k\right)}. \end{aligned} \tag{2.19}$$

Prova. Pelas expressões (2.12) e (2.17), temos que

$$\begin{aligned} L(u_k) &= \ln\left(\sum_{(s',s)} \frac{\bar{\alpha}_{k-1}(s') \cdot \gamma_k(s',s) \cdot \bar{\beta}_k(s)}{p(y)}\right) \\ &= \ln\left(\sum_{(s',s)} \frac{\bar{\alpha}_{k-1}(s) \cdot \gamma_k(s',s) \cdot \bar{\beta}_k(s) \cdot p(y_k)}{p(y)}\right) \\ &= \ln\left(\sum_{(s',s)} \frac{\bar{\alpha}_{k-1}(s')}{p\left(y_1^{k-1}\right)} \cdot \gamma_k(s',s) \cdot \frac{\bar{\beta}_k(s)}{p\left(y_{k+1}^N \mid y_1^k\right)}\right) \\ &= \ln\left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s)\right). \end{aligned}$$

■

Deste modo, as expressões recursivas finais de $\alpha_k(s)$ e $\beta_k(s)$ são dadas pelas proposições seguintes.

Proposição 2.4.10 *Se o canal é sem memória e* $p\left(y_1^k\right) = \sum_{todo\ s} \bar{\alpha}_k(s)$, *então*

$$\alpha_k(s) = \frac{\sum_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k(s',s)}{\sum_{todo\ s} \sum_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k(s',s)}. \tag{2.20}$$

Prova. Pelas hipóteses dadas e pelas expressões (2.19) e (2.13), segue-se que:

$$\begin{aligned}
\alpha_k(s) &= \frac{\bar{\alpha}_k(s)}{p\left(y_1^k\right)} \\
&= \frac{\bar{\alpha}_k(s)}{\sum\limits_{todo\ s} \bar{\alpha}_k(s)} \\
&= \frac{\sum\limits_{todo\ s'} \bar{\alpha}_{k-1}(s') \cdot \gamma_k\left(s', s\right)}{\sum\limits_{todo\ s}\sum\limits_{todo\ s'} \bar{\alpha}_{k-1}(s') \cdot \gamma_k\left(s', s\right)} \\
&= \frac{\sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s\right)}{\sum\limits_{todo\ s}\sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s\right)}.
\end{aligned}$$

■

Proposição 2.4.11 *Se o canal é sem memória e* $p\left(y_k^N \mid y_1^{k-1}\right) = \dfrac{p\left(y_1^k\right) \cdot p\left(y_{k+1}^N \mid y_1^k\right)}{p\left(y_1^{k-1}\right)}$, *então*

$$\beta_{k-1}(s) = \frac{\sum\limits_{todo\ s} \beta_k(s) \cdot \gamma_k(s', s)}{\sum\limits_{todo\ s}\sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s'\right)}. \tag{2.21}$$

Prova. Pelas expressões (2.19), (2.15) e (2.13), e pela expressão da hipótese dada, temos que:

$$\begin{aligned}
\beta_{k-1}(s) &= \frac{\bar{\beta}_k(s)}{p\left(y_k^N \mid y_1^{k-1}\right)} \\
&= \frac{\sum\limits_{todo\ s} \bar{\beta}_k(s) \cdot \gamma_k(s', s) \cdot p\left(y_1^{k-1}\right)}{p\left(y_1^k\right) \cdot p\left(y_{k+1}^N \mid y_1^k\right)} \\
&= \frac{\sum\limits_{todo\ s} \beta_k(s) \cdot \gamma_k(s', s) \cdot p\left(y_1^{k-1}\right) \cdot p\left(y_{k+1}^N \mid y_1^k\right)}{\sum\limits_{todo\ s}\sum\limits_{todo\ s'} \bar{\alpha}_{k-1}(s') \cdot \gamma_k\left(s', s\right) \cdot p\left(y_{k+1}^N \mid y_1^k\right)} \\
&= \frac{\sum\limits_{todo\ s} \beta_k(s) \cdot \gamma_k(s', s) \cdot p\left(y_1^{k-1}\right)}{\sum\limits_{todo\ s}\sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s\right) \cdot p\left(y_1^{k-1}\right)} \\
&= \frac{\sum\limits_{todo\ s} \beta_k(s) \cdot \gamma_k(s', s)}{\sum\limits_{todo\ s}\sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s\right)}.
\end{aligned}$$

■

Observe que $\alpha_k(s)$ e $\beta_k(s)$ são calculados recursivamente pelas equações (2.20) e (2.21) e possuem as mesmas condições iniciais que seus complementos naturais, dados em (2.14) e (2.16) para $\bar{\alpha}_k(s)$ e $\bar{\beta}_k(s)$, respectivamente.

Finalmente, calculamos agora a probabilidade de transição do estado prévio s' para o estado presente s, $\gamma_k\left(s', s\right)$.

Expressão da Probabilidade de Transição $\gamma_k\left(s', s\right)$

Corolário 2.4.12 *Se o canal é sem memória e* $\gamma_k\left(s, s'\right) = p(\{y_k, s\} \mid s')$, *então*

$$\gamma_k(s', s) = p(y_k \mid x_k) \cdot p(u_k). \tag{2.22}$$

Prova. Usando a regra de *Bayes* temos que:

$$\begin{aligned}
\gamma_k(s', s) &= p(s, y_k \mid s') \\
&= \frac{p(s, y_k, s')}{p(s')} \\
&= \frac{p(y_k \mid s, s') \cdot p(s', s)}{p(s')} \\
&= \frac{p(y_k \mid s, s') \cdot p(s \mid s') \cdot p(s')}{p(s')} \\
&= p(y_k \mid x_k) \cdot p(u_k),
\end{aligned}$$

onde:

u_k - símbolo de entrada necessário para causar (realizar) a transição na treliça do estado prévio, s', para o estado presente, s;

x_k - palavra código transmitida associada com esta transição;

y_k - palavra código recebida associada com a palavra código transmitida x_k;

$p(u_k)$ - probabilidade *a priori* deste símbolo.

■

Assim, a probabilidade de transição, $\gamma_k(s', s)$, é dada pelo produto da probabilidade *a priori* pela função densidade de probabilidade:

- $p(u_k)$ - probabilidade *a priori*. É a probabilidade do símbolo de entrada u_k necessário para realizar a transição. $p(u_k) = \dfrac{1}{M}$, onde M é o tamanho do alfabeto *M-ário*.
- $p(y_k \mid x_k)$ - função densidade de probabilidade. É a probabilidade de se receber a seqüência do canal y_k dado que a palavra código x_k associada com esta transição foi transmitida. Esta probabilidade é obtida a partir do sinal recebido, y_k, e dos possíveis símbolos transmitidos.

Desse modo, apresentamos as expressões $\alpha_{k-1}(s')$, $\beta_k(s)$ e $\gamma_k(s', s)$, necessárias para calcular a informação *a posteriori*.

2.4.2 Resumo do Algoritmo *MAP*

A Figura 2.11 apresenta de forma mais clara as expressões descritas no algoritmo MAP, usadas para calcular a informação *a posteriori* $L(u_k)$, ou ainda, para decodificar a seqüência recebida do canal y. Estas expressões serão organizadas da seguinte maneira:

1- A função densidade de probabilidade, $p(y_k \mid x_k)$, é usada juntamente com a probabilidade *a priori*, $p(u_k)$, para calcular a probabilidade de transição, $\gamma_k(s', s)$, de acordo com a expressão (2.22). Em seguida, calculamos as expressões das probabilidades recursivas $\alpha_k(s)$ e $\beta_k(s)$.

2- A partir da probabilidade de transição, $\gamma_k(s', s)$, dada na expressão (2.22), e da condição inicial $\alpha_k(0)$, dada na expressão (2.14), calculamos a expressão da probabilidade recursiva direta $\alpha_{k-1}(s')$, dada na expressão (2.20).

3- Com o uso da probabilidade de transição, $\gamma_k(s', s)$, dada na expressão (2.22), da expressão da probabilidade recursiva direta $\alpha_{k-1}(s')$, dada na expressão (2.20), e da condição inicial $\beta_k(0)$, dada na expressão (2.16), calculamos a expressão da probabilidade recursiva reversa $\beta_k(s)$, dada na expressão (2.21).

4- Finalmente, com o uso das expressões $\alpha_{k-1}(s')$, $\gamma_k(s', s)$ e $\beta_k(s)$, dadas nas expressões (2.20), (2.22) e (2.21), respectivamente, calculamos a informação *a posteriori* $L(u_k)$, dada na expressão (2.18), como explicitado na Figura 2.11.

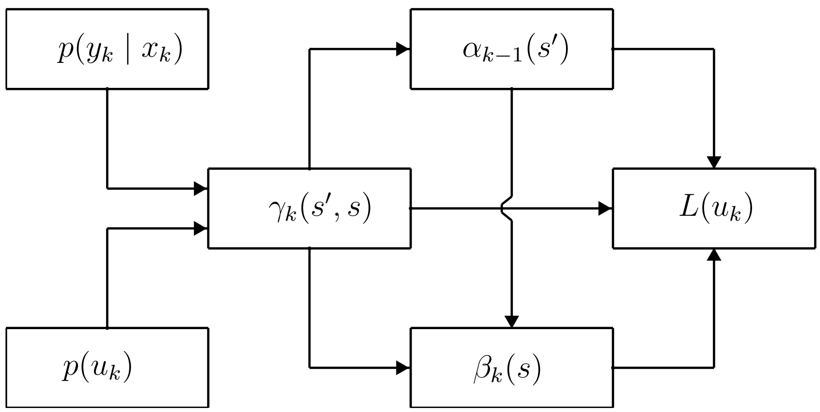


Figura 2.11: Resumo das principais expressões usadas no algoritmo *MAP*.

2.5 Decodificação Iterativa Turbo M-ária

Nesta seção apresentamos o processo de decodificação iterativa turbo *M-ário*, sobre $\mathbb{Z}_M$. Este processo de decodificação, esquematizado na Figura 2.12, é composto por dois decodificadores componentes, dois entrelaçadores e um desentrelaçador. Sua função é calcular a informação extrínseca na saída do estágio de decodificação anterior e usá-la como informação *a priori* na entrada do próximo estágio de decodificação, para reduzir a probabilidade de erro de símbolo (ou bit) à medida em que for crescendo o número de iterações.

Os conceitos de distância euclidiana e o fato do código convolucional ser sistemático são usados na probabilidade de transição $\gamma_k(s', s)$ para expressar a informação *a posteriori*, como a soma da informação *a priori*, da informação sistemática $\exp\left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0}\right)$ e da informação extrínseca, calculadas em cada estágio da decodificação da Figura 2.12.

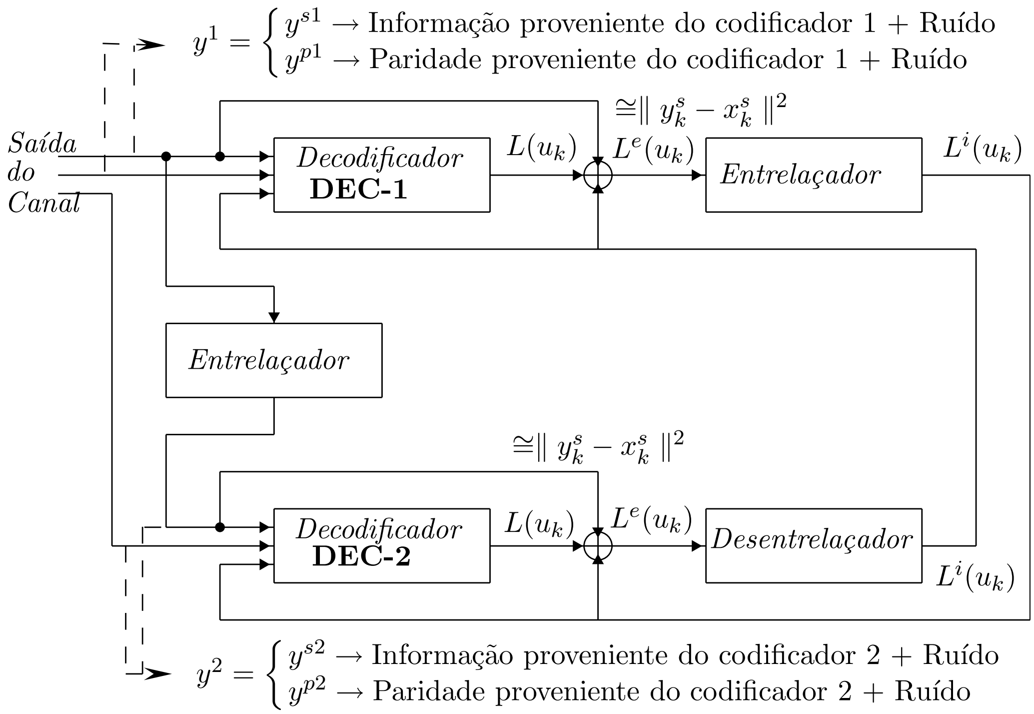


Figura 2.12: Decodificação iterativa turbo.

2.5.1 Um Estágio da Decodificação Turbo

Começamos expressando a função densidade de probabilidade em função dos símbolos de informação e dos símbolos de paridade e, em seguida, deduzimos as expressões da informação intrínseca, extrínseca e *a posteriori*.

Sendo x_k a palavra código transmitida no instante k e n_k o vetor ruído cujas componentes são variáveis aleatórias gaussianas com média zero e variância $\sigma^2 = \frac{N_0}{2}$ (ruído aditivo gaussiano branco), a palavra código recebida no instante k é dada por $y_k = x_k + n_k$.

Assumimos que cada codificador componente do esquema de codificação turbo tem taxa 1/2, a palavra código transmitida no instante k é dada por $x_k = x_k^s\ x_k^p = u_k\ x_k^p$, e a palavra código recebida no instante k é dada por $y_k = y_k^s\ y_k^p$.

Assim, se o canal é sem memória e gaussiano, então, condicionados aos símbolos transmitidos, os símbolos recebidos possuem distribuição gaussiana com função densidade de probabilidade dada por

$$p(y_k \mid x_k) \quad = \quad p(y_k^s \mid x_k^s) \cdot p(y_k^p \mid x_k^p), \tag{2.23}$$

onde

x_k^s - símbolo sistemático da palavra código transmitida no instante k, x_k;

x_k^p - símbolo de paridade da palavra código transmitida no instante k, x_k;

y_k^s - símbolo (sinal) recebido correspondente ao símbolo transmitido no instante k, x_k^s;

y_k^p - símbolo (sinal) recebido correspondente ao símbolo transmitido no instante k, x_k^p.

Considerando que a modulação *M-PSK* é utilizada para a transmissão dos símbolos *M-ários* pelo canal de comunicação, os símbolos da palavra código transmitida (x_k^s e x_k^p) e da palavra código recebida (y_k^s e y_k^p), são pontos do plano cartesiano $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$.

Assim, a função densidade de probabilidade da expressão (2.23) é dada por:

$$\begin{aligned} p(y_k \mid x_k) &= p(y_k^s \mid x_k^s) \cdot p(y_k^p \mid x_k^p) \\ &= \frac{1}{\pi N_0} \cdot \exp\left(\frac{- \parallel y_k^s - x_k^s \parallel^2}{N_0}\right) \cdot \exp\left(\frac{- \parallel y_k^p - x_k^p \parallel^2}{N_0}\right), \end{aligned} \tag{2.24}$$

onde N_0 é a variância do ruído.

Aplicando os conceitos de informação intrínseca e extrínseca usados por C. Berrou, A. Glavieux e P. Thitimajashima [2] nas expressões que compõem a informação *a posteriori*, $L(u_k)$, esta informação pode ser obtida através do algoritmo MAP, pelo cálculo dos seguintes termos:

1. $L^i(u_k)$ - informação intrínseca ou informação *a priori*;

2. $\exp\left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0}\right)$ - informação sistemática;

3. $L^e(u_k)$ - informação extrínseca.

Se o canal é gaussiano, sem memória e a modulação é M-PSK, então substituindo a expressão (2.24) na expressão

$$\gamma_k(s, s') \;\; = \;\; p(y_k \mid x_k) \cdot p(u_k)$$

temos:

$$\gamma_k(s, s') = \exp\left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0}\right) \cdot \gamma_k^e(s, s') \cdot p(u_k),$$

onde

$$\gamma_k^e(s, s') = \frac{1}{\pi N_0} \cdot \exp\left(\frac{-\parallel y_k^p - x_k^p \parallel^2}{N_0}\right).$$

Teorema 2.5.1 *Se o canal é gaussiano, sem memória e a modulação é M-PSK, então a informação a posteriori é dada por:*

$$L(u_k) = L^i(u_k) + \left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0}\right) + L^e(u_k), \tag{2.25}$$

onde

$$L^i(u_k) = \ln(p(u_k))$$

é informação intrínseca, e

$$L^e(u_k) = \ln\left(\sum_{(s,s')} \alpha_{k-1}(s') \cdot \beta_k(s) \cdot \gamma_k^e(s, s')\right)$$

é a informação extrínseca.

Prova. Aplicando o resultado anterior na expressão final da informação *a posteriori* (2.18), temos que:

$$
\begin{aligned}
L(u_k) &= \ln\left(\sum_{(s,s')} \alpha_{k-1}(s')\cdot\beta_k(s)\cdot\gamma_k(s,s')\right)\\
&= \ln\left(\sum_{(s,s')} \alpha_{k-1}(s')\cdot\beta_k(s)\cdot\exp\left(\frac{-\parallel y_k^s - x_k^s\parallel^2}{N_0}\right)\gamma_k^e(s,s')\cdot p(u_k)\right)\\
&= \ln\left(\sum_{(s,s')} \alpha_{k-1}(s')\cdot\beta_k(s)\cdot\gamma_k^e(s,s')\right) + \left(\frac{-\parallel y_k^s - x_k^s\parallel^2}{N_0}\right) + \ln\left(p(u_k)\right)\\
&= L_e(u_k) + \left(\frac{-\parallel y_k^s - x_k^s\parallel^2}{N_0}\right) + L^i(u_k).
\end{aligned}
$$

■

Com base no que foi apresentado poderemos expor com maiores detalhes o significado dos termos:

Informação *a priori* - ou informação intrínseca sobre um símbolo u_k, é a informação conhecida antes de iniciar a decodificação da seqüência de símbolos recebida na entrada do decodificador;

Informação *extrínseca* - em contraste com a referenciada informação intrínseca, a informação extrínseca sobre o símbolo u_k é a informação fornecida por um decodificador, baseado na seqüência de símbolos recebida e na informação *a priori*. Ou seja, com base na expressão (2.25), a informação extrínseca é calculada do seguinte modo: informação *a posteriori*, $L(u_k)$, menos a informação sistemática, $\left(\frac{-\parallel y_k^s - x_k^s\parallel^2}{N_0}\right)$, menos a informação *a priori* $L^i(u_k)$.

Informação *a posteriori* - a informação *a posteriori* sobre um símbolo é a informação que o decodificador fornece levando em conta toda informação que entra no decodificador sobre o símbolo u_k.

A partir da informação *a posteriori* $L(u_k)$, obtemos a informação extrínseca, de acordo com a expressão (2.25), para usar no estágio de decodificação seguinte.

2.5.2 Processo de Decodificação Iterativa

No início desta seção, vimos como se realiza um estágio da decodificação, isto é, como se obtém as expressões da informação *a posteriori*, da informação *a priori* e da informação extrínseca. A seguir, apresentaremos o processo de decodificação iterativa turbo *M-ário*. Este processo baseia-se no cálculo da informação extrínseca do estágio de decodificação anterior e usá-la como informação *a priori* no outro estágio de decodificação para que, à medida em que o número de iterações for crescendo, diminua-se a probabilidade de erro.

Assim, note que a seqüência recebida do canal y pode ser organizada em duas partes, como mostrado na Figura 2.13.

$y^1 = y^{s1}, y^{p1}$ **- Seqüência proveniente do codificador *RSC-1*** - recebida pelo primeiro decodificador componente, contém a versão recebida dos símbolos sistemáticos, y^{s1}, e dos símbolos de paridade, y^{p1}, provenientes do primeiro codificador;

$y^2 = y^{s2}, y^{p2}$ **- Seqüência proveniente do codificador *RSC-2*** - recebida pelo segundo decodificador componente, contém a versão entrelaçada dos símbolos sistemáticos, y^{s2}, e dos símbolos de paridade, y^{p2}, provenientes do segundo codificador.

De acordo com o esquema de decodificação mostrado na Figura 2.13, se a decodificação ocorreu no primeiro decodificador, basta entrelaçar a informação extrínseca, obtida na saída deste decodificador, para obter a informação *a priori* a ser usada no segundo decodificador. Entretanto, se a decodificação ocorreu no segundo decodificador, devemos desentrelaçar a informação extrínseca, obtida na saída deste decodificador, para obter a informação *a priori* a ser usada no primeiro decodificador.

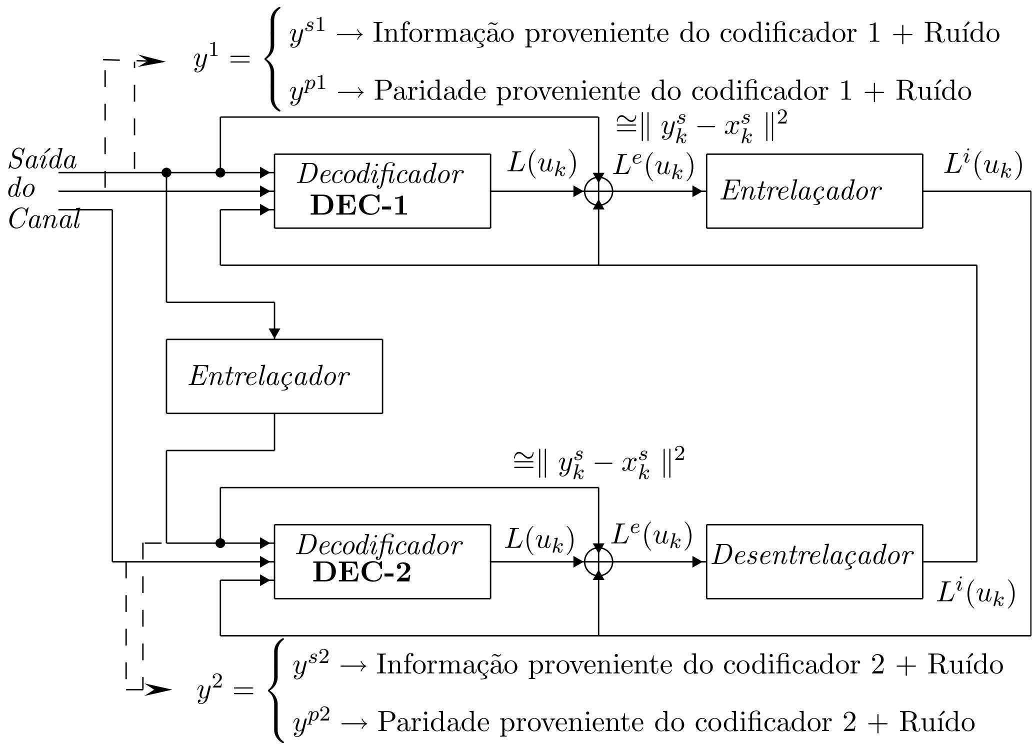


Figura 2.13: Decodificação iterativa turbo.

Processo Iterativo

Considere inicialmente o primeiro decodificador **DEC-1** componente na primeira iteração. Este decodificador recebe a seqüência de canal, y^1, e produz uma estimativa da informação *a posteriori* $L_{11}(u_k)$ dos símbolos de informação u_k, onde $k \in \{1, \cdots, N\}$ e N é o comprimento da seqüência de informação. Note que o subscrito 11 de $L_{11}(u_k)$ indica a primeira iteração do primeiro decodificador. Nesta primeira iteração a informação *a priori* recebida pelo primeiro decodificador componente é $\ln(p(u_k = \theta)) = \ln(1/M)$.

O segundo decodificador **DEC-2** componente recebe a seqüência de canal y^2 junto com a informação extrínseca entrelaçada do primeiro decodificador componente e fornece uma estimativa da informação *a posteriori* $L_{12}(u_k)$ dos símbolos de informação u_k, onde o subscrito 12 de $L_{12}(u_k)$ indica a primeira iteração do segundo decodificador componente. Note que a informação extrínseca entrelaçada, utilizada pelo segundo decodificador componente é, na verdade, a informação *a priori* obtida com o entrelaçamento da expressão: informação *a posteriori*, $L_{11}(u_k)$, menos a informação *a priori*, $\ln(p(u_k = \theta)) = \ln(1/M)$ e menos a informação sistemática, $\left(\dfrac{- \parallel y_k^s - x_k^s \parallel^2}{N_0}\right)$, todas do primeiro decodificador componente. Esta informação *a priori*(usada no segundo decodificador) é calculada através da expressão (2.25) na saída do primeiro decodificador componente.

Na segunda iteração, o primeiro decodificador componente processa novamente sua seqüência recebida de canal y^1, levando-se em conta a informação *a priori* $L^i(u_k)$, fornecida pela porção extrínseca $L^e(u_k)$ da informação *a posteriori* $L_{12}(u_k)$. Esta informação extrínseca é fornecida pelo segundo decodificador componente, na primeira iteração. Note que $L_{21}(u_k)$ é a informação *a posteriori* na segunda iteração do primeiro decodificador componente.

Ainda na segunda iteração, o segundo decodificador componente usa a informação *a posteriori* $L_{21}(u_k)$ do primeiro decodificador para obter a informação *a priori* $L^i(u_k)$. Esta informação *a priori* é usada junto com a seqüência recebida de canal y^2 para calcular a informação *a posteriori* $L_{22}(u_k)$.

Este processo iterativo continua, sendo que a cada iteração, a taxa de erro de bit média diminui para uma mesma relação sinal ruído - SNR (*signal to noise ratio*).

Exemplo 2.5.2 *Considere a seqüência de informação com* $N = 100$ *símbolos distribuídos através das linhas da Tabela 2.8. Esta seqüência, depois de ser processada pelo esquema de codificação turbo da Figura 2.2 (para o codificador RSC com matriz geradora* $g(D) = \left[1 \;\; \dfrac{2 + D + 2D^2}{1 + D + 3D^2}\right]$ *e taxa de codificação turbo* $1/2$*), passa pelo canal que pode introduzir erros devido ao ruído, produzindo, neste exemplo, a seqüência de informação da Tabela 2.9 na saída do primeiro decodificador componente do esquema de decodificação turbo da Figura 2.13. Esta seqüência do primeiro decodificador componente na primeira iteração, contém* 23 *erros entre os* 100 *símbolos decodificados e estes erros estão destacados na tabela referenciada. Já a* *seqüência da Tabela 2.10, obtida na saída do segundo decodificador componente e na primeira iteração, contém 16 erros apenas.*

Na segunda iteração, as seqüências das Tabelas 2.11 e 2.12, obtidas nas saídas do primeiro e do segundo decodificador componente, contêm 15 erros e 11 erros, respectivamente.

Finalmente, a Tabela 2.13 apresenta a seqüência que contém apenas 1 erro em 100 símbolos decodificados, na saída do primeiro decodificador componente e na terceira iteração. Enquanto a Tabela 2.14, apresenta a seqüência sem erro para 100 símbolos decodificados na saída do segundo decodificador componente e na terceira iteração.

Seqüência de informação a ser transmitida																			
1	2	1	2	3	1	0	2	3	1	2	2	2	3	2	0	1	1	3	3
3	1	3	1	2	2	0	3	2	2	1	2	1	1	3	1	0	0	0	3
3	2	1	2	1	1	1	3	2	1	2	1	1	2	3	3	1	2	1	0
2	1	2	0	1	2	2	3	1	1	2	3	1	0	0	1	1	3	1	2
1	3	2	2	1	3	2	2	2	1	0	1	1	0	0	2	0	2	3	0

Tabela 2.8: Seqüência de informação de comprimento $N = 100$ símbolos.

Primeiro decodificador - primeira iteração																			
1	2	1	3	3	1	0	2	3	1	2	2	2	3	2	0	1	1	3	1
2	1	3	1	2	2	1	3	2	2	0	2	2	1	0	1	0	0	0	0
3	2	2	2	1	1	0	2	2	1	2	1	1	1	3	0	0	2	1	0
2	1	3	0	1	2	2	3	2	1	2	3	2	0	1	1	1	2	3	2
1	3	2	2	1	3	2	2	2	1	0	0	3	0	0	2	1	2	3	0

Tabela 2.9: 100 Símbolos decodificados - 23 erros ocorridos.

Segundo decodificador - primeira iteração																			
1	2	1	3	3	1	0	2	3	1	2	2	2	3	2	0	1	1	3	1
2	1	3	1	2	2	1	3	2	2	1	2	1	1	3	1	0	0	0	3
3	2	2	2	1	1	0	3	2	1	2	1	1	2	3	3	0	2	1	0
2	1	3	0	1	2	2	3	1	1	2	3	1	3	1	0	0	2	1	2
1	3	2	2	1	3	2	2	2	1	0	0	3	0	0	2	1	2	3	0

Tabela 2.10: 100 Símbolos decodificados - 16 erros ocorridos.

Primeiro decodificador - segunda iteração																			
2	2	1	3	3	1	0	2	3	1	2	2	2	3	2	0	1	1	3	1
2	3	2	1	1	2	2	3	2	2	1	2	1	1	3	1	0	0	0	3
3	2	1	2	1	1	1	3	2	1	2	1	1	2	3	3	1	2	1	0
2	1	2	0	1	2	2	3	1	1	2	3	1	3	1	0	1	2	1	2
1	3	2	2	1	3	2	2	2	1	0	0	3	0	0	2	1	2	3	0

Tabela 2.11: 100 Símbolos decodificados - 15 erros ocorridos.

Segundo decodificador - segunda iteração																			
1	2	1	3	3	1	0	2	3	1	2	2	2	3	2	0	1	1	3	1
2	3	3	1	1	2	0	3	2	2	1	2	1	1	3	1	0	0	0	3
3	2	1	2	1	1	1	3	2	1	2	1	1	2	3	3	1	2	1	0
2	1	2	0	1	2	2	3	1	1	2	3	1	3	3	0	1	3	1	2
1	3	2	2	3	3	2	2	2	1	0	0	3	0	0	2	0	2	3	0

Tabela 2.12: 100 Símbolos decodificados - 11 erros ocorridos.

Primeiro decodificador - terceira iteração																			
1	2	1	2	3	1	0	2	3	1	2	2	2	3	2	0	1	1	3	3
3	1	3	1	1	2	0	3	2	2	1	2	1	1	3	1	0	0	0	3
3	2	1	2	1	1	1	3	2	1	2	1	1	2	3	3	1	2	1	0
2	1	2	0	1	2	2	3	1	1	2	3	1	0	0	1	1	3	1	2
1	3	2	2	1	3	2	2	2	1	0	1	1	0	0	2	0	2	3	0

Tabela 2.13: 100 Símbolos decodificados - 1 erro ocorrido.

Segundo decodificador - terceira iteração																			
1	2	1	2	3	1	0	2	3	1	2	2	2	3	2	0	1	1	3	3
3	1	3	1	2	2	0	3	2	2	1	2	1	1	3	1	0	0	0	3
3	2	1	2	1	1	1	3	2	1	2	1	1	2	3	3	1	2	1	0
2	1	2	0	1	2	2	3	1	1	2	3	1	0	0	1	1	3	1	2
1	3	2	2	1	3	2	2	2	1	0	1	1	0	0	2	0	2	3	0

Tabela 2.14: 100 Símbolos decodificados - sem erro.

O exemplo apresentado aqui mostra a eficiência do processo iterativo, ou seja, como a confiabilidade sobre o símbolo decodificado melhora a cada passo da decodificação, ou como a taxa de erro de bit diminui a cada iteração.

Assim, o processo de decodificação iterativa torna o algoritmo de decodificação *MAP* bastante eficiente na correção de erros dos símbolos decodificados, proporcionando, em média, uma diminuição na probabilidade de erro de símbolo (ou *bit*) a cada iteração. A Figura 2.14, apresenta os pontos referentes à taxa de erro de bit em 1.1735×10^{-1}, 5.6122×10^{-2} e 5.1020×10^{-3} para $E_b/N_0 = 1.25 \;\; dB$, correspondentes à primeira, segunda e terceira iteração, respectivamente.

Portanto, o processo de decodificação iterativa é considerado a principal ferramenta do algoritmo de decodificação turbo, e é um dos mais eficientes na correção de erros para uma baixa relação sinal ruído.

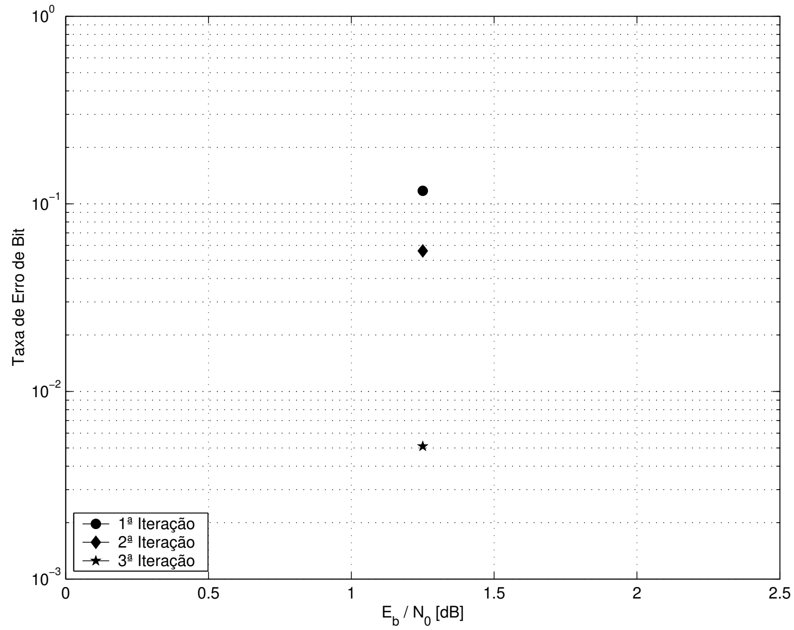


Figura 2.14: Efeito da variação do número de iterações para $N = 100$ símbolos ou $N = 200$ *bits*, taxa de codificação turbo 1/2 e $g(D) = \left[1 \quad \dfrac{2 + D + 2D^2}{1 + D + 3D^2}\right]$, sobre a curva da *taxa de erro de bit* $\times$ E_b/N_0.

2.6 Conclusão

Neste capítulo estabelecem-se critérios para analisar o desempenho do sistema de comunicação digital para códigos turbo multiníveis definidos sobre os campos e anéis de inteiros $\mathbb{Z}_M$. O tratamento teórico referente à codificação turbo *M-ária* e à modulação *M-PSK* é para avaliar os códigos que proporcionam maior eficiência no processo de decodificação turbo. Já o tratamento teórico do algoritmo de decodificação *MAP* é essencial na implementação e compreensão do processo de decodificação iterativa turbo *M-ário*. Finalmente, a estrutura teórica dos códigos turbo multiníveis nos permite avaliar as características de cada sistema com relação a complexidade de decodificação, capacidade de correção de erro, cardinalidade do alfabeto, largura de banda e mapeamento de símbolos para bits.

Capítulo 3

Resultados dos Códigos Turbo Multiníveis

3.1 Introdução

Neste capítulo, apresentamos as curvas da taxa de erro de bit em termos da energia por bit por densidade espectral de potência (E_b/N_0) referentes às classes obtidas para códigos turbo *M-ário*, definidos sobre os campos e anéis de inteiros. As simulações foram realizadas sobre o esquema turbo *M-ário* visando avaliar sistemas que possuam menor complexidade e melhor desempenho. A variação de alguns parâmetros, tais como o comprimento da seqüência de informação, a taxa do codificador turbo, o codificador RSC, o tamanho do alfabeto e o número de iterações estão diretamente associados com o desempenho e a complexidade dos códigos turbo.

3.2 Resultados

Os codificadores convolucionais constituintes são codificadores RSC idênticos, com taxa de codificação igual a 1/2, com matriz geradora do tipo $g(D) = \left[1 \; \frac{g_2(D)}{g_1(D)}\right]$, e organizados conforme o esquema de codificação turbo mostrado na Figura 3.1. O aumento do número de memórias nos codificadores proporciona uma melhora no desempenho do sistema. Entretanto, este aumento causa um aumento exponencial no número de estados na treliça e conseqüentemente um aumento da complexidade do sistema, já que o número de estados na treliça de um codificador *RSC* *M-ário* com q memórias é M^q.

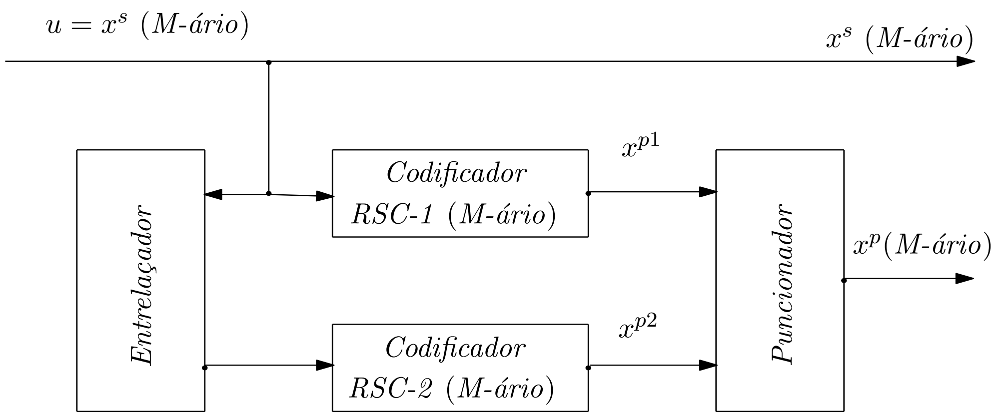


Figura 3.1: Esquema de codificação turbo.

O entrelaçador, usado na concatenação paralela dos dois codificadores *RSC-1* e *RSC-2* do esquema de codificação da Figura 3.1, evita que o erro ocorrido em uma dada posição do símbolo associado ao codificador *RSC-1*, ocorra na mesma posição do símbolo associado ao codificador *RSC-2*. Desse modo, a presença do entrelaçador no processo de codificação produz uma maior eficiência no processo de decodificação iterativa. Escolhemos para as simulações o entrelaçador s-aleatório, pois este apresenta melhores propriedades aleatórias.

O puncionador descarta alguns símbolos de paridade na saída do esquema de codificação turbo através de uma matriz de apagamento previamente selecionada. Nas simulações realizadas, o codificador turbo terá taxa 1/2 quando usado o puncionamento, e terá taxa 1/3 quando não usado o puncionamento.

O ruído do canal é aditivo gaussiano branco ($AWGN$) com densidade espectral de potência unilateral igual a N_0 e a modulação é *M-PSK*. O fato de o canal ser $AWGN$ e do codificador ser *RSC* facilita o cálculo da informação extrínseca usada na construção do processo de decodificação iterativa. Note que se o canal não fosse *AWGN* também seria possível obter a informação extrínseca.

Os decodificadores são de máximo *a posteriori*, MAP, como apresentado no esquema de decodificação iterativa turbo *M-ário* da Figura 3.2. Estes decodificadores atuam de forma conjunta e iterativa. Isto é, inicialmente, o primeiro decodificador fornece a informação extrínseca que é usada no segundo decodificador, então, este segundo decodificador produz uma nova informação extrínseca que será usada no primeiro decodificador e assim por diante. A cada iteração deste procedimento o decodificador turbo fornece, em média, uma melhor informação *a posteriori* sobre o símbolo a ser decodificado.

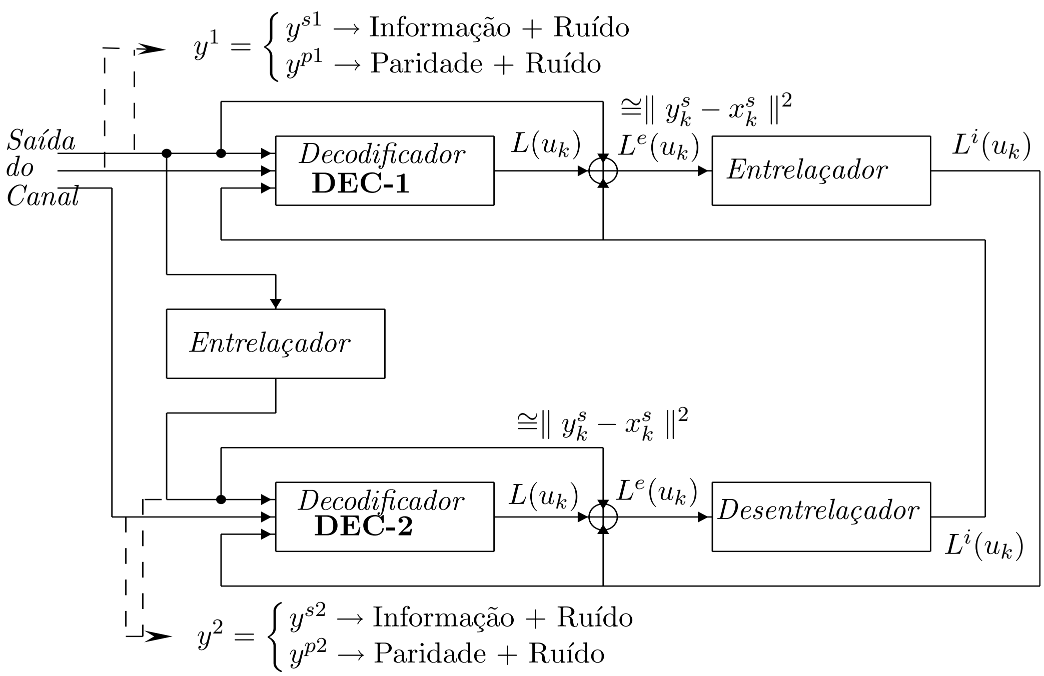


Figura 3.2: Decodificação iterativa turbo.

A cardinalidade do alfabeto dá mais flexibilidade à combinação: largura de banda, desempenho e complexidade. Em geral, o aumento na cardinalidade M do alfabeto proporciona maior eficiência ao sistema turbo, pois um símbolo *M-ário* corresponde a $\log_2(M)$ bits. Entretanto, este aumento causa um aumento exponencial da complexidade e quando M cresce, isto causa também uma diminuição da distância mínima entre os símbolos da constelação *M-PSK* com mesma energia média.

O desempenho do sistema turbo com respeito a variação dos parâmetros cardinalidade do alfabeto, tamanho da seqüência de informação, número de estados e número de iterações, é obtido por simulação computacional através das curvas da probabilidade de erro de bit média versus a relação sinal ruído $\left(P_e(b) \times \dfrac{E_b}{N_0}\right)$.

3.2.1 Esquema Turbo Binário com Modulação *B-PSK*

Como já existem resultados na literatura ([2], [4]) sobre o sistema turbo binário, apresentamos nesta seção os resultados simulados que serão usados como referência para efeito de comparação com os resultados dos sistemas turbo ternário, quaternário e 5-ário, apresentados nas próximas seções.

A Figura 3.3 apresenta as curvas da probabilidade $P_e(b)$ de erro de bit versus E_b/N_0, para os esquemas turbo binários que possuem taxa de codificação turbo igual a 1/2, comprimento da seqüência de informação $N = 2000$ *bits*, codificadores RSC com matrizes geradoras $g(D) = \left[1 \;\; \dfrac{1+D+D^2+D^3}{1+D+D^3}\right]$ (treliça com $2^3 = 8$ estados) e $g(D) = \left[1 \;\; \dfrac{1+D+D^3+D^4}{1+D+D^4}\right]$ (treliça com $2^4 = 16$ estados) e número de iterações igual a 8.

A curva de desempenho, em termos de probabilidade de erro de bit média versus E_b/N_0, referente ao codificador cuja treliça possui 16 estados, apresenta melhor desempenho do que a referente ao codificador de 8 estados, para $E_b/N_0 > 0.5\ dB$. O tempo gasto pela decodificação no esquema com 16 estados é 2 vezes maior que no esquema com 8 estados.

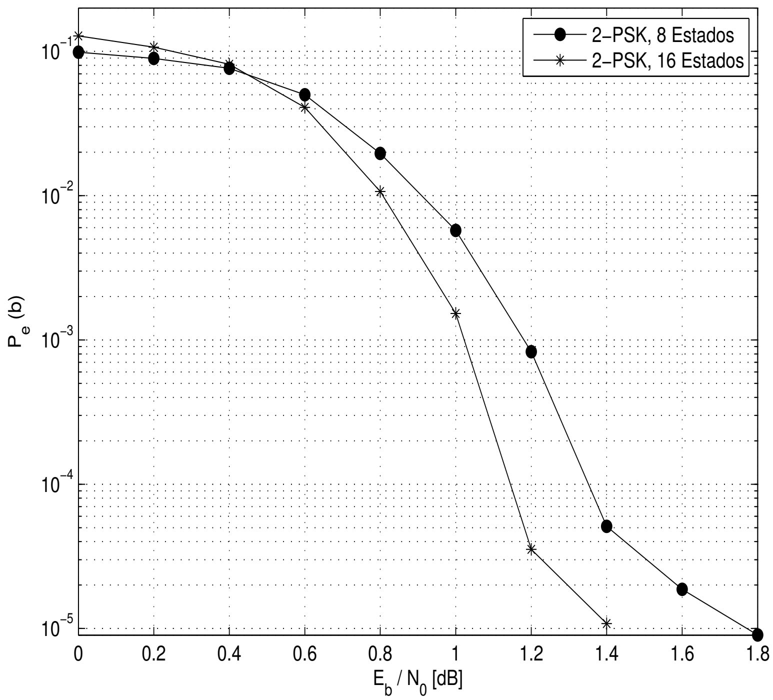


Figura 3.3: Desempenho dos códigos turbo para taxa de codificação turbo igual a 1/2 e comprimento da seqüência de informação $N = 2000$ bits.

3.2.2 Esquema Turbo Ternário com Modulação *3-PSK*

Nesta seção analisamos o desempenho e a complexidade do esquema turbo ternário com relação ao esquema turbo binário. Nos sistemas *M-ários*, $M > 2$, as probabilidades de erro de símbolo média são divididas por $\log_2(M)$ para obtermos aproximadamente as probabilidades de erro de bit média versus E_b/N_0. Estes sistemas processam N símbolos *M-ários* e são comparados com sistemas binários que processam a mesma quantidade de informação em bits, isto é, $N \cdot \log_2(M)$ bits.

A Figura 3.4 apresenta as curvas da taxa de erro de bit versus E_b/N_0 para o esquema turbo ternário que possui taxa de codificação turbo 1/2, codificadores RSC com matrizes geradoras $g(D) = \left[1 \ \dfrac{1+2D^2}{1+D+D^2}\right]$ (treliça com 9 estados) e $g(D) = \left[1 \ \dfrac{2+D+2D^2+2D^3}{1+D+D^2+2D^3}\right]$ (treliça com 27 estados), comprimento da seqüência de informação $N = 1261$ símbolos e o número de iterações igual a 8.

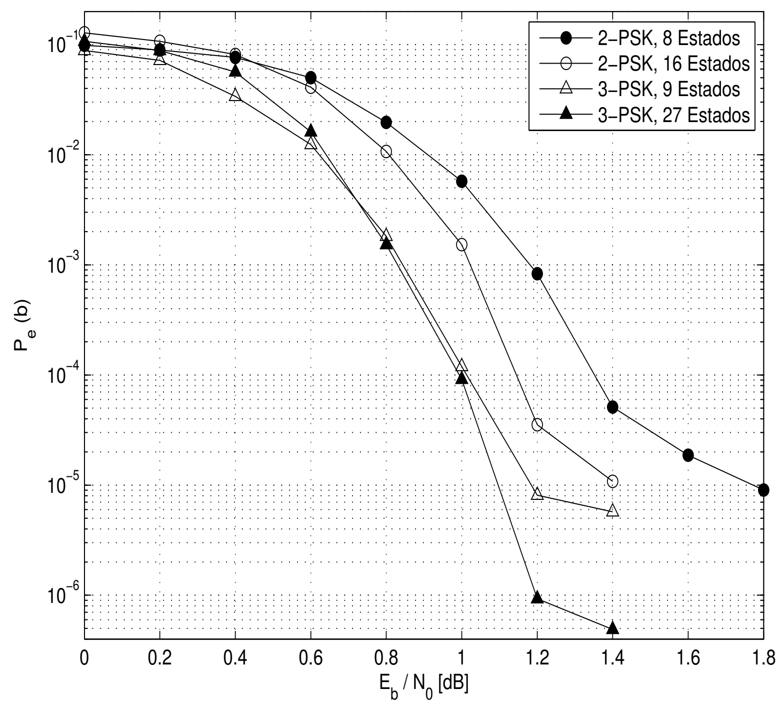


Figura 3.4: Desempenho dos códigos turbo binários e ternários para taxa de codificação turbo 1/2 e comprimento da seqüência de informação $N = 2000$ bits.

Como esperado, a curva de desempenho em termos de probabilidade de erro de bit média versus E_b/N_0, referente ao codificador cuja treliça possui 27 estados, apresenta melhor desempenho do que a curva do codificador de 9 estados, para $E_b/N_0 > 0.7\ dB$. Além disso, para $E_b/N_0 \geq 1.2\ dB$ a probabilidade de erro de bit do codificador de 27 estados é pelo menos 10 vazes menor que a probabilidade do codificador de 9 estados. Por outro lado, o tempo gasto na decodificação do esquema de 27 estados, é 2.19 vezes maior do que o tempo no esquema de 9 estados.

Logo, o esquema ternário com 9 estados tem bom desempenho e baixa complexidade de decodificação. Já o esquema ternário com 27 estados tem maior complexidade de decodificação e melhor desempenho que o esquema referente a 9 estados. Note também o *"erro floor"* mais baixo no esquema com 27 estados. *"Error floor"* é a possibilidade de previsão de um eventual patamar “intransponível” de erro de bit. Este fenômeno que ocorre a partir de um certo valor de relação sinal-ruído, onde as palavras código de baixo peso começam a dominar o desempenho do esquema de codificação de canal, não permitindo que a taxa de erro de bit seja reduzida de forma significativa com o aumento da relação sinal-ruído [28].

A Figura 3.4 também mostra o desempenho dos esquemas binário e ternário juntos. O esquema ternário com 9 estados, tem ganhos maiores do que 0.3 dB e 0.2 dB, para $P_e(b) < 10^{-2}$, quando comparado com esquemas binários com 8 e 16 estados, respectivamente. Além disso, o tempo gasto na decodificação do esquema ternário com 9 estados é 1.19 e 2.33 vezes menor que o tempo dos esquemas binários com 8 e 16 estados, respectivamente. Note que o sistema ternário processa $N = 1261$ símbolos em vez de $N = 2000$ bits como o sistema binário. Denotamos como ganho o valor absoluto da diferença entre as duas razões $E_b/N_0(1)$ e $E_b/N_0(2)$, calculada para a mesma taxa de erro de bit.

Observe que o número de estados do codificador binário, 2^l, é diferente do número de estados do codificador ternário, 3^m, para todo l e m, onde l e m são números inteiros positivos que representam o número de memória de cada codificador. Como o número de estados de cada codificador influencia tanto no desempenho quanto na complexidade do sistema turbo, a comparação do sistema binário com o ternário não é muito justa, mas é uma indicação de que os codificadores ternários proporcionam bom desempenho.

Logo, o sistema ternário com modulação *3-PSK*, apresenta melhor desempenho, menor complexidade de decodificação e ocupa menos largura de faixa que o sistema binário com modulação *2-PSK*. Ou seja, este sistema ternário transporta 1.58 vezes mais informação do que o sistema binário, tornando-se assim mais eficiente quando aplicado a sistemas de comunicação que possuam largura de faixa limitada. Todavia, este sistema não possui um perfeito mapeamento da fonte binária para os símbolos ternários.

3.2.3 Esquema Turbo Quaternário com Modulação *4-PSK*

A Figura 3.5 apresenta as curvas da taxa de erro de bit versus E_b/N_0 para o esquema turbo quaternário que possui taxa de codificação turbo 1/2, codificador *RSC* com matriz geradora $g(D) = \left[1 \; \frac{2+D+2D^3}{1+D+3D^2}\right]$ (treliça com 16 estados), comprimento da seqüência de informação, $N = 1000$ *símbolos* e o número de iterações na decodificação igual a 8.

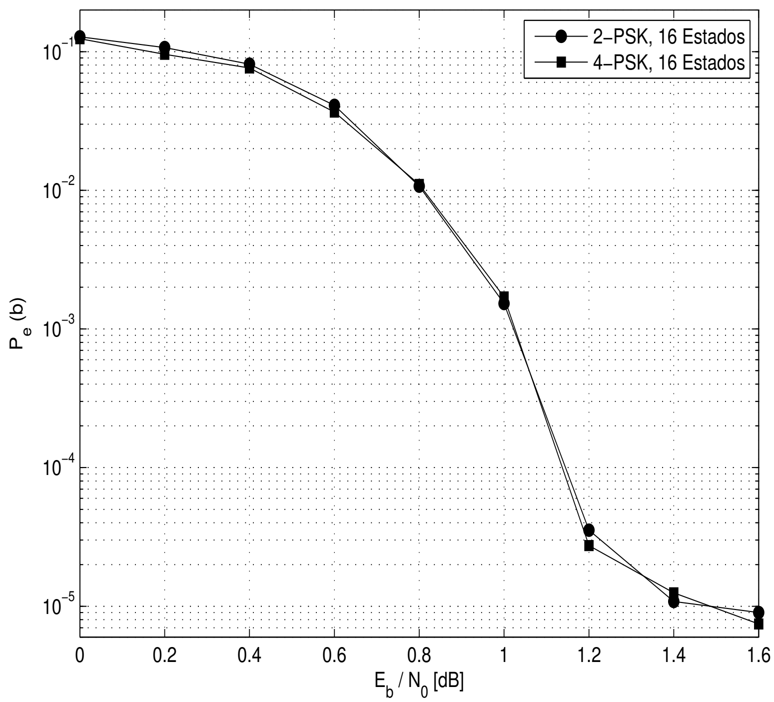


Figura 3.5: Desempenho dos códigos turbo binários e quaternários para taxa de codificação turbo 1/2 e comprimento da seqüência de informação $N = 2000$ bits.

O codificador quaternário apresenta desempenho equivalente ao desempenho do codificador binário com 16 estados e o esquema turbo quaternário processa $N = 1000$ símbolos em vez de $N = 2000$ bits como o sistema binário.

Logo, o esquema quaternário com modulação *4-PSK* gasta metade do tempo de decodificação em relação ao esquema binário com modulação *2-PSK*. Ou seja, este esquema quaternário processa o dobro da quantidade de informação do esquema binário no mesmo tempo de decodificação, artigo "Códigos Turbo Quaternários" apresentado em [4],

3.2.4 Esquema Turbo 5-ário com Modulação *5-PSK*

Vamos analisar nesta seção o desempenho, a complexidade, o mapeamento de fonte e a largura de banda do esquema 5-ário com relação ao esquema binário. Este esquema realiza o processamento com $N = 2000/\log_2(5)$ símbolos e 5 transições associadas a cada estado, enquanto que o esquema binário realiza o processamento com $N = 2000$ bits e 2 transições associadas a cada estado. Além disso, a probabilidade de erro de símbolo é dividida por $\log_2(5)$ para aproximar-se da probabilidade de erro de bit, pois neste esquema, não existe um perfeito mapeamento de bits para símbolos 5-ários.

A Figura 3.6 apresenta a curva da taxa de erro de bit média versus E_b/N_0 para o esquema turbo 5-ário que possui taxa de codificação turbo 1/2, codificador *RSC* com matriz geradora $g(D) = \left[1 \; \frac{4+4D+3D^3}{1+D+D^2}\right]$ (treliça com 25 estados), comprimento da seqüência de informação $N = 861$ símbolos e o número de iterações no decodificador igual a 8.

A curva referente ao codificador 5-ário encontra-se acima da curva do codificador binário com 8 estados. O tempo gasto na decodificação do esquema 5-ário é 1,3 vezes maior e a quantidade de informação transmitida é 2,3 vezes maior do que no esquema binário.

Portanto, para um sistema que possua largura de faixa limitada, o esquema 5-ário é mais vantajoso que o esquema binário. Além disso, o esquema *5-ário* apresenta um "error floor" mais baixo.

O número de estados dos codificadores 5-ários são as potências de 5, desse modo, os codificadores com 1 memória, 5 estados, não proporcionam bom desempenho e os codificadores com 2 memórias, 25 estados, apresentam o desempenho discutido acima. Já para os codificadores com 3 memórias, 125 estados, a simulação para obtenção da probabilidade de erro é impraticável devido à complexidade no algoritmo de decodificação.

Para sistemas turbo *M-ários* em que $M > 7$ e o número de memórias nos codificadores componentes é maior do que 2, a simulação para obtenção da probabilidade de erro torna-se impraticável devido o tempo gasto pelo algoritmo de decodificação. Note que um codificador *8-ário* com 2 memórias possui $8^2 = 64$ estados e com 3 memórias possui $8^3 = 512$ estados.

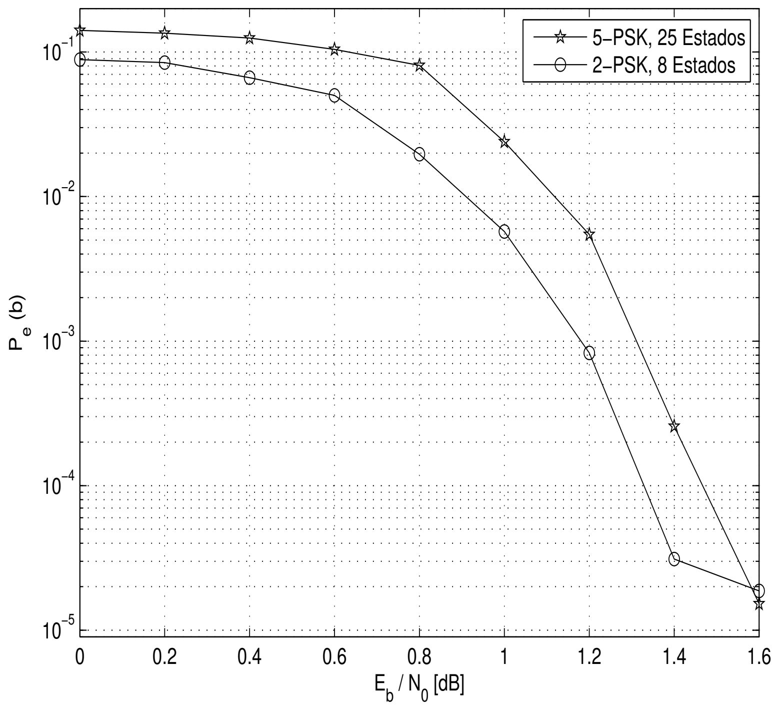


Figura 3.6: Desempenho dos códigos turbo binários e 5-ários para taxa de codificação turbo 1/2 e comprimento da seqüência de informação $N = 2000$ bits.

3.2.5 Discussão

Nesta seção apresentamos as curvas de probabilidade de erro de bit referentes a variação do número de iterações, do comprimento da seqüência de informação e do número de estados na treliça do codificador *RSC*. Quanto maior for o valor destes parâmetros melhor é o desempenho do sistema turbo. Todavia, é importante notar que:

- Existe um determinado valor para o número de iterações tal que, se aumentarmos este valor, aumentamos em conseqüência disto o tempo gasto na decodificação, mas não obtemos um ganho de codificação adicional significativo para o esquema turbo.

A Figura 3.7 mostra que as curvas referentes a 8ª, 7ª, 6ª, 5ª, 4ª, 3ª, 2ª e 1ª iteração alcançam a probabilidade de erro de bit de 10^{-5} para E_b/N_0 igual a 1.19, 1.20, 1.28, 1.38, 1.49, 1.79, 2.40 e 4.20 dB, com tempos de decodificação de 21.71, 19.38, 17.00, 14.70, 12.33, 9.95, 7.61 e 5.25 segundos, respectivamente. Note que estas curvas foram obtidas para o código ternário com 9 estados da seção 3.2.2, entretanto, a afirmação sobre o crescimento do número de iterações vale para qualquer sistema turbo *M-ário*.

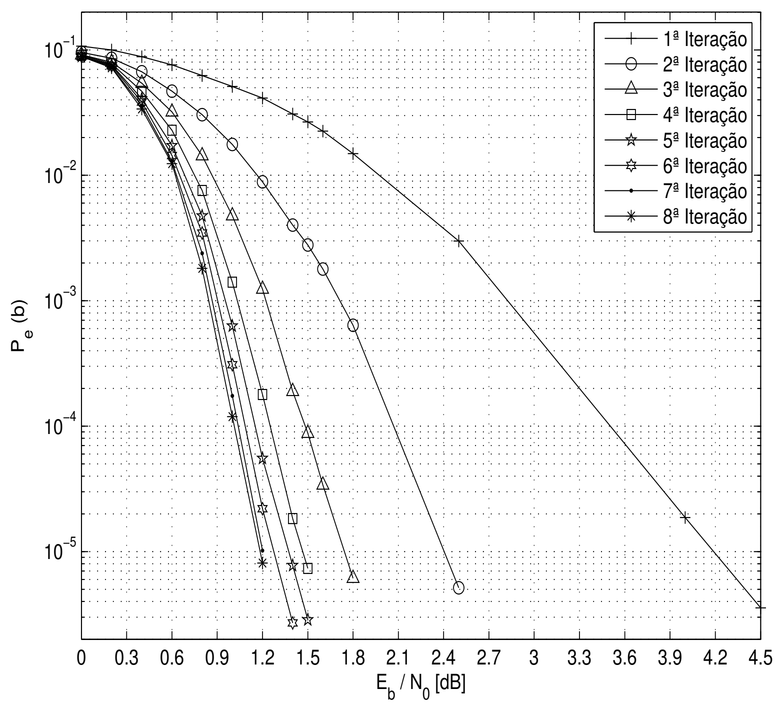


Figura 3.7: Processo iterativo turbo ternário.

- Existe um determinado valor para o comprimento da seqüência de informação (este valor depende do codificador *RSC*, do número de iterações, da taxa de codificação turbo e da cardinalidade do alfabeto), que, a partir deste valor, se aumentarmos o comprimento da seqüência de informação, aumentamos em conseqüência disto o tempo gasto na decodificação, mas o ganho de codificação obtido com esse aumento é muito pequeno.

A Figura 3.8 mostra que as curvas na 8ª iteração referentes aos comprimentos das seqüências de informação de $N = 8000$, $N = 4000$, $N = 2000$, $N = 1000$ e $N = 500$ bits, alcançam a probabilidade de erro de bit de 10^{-5} para E_b/N_0 igual a 0.8, 0.99, 1.19, 1.54 e 2.00 dB, com tempos de decodificação de 141.70, 50.93, 21.91, 10.02 e 4.89 segundos, respectivamente.

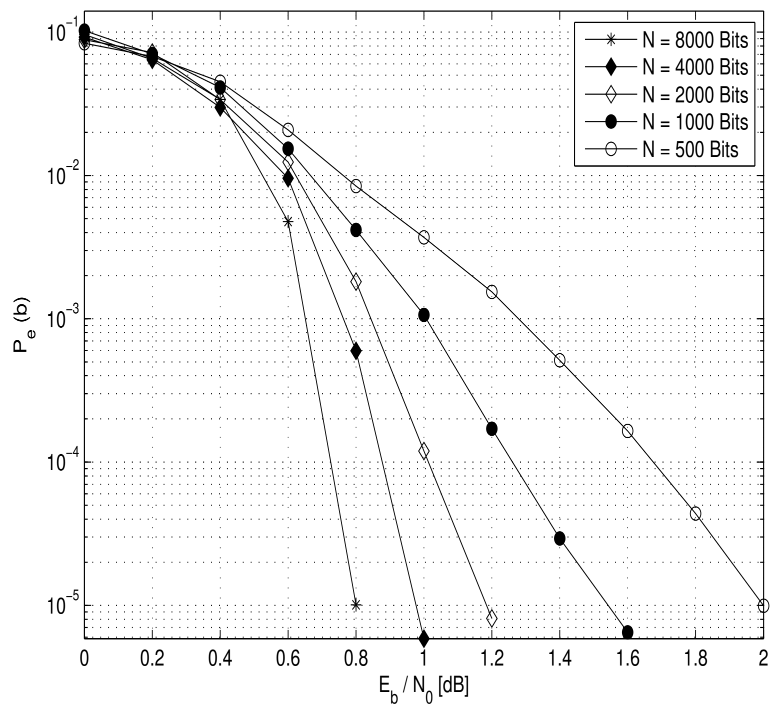


Figura 3.8: Variação do comprimento da seqüência de informação do sistema ternário.

- Existe um determinado valor para o número de estados da treliça do codificador *RSC*, que, a partir deste valor, o tempo gasto na decodificação é grande tornando assim o algoritmo de decodificação iterativo impraticável. O número de estados da treliça do codificador *RSC M-ário* é da ordem de M^l, onde l é o número de memórias do codificador *RSC* e M é a cardinalidade do alfabeto usado na modulação *M-PSK*.

  A Figura 3.9 mostra que as curvas na 8ª iteração do esquema binário referentes a 4, 8 e 16 estados, alcançam a probabilidade de erro de bit de 2×10^{-5} para E_b/N_0 igual a 1.3, 1.6 e 1.9 dB, com tempos de decodificação de 69.35, 37.56 e 21.00 segundos, respectivamente, e que as curvas da 8ª iteração do esquema ternário referentes a 9 e 27 estados, alcançam a probabilidade de erro de bit de 6×10^{-6} nas taxas de 1.4 e 1.11 dB, com tempos de decodificação de 23.21 e 55.02 segundos, respectivamente.

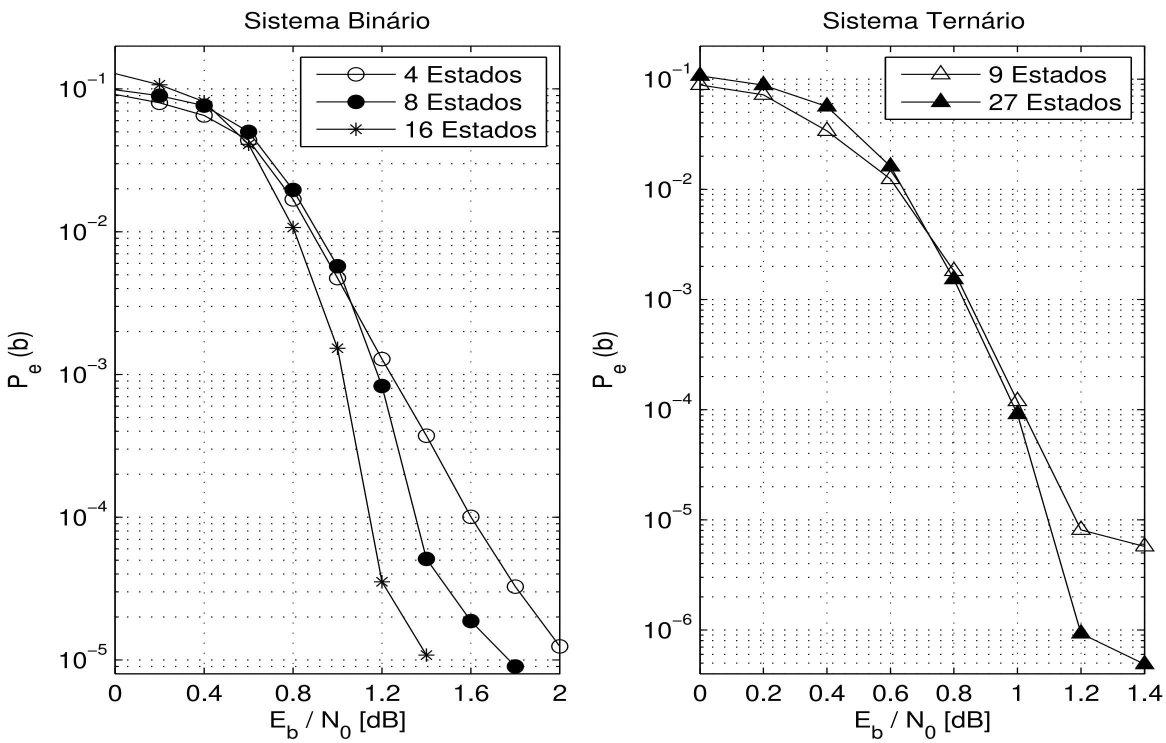


Figura 3.9: Variação do número de estados.

Neste capítulo, implementamos o algoritmo de codificação e decodificação, traçamos as curvas de probabilidade de erro de bit média referentes aos esquemas turbo *M-ários*e verificamos que quanto maior for:

- o número de iterações, Figura 3.7;
- o comprimento da seqüência de informação, Figura 3.8;
- o número de estados da treliça referente ao codificador *RSC*, Figura 3.9;

melhor o desempenho, em termos de probabilidade de erro de bit média versus E_b/N_0 da curva referente ao esquema turbo *M-ário*.

Portanto, entre a faixa de valores que existe para esses parâmetros, devemos sempre escolher valores que proporcionem baixa taxa de erro de *bit* (ou *símbolo*), baixa relação sinal ruído, baixa complexidade no algoritmo de decodificação e transmissão do máximo número de símbolos de informação. Mais geralmente, para cada sistema turbo *M-ário* analisamos os parâmetros (tamanho do alfabeto, comprimento da seqüência de informação, número de iterações, número de estados no codificador componente e taxa de codificação turbo) para os quais o sistema apresenta melhor desempenho ou converge mais rápido para o limitante de Shannon.

O desempenho do sistema turbo com relação à variação da taxa de codificação turbo será mostrado nos próximos capítulos.

3.3 Conclusão

Neste capítulo, apresentamos as curvas da taxa de erro de bit versus E_b/N_0 referente às simulações realizadas para os códigos turbo multiníveis definidos sobre os anéis e campos de números inteiros módulo-M.

No sistema *M-ário* definido sobre o anel de inteiros $\mathbb{Z}_M$, existe um número reduzido de polinômios para a matriz geradora do codificador *RSC*, pois nem todo elemento do anel $\mathbb{Z}_M$ possui inverso multiplicativo. Por outro lado, alguns anéis possuem um perfeito mapeamento de fonte quando comparado com o sistema binário, como os anéis $\mathbb{Z}_4$, $\mathbb{Z}_8$, $\mathbb{Z}_{16}$ e etc.

No sistema *M-ário* definido sobre o campo $\mathbb{Z}_M$, existe um número bem maior de polinômios para a matriz geradora do codificador *RSC*, mas não existe um perfeito mapeamento da fonte binária nos símbolos de $\mathbb{Z}_M$.

As curvas simuladas mostram que os códigos turbo multiníveis apresentam bom desempenho com baixa relação sinal ruído. Além disso, quanto maior o tamanho do alfabeto maior será a eficiência espectral do sistema turbo *M-ário*, pois 1 símbolo transmitido corresponde a log_2M bits transmitidos no sistema *M-ário*. Para cada sistema turbo *M-ário* existe uma faixa de valores para os parâmetros (tamanho do alfabeto, comprimento da seqüência de informação, número de iterações, taxa de codificação e codificador RSC) para os quais o sistema turbo apresenta melhor desempenho. Note que o esquema ternário com 9 estados apresenta melhor desempenho e menor complexidade de decodificação do que os outros esquemas apresentados.

Capítulo 4

Códigos Turbo *L-M-ário* com Modulação *Q-L-M-PSK*

4.1 Introdução

Nos códigos turbo multiníveis descritos na primeira parte desta tese, analisamos a complexidade de decodificação e a capacidade de correção de erro em termos da cardinalidade do alfabeto e da largura de banda, para sistemas turbo definidos sobre os anéis e campos de números inteiros $\mathbb{Z}_M$. Estes códigos multiníveis possuem a mesma taxa de codificação e diferentes números de estados em cada sistema comparado. Além disso, só existe perfeito mapeamento de bits para símbolos quando os códigos estão definidos sobre um anel $\mathbb{Z}_M$, cuja cardinalidade é uma potência de 2 ($\mathbb{Z}_4$, $\mathbb{Z}_8$, $\mathbb{Z}_{16}$, $\cdots$).

Neste capítulo, propomos um sistema turbo com codificação, modulação e decodificação híbrida multinível, que proporciona um perfeito mapeamento de bits para símbolos, além de manter-se a mesma taxa de codificação e o mesmo número de estados em cada sistema a ser comparado.

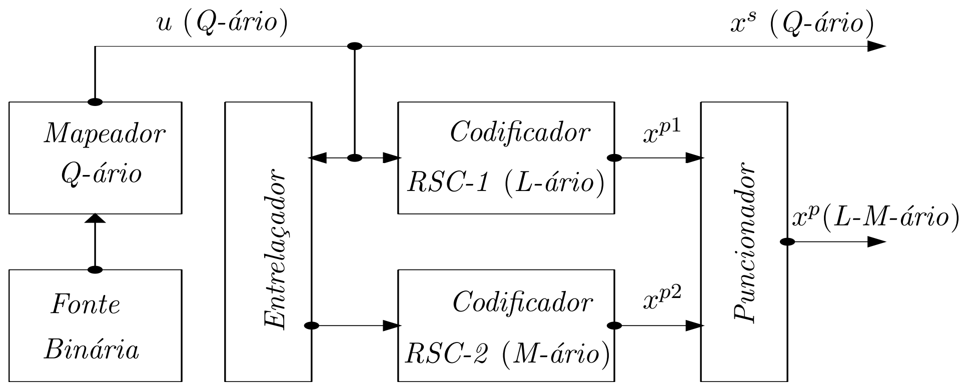
O esquema híbrido de codificação turbo multinível proposto processa seqüências de informações pertencentes ao anel de inteiros $\mathbb{Z}_Q$ e produz seqüências código com símbolos de paridade pertencentes aos campos de números inteiros $\mathbb{Z}_L$ e $\mathbb{Z}_M$. O esquema de modulação é o *Q-L-M-PSK* (*Q-L-M- Phase Shift Keying*), $Q \leq L \leq M$, que alterna no tempo as modulações *Q-PSK*, *L-PSK* e *M-PSK* na transmissão dos símbolos gerados pelo codificador. Este esquema híbrido de modulação portanto, se adapta melhor às operações definidas sobre os símbolos de informação em $\mathbb{Z}_Q$ e os símbolos de paridade definidos sobre $\mathbb{Z}_L$ e $\mathbb{Z}_M$. O canal considerado é o *AWGN* (*Aditive White Gaussian Noise*).

O esquema de decodificação é composto por um algoritmo de decodificação de máximo *a posteriori* (*MAP*), símbolo a símbolo, que utiliza os conceitos de código de treliça [2] [4]. O esquema de decodificação iterativo utiliza o algoritmo *MAP*, em cada decodificador componente, para obter a informação extrínseca referente aos símbolos de paridade definidos sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$, no estágio de decodificação anterior, e usá-la como informação *a priori* referente aos símbolos de informação definidos sobre $\mathbb{Z}_Q$, no próximo estágio de decodificação. Este processo geralmente produz uma diminuição na probabilidade de erro de símbolo (ou *bit*) a cada iteração.

4.2 Codificação Turbo Híbrida Multinível

O esquema de codificação turbo *L-M-ário* é constituído por um codificador *L-ário* de taxa 1/2, concatenado via um entrelaçador de N símbolos com outro codificador *M-ário* de taxa 1/2, juntamente com um mecanismo opcional de puncionamento. Esses dois codificadores constituintes são codificadores convolucionais sistemáticos recursivos (*RSC-Recursive Systematic Convolutional)* e concatenados em paralelo de acordo com o esquema da Figura 4.1.

Observe que a fonte de informação binária é mapeada em símbolos do anel de inteiros $\mathbb{Z}_Q$. Estes símbolos *Q-ários*, são inseridos na entrada de um codificador turbo *L-M-ário*, que produz em sua saída símbolos de informação pertencente ao alfabeto *Q-ário*, símbolos de paridade do codificador *RSC-1* pertencentes ao alfabeto *L-ário* e símbolos de paridade do codificador *RSC-2* pertencentes ao alfabeto *M-ário*, com $Q \leq L \leq M$.



$u = x^s$	Seqüência de símbolos de informação (*Q-ário*)
x^{p1}	Seqüência de símbolos de paridade do codificador $RSC-1$ (*L-ário*)
x^{p2}	Seqüência de símbolos de paridade do codificador $RSC-2$ (*M-ário*)
x^p	Seqüência de símbolos do puncionador (*L-M-ário*)

Figura 4.1: Esquema de codificação turbo *L-M-ário* com símbolos de entrada do alfabeto *Q-ário*.

Os símbolos dos alfabetos *L-ário* e *M-ário* pertencem aos respectivos campos de números inteiros $\mathbb{Z}_L$ e $\mathbb{Z}_M$ e os símbolos do alfabeto *Q-ário* pertencem ao anel de inteiros $\mathbb{Z}_Q$. Note que sempre existe um perfeito mapeamento dos símbolos de informação pertencentes ao alfabeto *Q-ário* nos bits da fonte de informação binária, pois a cardinalidade de $\mathbb{Z}_Q$ é uma potência de 2 ($\mathbb{Z}_4$, $\mathbb{Z}_8$, $\cdots$), ou seja, $Q = 2^q$, onde q é o comprimento da seqüência de bits que entra no mapeador para formar um símbolo *Q-ário*.

4.2.1 Codificador Convolucional Sistemático Recursivo

O esquema turbo da Figura 4.1 possui dois codificadores RSC definidos sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$ com entradas definidas sobre o anel $\mathbb{Z}_Q$. As definições apresentadas a seguir para os códigos RSC definidos sobre o campo $\mathbb{Z}_M$ são válidas para os códigos RSC definidos sobre o campo $\mathbb{Z}_L$.

Seja $\mathbb{Z}_M$ um campo de inteiros e seja $L_r(D)$ o campo formado por polinômios racionais do tipo $q(D)/p(D)$, onde $q(D)$ e $p(D)$ são polinômios com coeficientes em $\mathbb{Z}_M$, expressos pela série de *Laurent* [20] [21] [25],

$$f(D) = \sum_{i=0}^{n} f_i D^i,$$

com $f_i \in \mathbb{Z}_M$ - chamado campo das funções realizáveis sobre $\mathbb{Z}_M$.

Assim, um codificador convolucional sistemático recursivo, com taxa de codificação k/n sobre o campo de funções realizáveis $L_r(D)$, é um mapeamento linear expresso como

$$v(D) = u(D)G_r(D),$$

onde $u(D)$ é o polinômio correspondente a seqüência de informação com coeficientes em $\mathbb{Z}_Q$ e $G_r(D)$ é uma matriz $k{\times}n$ na forma sistemática

$$G_r(D) = \begin{bmatrix} I_k & X(D) \end{bmatrix}.$$

Observe que I_k é uma matriz identidade $k{\times}k$ e $X(D)$ é uma matriz $k{\times}$*(n-k)* com elementos cuja representação é $q(D)/p(D)$ em $L_r(D)$ e os coeficientes de $q(D)$ e $p(D)$ pertencem a $\mathbb{Z}_M$.

Portanto, o conjunto

$$C = \{u(D)g(D) \mid u(D) \in L_r(D)^k\},$$

é um código *RSC* sobre $\mathbb{Z}_M$, com taxa de codificação k/n, onde $g(D) = G_r(D)$ é a matriz geradora com elementos em $L_r(D)$. A Figura 4.2 mostra o diagrama de um codificador *RSC*, definido sobre o campo $\mathbb{Z}_5$, cuja matriz geradora é:

$$g(D) = \left[1 \quad \frac{4 + 2D + 3D^2}{1 + 4D + 4D^2}\right],$$

onde os números da Figura 4.2 correspondem aos coeficientes dos polinômios da matriz $g(D)$.

Exemplo 4.2.1 *O codificador RSC 5-ário da Figura 4.2 possui* $5^2 = 25$ *estados na treliça e utiliza apenas 4 das 5 possíveis transições associadas a cada estado. A vantagem desta codificação é o perfeito mapeamento dos bits para símbolos do alfabeto quaternário, a redução* *da complexidade de decodificação em relação ao esquema 5-ário e o aumento das distâncias* D_{free} *e* D_{efet} *do código turbo.*

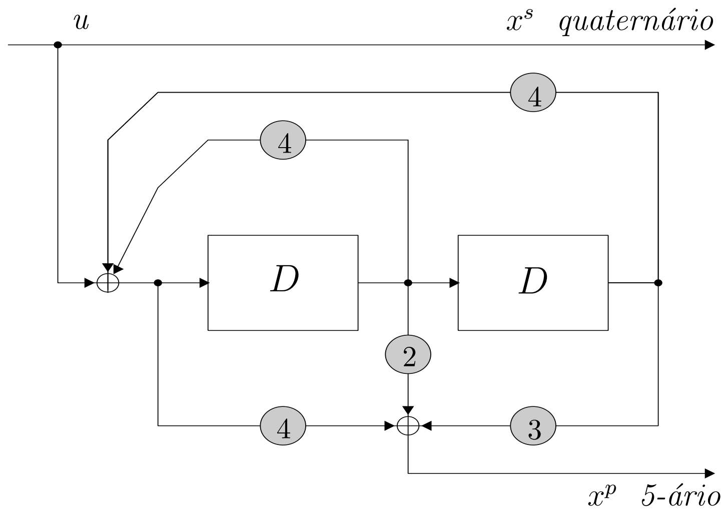


Figura 4.2: Diagrama de um codificador *RSC* definido sobre $\mathbb{Z}_5$ com entrada definida sobre $\mathbb{Z}_4$.

4.3 Taxa de Codificação Turbo

A taxa de codificação de um codificador que possui k entradas e n saídas é $r = \dfrac{k}{n}$ se a cardinalidade do alfabeto de entrada for a mesma do alfabeto de saída. No codificador proposto neste capítulo a taxa de codificação turbo relaciona a cardinalidade do alfabeto de entrada com as diferentes cardinalidades dos alfabetos de saída.

4.3.1 Taxa de Codificação de Códigos Multiníveis

O codificador multinível mostrado na Figura 4.3 possui k entradas com símbolos do alfabeto *Q-ário* e n saídas com símbolos do alfabeto *M-ário*. A taxa de codificação é dada pela expressão:

$$r = \frac{\log Q^k}{\log M^n} = \frac{k \log Q}{n \log M}.$$

Se $Q = M$, então $r = \dfrac{k}{n}$.

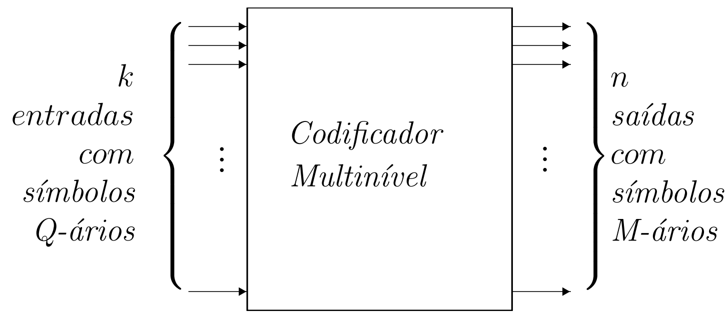


Figura 4.3: Diagrama do codificador multinível.

4.3.2 Codificador com Múltiplos Alfabetos de Saída

O codificador multinível mostrado na Figura 4.4 possui k entradas com símbolos pertencentes ao alfabeto *Q-ário* e $n = n_1 + n_2 + \cdots + n_l$ saídas com símbolos pertencentes aos respectivos alfabetos M_1-*ário*, M_2-*ário*, $\cdots$, M_l-*ário*.

A taxa de codificação é dada pela expressão:

$$r = \frac{\log Q^k}{\log(M_1^{n_1} \cdot M_2^{n_2} \cdots M_l^{n_l})} = \frac{k \log Q}{n_1 \log M_1 + n_2 \log M_2 + \cdots + n_l \log M_l}.$$

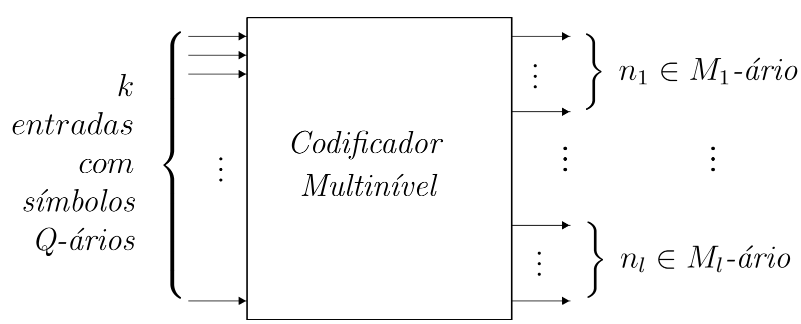


Figura 4.4: Diagrama do codificador multinível com múltiplos alfabetos de saída.

No esquema turbo *L-M-ário* com entrada *Q-ária* da Figura 4.1, o primeiro codificador componente é definido sobre o campo $\mathbb{Z}_L$ e o segundo sobre o campo $\mathbb{Z}_M$. Logo, com puncionamento, a taxa de codificação turbo é:

$$R_1 = \frac{\log Q}{\log(Q \cdot L^{1/2} \cdot M^{1/2})} = \frac{\log Q}{\log Q + \dfrac{\log L + \log M}{2}},$$

e sem puncionamento, a taxa de codificação turbo é:

$$R_2 = \frac{\log Q}{\log(Q \cdot L \cdot M)} = \frac{\log Q}{\log Q + \log L + \log M}.$$

Exemplo 4.3.1 *A Figura 4.5 apresenta o codificador turbo 5-7-ário. Este codificador é constituído pelo codificador RSC-1 com entrada quaternária e saída quaternária e 5-ária e pelo codificador RSC-2 com entrada quaternária e saída quaternária e 7-ária.*

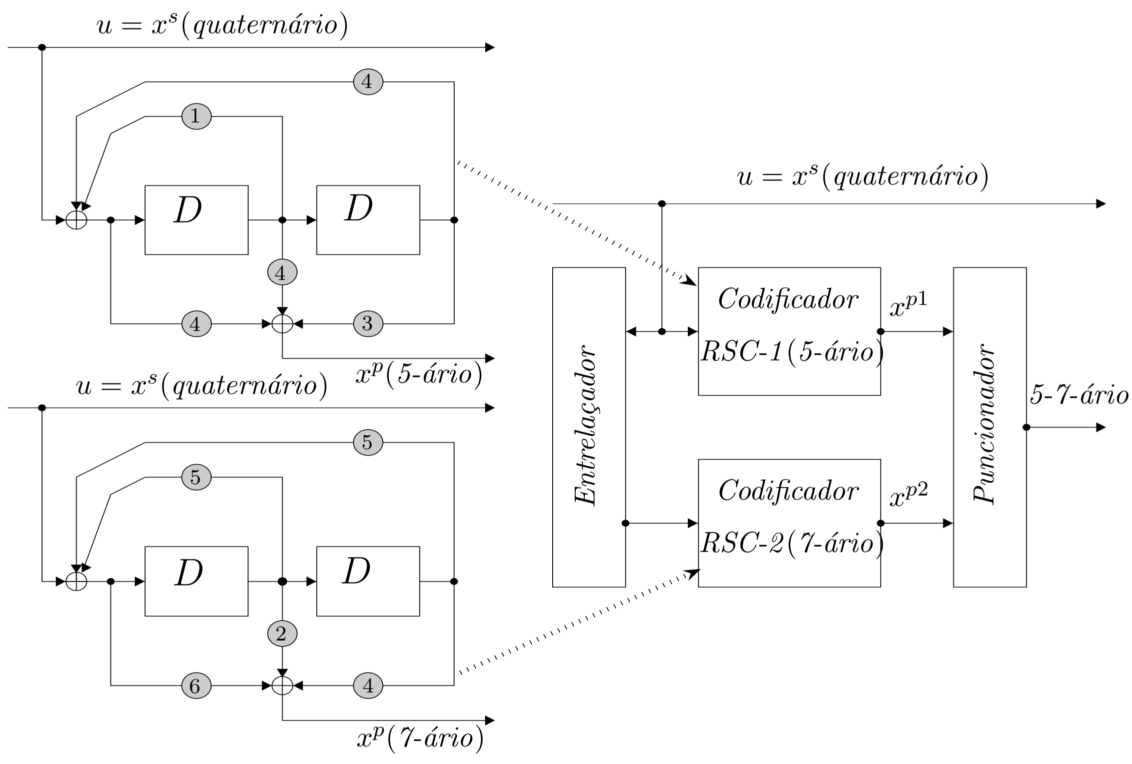


Figura 4.5: Diagrama do codificador turbo com codificadores componentes definidos sobre $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e símbolos de informação definidos sobre $\mathbb{Z}_4$.

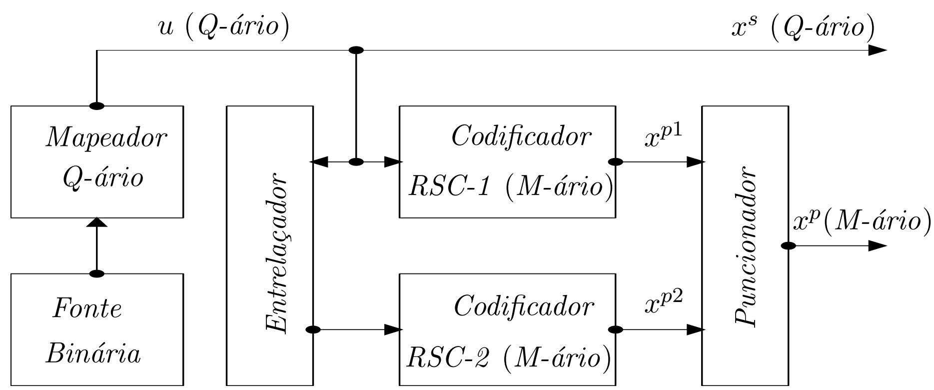
Como o primeiro codificador componente é definido sobre o campo $\mathbb{Z}_5$ e o segundo sobre o campo $\mathbb{Z}_7$, com puncionamento, a taxa de codificação turbo é dada pela expressão:

$$R_1 = \frac{\log 4}{\log(4 \cdot 5^{1/2} \cdot 7^{1/2})} = \frac{\log 4}{\log 4 + \dfrac{\log 5 + \log 7}{2}} = 0.4382,$$

e sem puncionamento, a taxa de codificação turbo é dada pela expressão:

$$R_2 = \frac{\log 4}{\log(4 \cdot 5 \cdot 7)} = \frac{\log 4}{\log 4 + \log 5 + \log 7} = 0.2805.$$

Fazendo $L = M$, temos o codificador turbo multinível *M-ário* com entrada *Q-ária* da Figura 4.6, cujos codificadores componentes são definidos sobre o campo $\mathbb{Z}_M$. Este codificador processa símbolos de informações pertencentes ao anel de inteiros $\mathbb{Z}_Q$ cuja cardinalidade é uma potência de 2 e produz símbolos de paridade pertencentes ao campo $\mathbb{Z}_M$.



$u = x^s$	Seqüência de símbolos de informação (*Q-ário*)
x^{p1}	Seqüência de símbolos de paridade do codificador $RSC-1$ (*M-ário*)
x^{p2}	Seqüência de símbolos de paridade do codificador $RSC-2$ (*M-ário*)
x^p	Seqüência de símbolos do puncionador (*M-ário*)

Figura 4.6: Esquema de codificação turbo *M-ário* com símbolos de entrada do alfabeto *Q-ário*.

Como o puncionador apaga os símbolos ímpares da seqüência de paridade do primeiro codificador e os pares do segundo codificador componente, com puncionamento, a taxa de codificação turbo é dada pela expressão:

$$R_3 = \frac{\log Q}{\log(Q \cdot M)} = \frac{\log Q}{\log Q + \log M},$$

e sem puncionamento, a taxa de codificação turbo é:

$$R_4 = \frac{\log Q}{\log(Q \cdot M^2)} = \frac{\log Q}{\log Q + 2\log M}.$$

Exemplo 4.3.2 *A Figura 4.7 apresenta o codificador turbo 5-ário com entrada quaternária e saída quaternária e 5-ária e o codificador turbo 7-ário com entrada quaternária e saída quaternária e 7-ária.*

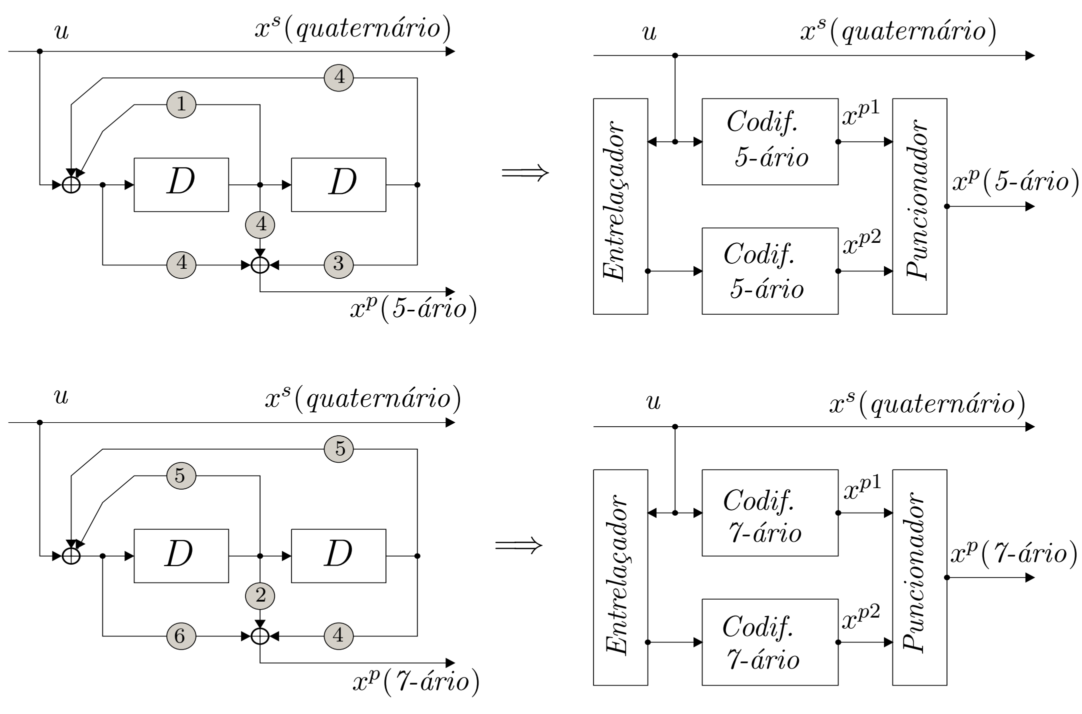


Figura 4.7: Diagrama dos codificadores *RSC* definidos sobre $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e dos respectivos codificadores turbo com entradas quaternárias.

Quando os dois codificadores componentes do código turbo são definidos sobre o campo $\mathbb{Z}_5$, *com puncionamento, a taxa de codificação turbo é:*

$$R_3 = \frac{\log 4}{\log(4 \cdot 5)} = \frac{\log 4}{\log 4 + \log 5} = 0.4628,$$

e sem puncionamento, a taxa de codificação turbo é:

$$R_4 = \frac{\log 4}{\log(4 \cdot 5^2)} = \frac{\log 4}{\log 4 + 2\log 5} = 0.3010.$$

Além disso, quando os dois codificadores componentes do código turbo são definidos sobre o campo $\mathbb{Z}_7$, com puncionamento, a taxa de codificação turbo é:

$$R_3 = \frac{\log 4}{\log(4 \cdot 7)} = \frac{\log 4}{\log 4 + \log 7} = 0.4160,$$

e sem puncionamento, a taxa de codificação turbo é:

$$R_4 = \frac{\log 4}{\log(4 \cdot 7^2)} = \frac{\log 4}{\log 4 + 2\log 7} = 0.2626.$$

As tabelas seguintes mostram as taxas de codificação e os estados dos esquemas turbo híbridos com entrada definida sobre $\mathbb{Z}_Q$ e codificadores componentes definidos sobre $\mathbb{Z}_L$ e $\mathbb{Z}_M$, onde $2 \leq Q \leq L \leq M \leq 7$. As taxas r_2, r_3, r_4, r_5, r_6 e r_7 são referentes aos codificadores turbo *L-M-ários* com entradas definidas sobre $\mathbb{Z}_2$, $\mathbb{Z}_3$, $\mathbb{Z}_4$, $\mathbb{Z}_5$, $\mathbb{Z}_6$ e $\mathbb{Z}_7$, respectivamente.

A Tabela 4.1 apresenta os estados dos codificadores turbo com codificador *RSC-1* definido sobre o campo $\mathbb{Z}_2$ e codificador *RSC-2* definido sobre os campos $\mathbb{Z}_2$, $\mathbb{Z}_3$, $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e os anéis $\mathbb{Z}_4$ e $\mathbb{Z}_6$. A taxa r_2 é referente ao codificador turbo com entrada binária.

Estados	$\mathbb{Z}_2$					$\mathbb{Z}_3$			$\mathbb{Z}_4$		$\mathbb{Z}_5$		$\mathbb{Z}_6$		$\mathbb{Z}_7$	
	+	2	4	8	16	3	9	27	4	16	5	25	6	36	7	49
	2	4	6	10	18	5	11	29	6	18	7	27	8	38	9	51
$\mathbb{Z}_2$	4	6	8	12	20	7	13	31	8	20	9	29	10	40	11	53
	8	10	12	16	24	11	17	35	12	24	13	33	14	44	15	57
	16	18	20	24	32	19	25	43	20	32	21	41	22	52	23	65
Taxa	$r_2 = 0.50$					$r_2 = 0.43$			$r_2 = 0.4$		$r_2 = 0.37$		$r_2 = 0.35$		$r_2 = 0.34$	

Tabela 4.1: Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entrada binária e $2 = L \leq M \leq 7$.

A Tabela 4.2 apresenta os estados dos codificadores turbo com codificador *RSC-1* definido sobre o campo $\mathbb{Z}_3$ e codificador *RSC-2* definido sobre os campos $\mathbb{Z}_3$, $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e os anéis $\mathbb{Z}_4$ e $\mathbb{Z}_6$. A taxa r_3 é referente ao codificador turbo com entrada ternária.

Estados	$\mathbb{Z}_3$				$\mathbb{Z}_4$		$\mathbb{Z}_5$		$\mathbb{Z}_6$		$\mathbb{Z}_7$	
	+	3	9	27	4	16	5	25	6	36	7	49
	3	6	12	30	7	19	8	28	9	39	10	56
$\mathbb{Z}_3$	9	12	18	36	13	25	14	34	15	45	16	58
	27	30	36	54	31	43	32	52	33	63	34	76
Taxa	$r_2 = 0.38$ $r_3 = 0.50$				$r_2 = 0.35$ $r_3 = 0.46$		$r_2 = 0.33$ $r_3 = 0.44$		$r_2 = 0.32$ $r_3 = 0.43$		$r_2 = 0.31$ $r_3 = 0.41$	

Tabela 4.2: Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária e ternária e $3 = L \leq M \leq 7$.

A Tabela 4.3 apresenta os estados dos codificadores turbo com codificador *RSC-1* definido sobre o anel $\mathbb{Z}_4$ e codificador *RSC-2* definido sobre os campos $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e os anéis $\mathbb{Z}_4$ e $\mathbb{Z}_6$. A taxa r_4 é referente ao codificador turbo com entrada quaternária.

Estados	$\mathbb{Z}_4$			$\mathbb{Z}_5$		$\mathbb{Z}_6$		$\mathbb{Z}_7$	
	+	4	16	5	25	6	36	7	49
$\mathbb{Z}_4$	4	8	20	9	29	10	40	11	51
	16	20	32	21	41	22	52	23	65
Taxa	$r_2 = 0.33$ $r_3 = 0.44$ $r_4 = 0.50$			$r_2 = 0.31$ $r_3 = 0.42$ $r_4 = 0.48$		$r_2 = 0.30$ $r_3 = 0.40$ $r_4 = 0.46$		$r_2 = 0.29$ $r_3 = 0.39$ $r_4 = 0.45$	

Tabela 4.3: Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária e quaternária e $4 = L \leq M \leq 7$.

A Tabela 4.4 apresenta os estados dos codificadores turbo com codificador *RSC-1* definido sobre o campo $\mathbb{Z}_5$ e codificador *RSC-2* definido sobre os campos $\mathbb{Z}_5$ e $\mathbb{Z}_7$ e o anel $\mathbb{Z}_6$. A taxa r_5 é referente ao codificador turbo com entrada *5-ária*.

Estados	$\mathbb{Z}_5$			$\mathbb{Z}_6$		$\mathbb{Z}_7$	
$\mathbb{Z}_5$	+	5	25	6	36	7	49
	5	10	30	11	41	12	54
	25	30	50	31	61	32	74
Taxa	$r_2 = 0.30$			$r_2 = 0.28$		$r_2 = 0.28$	
	$r_3 = 0.40$			$r_3 = 0.39$		$r_3 = 0.38$	
	$r_4 = 0.46$			$r_4 = 0.44$		$r_4 = 0.43$	
	$r_5 = 0.50$			$r_5 = 0.48$		$r_5 = 0.47$	

Tabela 4.4: Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária, quaternária e *5-ária* e $5 = L \leq M \leq 7$.

A Tabela 4.5 apresenta os estados dos codificadores turbo com codificador *RSC-1* definido sobre o anel $\mathbb{Z}_6$ e codificador *RSC-2* definido sobre o anel $\mathbb{Z}_6$ e o campo $\mathbb{Z}_7$. A taxa r_6 é referente ao codificador turbo com entrada *6-ária*.

Estados	$\mathbb{Z}_6$			$\mathbb{Z}_7$	
$\mathbb{Z}_6$	+	6	36	7	49
	6	12	42	13	55
	36	42	72	43	85
Taxa	$r_2 = 0.27$			$r_2 = 0.27$	
	$r_3 = 0.38$			$r_3 = 0.37$	
	$r_4 = 0.43$			$r_4 = 0.42$	
	$r_5 = 0.47$			$r_5 = 0.46$	
	$r_6 = 0.50$			$r_6 = 0.48$	

Tabela 4.5: Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária, quaternária, *5-ária* e *6-ária* e $6 = L \leq M \leq 7$.

A Tabela 4.6 apresenta os estados dos codificadores turbo com codificador *RSC-1* definido sobre o campo $\mathbb{Z}_7$ e codificador *RSC-2* definido sobre o campo $\mathbb{Z}_7$. A taxa r_7 é referente ao codificador turbo com entrada *7-ária*.

Estados	$\mathbb{Z}_7$		
$\mathbb{Z}_7$	+	7	49
	7	14	56
	49	56	98
Taxa	$r_2 = 0.26$		
	$r_3 = 0.36$		
	$r_4 = 0.41$		
	$r_5 = 0.45$		
	$r_6 = 0.47$		
	$r_7 = 0.50$		

Tabela 4.6: Número de estados e taxas de codificação dos codificadores turbo *L-M-ários* com entradas binária, ternária, quaternária, *5-ária*, *6-ária* e *7-ária* e $L = M = 7$.

Para garantir o perfeito mapeamento de bits para símbolos *Q-ários*, as taxas r_3, r_5, r_6 e r_7 não serão consideradas nas simulações deste capítulo. Entretanto, estas tabelas descrevem todas as possibilidades de comparações dos sistemas turbo *L-M-ários* referentes a taxa de codificação e ao número de estados em cada codificador turbo. Note que fazendo $Q = L = M$ retornamos aos sistemas turbo descritos no Capítulo 2.

Observe que também podemos obter taxas diferentes das convencionais modificando-se a matriz de apagamento do puncionador, ou seja, não só apagando-se os símbolos pares de um codificador e os ímpares do outro. Todavia, deve ser estabelecido critério na escolha desta matriz de apagamento, pois excluir símbolos que ocupam a mesma posição nas seqüências de paridade dos codificador *RSC-1* e *RSC-2* pode causar sérios problemas na decodificação.

4.3.3 Modulador *Q-L-M-PSK*

A Figura 4.8 mostra o diagrama do esquema turbo *L-M-ário* com modulação do tipo *Q-L-M-PSK*. Neste esquema, a fonte de informação gera seqüências de símbolos pertencentes ao anel de inteiros $\mathbb{Z}_Q$. Estes símbolos são codificados através de um esquema de codificação turbo *L-M-ário* que produz palavras código em sua saída, com símbolos de informação pertencentes ao anel $\mathbb{Z}_Q$, e com símbolos de paridade pertencentes aos campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$.

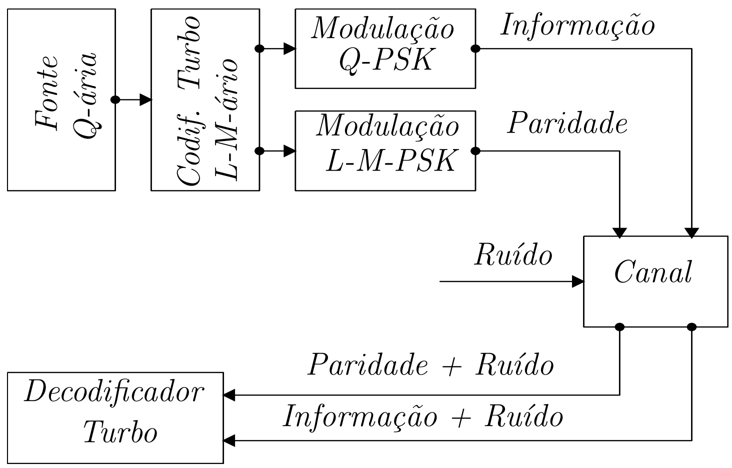


Figura 4.8: Diagrama do sistema turbo com modulação *Q-L-M-PSK*

Portanto, os símbolos de informação do anel $\mathbb{Z}_Q$, $\theta_1 = \{0, 1, \cdots, Q-1\}$, são transmitidos usando a modulação *Q-PSK*, com sinais do tipo $S_{\theta_1} = \exp\left(j\dfrac{2\pi\theta_1}{Q}\right)$; os símbolos de paridade do campo $\mathbb{Z}_L$, $\theta_2 = \{0, 1, \cdots, L-1\}$, são transmitidos usando a modulação *L-PSK* com sinais do tipo $\bar{S}_{\theta_2} = \exp\left(j\dfrac{2\pi\theta_2}{L}\right)$; e os símbolos de paridade do campo $\mathbb{Z}_M$, $\theta_3 = \{0, 1, \cdots, M-1\}$, são transmitidos usando a modulação *M-PSK* com sinais do tipo $\dot{S}_{\theta_3} = \exp\left(j\dfrac{2\pi\theta_3}{M}\right)$, de acordo com a Figura 4.9.

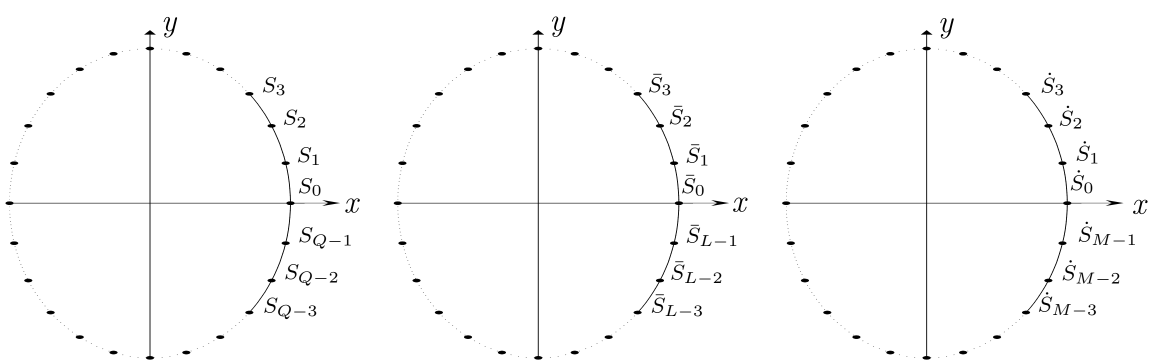


Figura 4.9: Constelações *Q-PSK*, *L-PSK* e *M-PSK*

No esquema turbo proposto, os símbolos de informação *Q-ários*, sem codificação, devem ser transmitidos com a modulação *Q-PSK*, pois a transmissão destes símbolos não codificados, com uma modulação *L-PSK*, onde $L > Q$, implica em redução nas distâncias livre e efetiva do código. Para os símbolos de paridade *L-M-ários* a modulação usada é a *L-M-PSK* e não há mapeamento de bits para símbolos, além disso, os símbolos de paridade definidos sobre campos cuja cardinalidade é maior do que Q, proporcionam maior confiabilidade no cálculo da informação extrínseca.

As características relacionadas com a fonte de informação, a codificação e a modulação no sistema turbo da Figura 4.8 são:

1. mapeamento perfeito de bits para símbolos do alfabeto *Q-ário*, pois a cardinalidade do anel $\mathbb{Z}_Q$ é uma potência de 2 ($\mathbb{Z}_4$, $\mathbb{Z}_8$, $\mathbb{Z}_{16}$, $\cdots$);

2. maior número de polinômios para a matriz geradora do codificador definido sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$;

3. maiores distâncias (D_{free} e D_{efet}) do que no esquema turbo multiníveis convencionais;

4. maior variação da taxa de codificação turbo;

5. menor complexidade de codificação que o esquema turbo convencional definido sobre o campo $\mathbb{Z}_M$;

6. maior possibilidade de manter-se o mesmo número de estados e a mesma taxa de codificação na comparação entre os sistemas.

4.4 Algoritmo de Decodificação

No Capítulo 2, apresentamos as expressões usadas no algoritmo de decodificação dos códigos turbo *M-ários* definidos sobre os anéis e campos de números inteiros $\mathbb{Z}_M$. Devido a similaridade na obtenção das expressões do algoritmo de decodificação turbo padrão com as expressões do algoritmo proposto neste capítulo, apresentamos a seguir apenas as principais expressões utilizadas para decodificar os códigos turbo definidos sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$, com fonte de informação definida sobre o anel $\mathbb{Z}_Q$.

O algoritmo de decodificação turbo de máximo *a posteriori* - *MAP*, fornece a informação *a posteriori*

$$L(u_k) = \ln(p(u_k = \theta \mid y)), \tag{4.1}$$

onde $y = (y_1 \ y_2 \ \cdots \ y_{k-1} \ y_k \ y_{k+1} \ \cdots \ y_N)$ é a seqüência de símbolos recebida, $u_k = \theta \in \{0, 1, ..., Q-1\}$ e $\theta \in \mathbb{Z}_Q$.

Se as transições entre o estado prévio, $S_{k-1} = s'$, e o estado presente, $S_k = s$, são mutuamente exclusivas (isto é, apenas uma delas pode ter ocorrido na treliça referente ao codificador *RSC*), então usando a regra de *Bayes*

$$L(u_k) = \ln\left(\frac{\sum\limits_{(s',s)} p(S_{k-1} = s', S_k = s, y)}{p(y)}\right), \tag{4.2}$$

onde (s', s) é o conjunto de transições do estado prévio, $S_{k-1} = s'$, para o estado presente, $S_k = s$, que pode ocorrer se o símbolo de entrada u_k for igual a θ.

Como o canal é sem memória, usando a regra de *Bayes* e substituindo $p\left(y\right)$ por $p\left(y\right)/p\left(y_k\right)$ na expressão (4.2) temos:

$$L(u_k) \ = \ \ln\left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s\right) \cdot \beta_k(s)\right), \tag{4.3}$$

onde $\alpha_k(s)$ e $\beta_k(s)$ são as probabilidades definidas, respectivamente, como:

$$\alpha_k(s) \ = \ \frac{\sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s\right)}{\sum\limits_{todo\ s}\sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k\left(s', s\right)}, \tag{4.4}$$

$$\beta_{k-1}(s) \;=\; \frac{\sum\limits_{todo\ s} \beta_k(s) \cdot \gamma_k(s', s)}{\sum\limits_{todo\ s} \sum\limits_{todo\ s'} \alpha_{k-1}(s') \cdot \gamma_k(s', s)}. \tag{4.5}$$

Observe que $\alpha_k(s)$ e $\beta_k(s)$ são calculadas recursivamente pelas equações (4.4) e (4.5) e possuem as condições iniciais.

$$\alpha_0(S_0 = s) = \beta_N(S_N = s) = \begin{cases} 1 \ \text{ para } & s = 0 \\ 0 \ \text{ para } & s \neq 0. \end{cases} \tag{4.6}$$

Com base nas mesmas hipóteses temos que a probabilidade de transição é dada por

$$\gamma_k(s', s) \;=\; p(y_k \mid x_k) \cdot p(u_k). \tag{4.7}$$

Assim, apresentamos as expressões de $\alpha_{k-1}(s')$, $\beta_k(s)$ e $\gamma_k(s', s)$, necessárias para calcular a informação *a posteriori* $L(u_k)$, na saída de cada decodificador componente.

Observe que cada decodificador componente calcula a informação *a posteriori* $L(u_k)$, utilizando o algoritmo de decodificação *MAP*. De acordo com a Figura 4.10, o primeiro decodificador componente fornece a informação *a posteriori* referente aos símbolos de paridade do campo $\mathbb{Z}_L$, e o segundo decodificador componente fornece a informação *a posteriori* referente aos símbolos de paridade do campo $\mathbb{Z}_M$.

4.4.1 Decodificação Iterativa Turbo *L-M-ária*

No processo de decodificação iterativa turbo definido sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$, esquematizado na Figura 4.10, calcula-se a informação extrínseca referente a paridade *L-ária*, na saída do primeiro decodificador. Esta informação extrínseca é usada como informação *a priori* referente a informação *Q-ária*, na entrada do segundo decodificador. Em seguida, a informação extrínseca referente a paridade *M-ária*, é calculada na saída do segundo decodificador e usada como informação *a priori* referente a informação *Q-ária*, na entrada do primeiro decodificador.

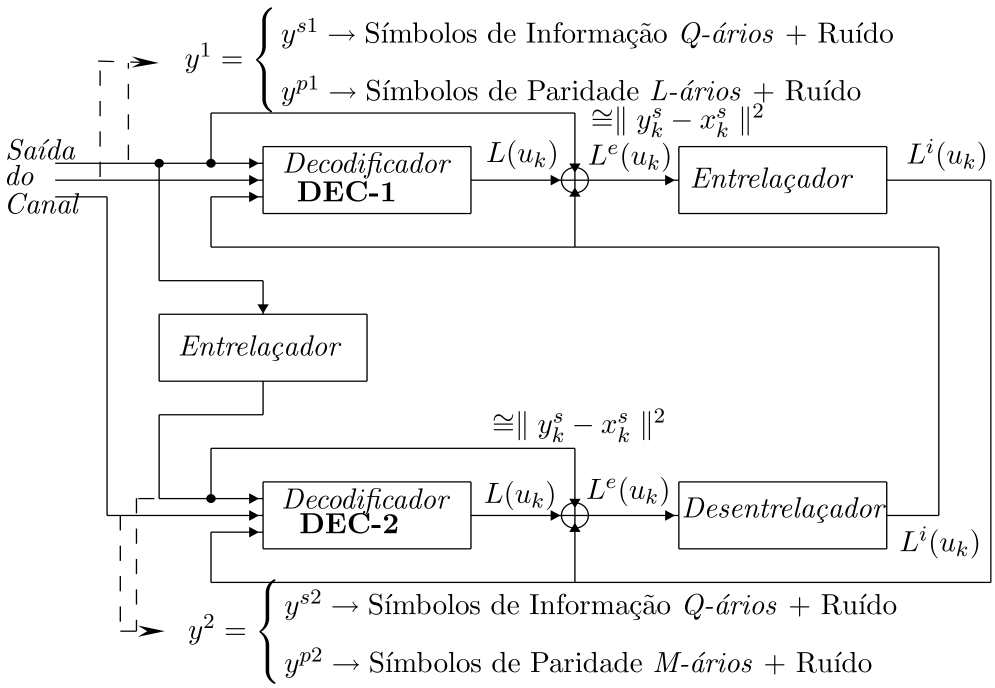


Figura 4.10: Decodificação iterativa turbo.

Sabendo que cada codificador RSC do esquema de codificação turbo tem taxa de codificação 1/2, a palavra código transmitida é dada por $x_k = x_k^s\ x_k^p = u_k\ x_k^p$, e a palavra código recebida é dada por $y_k = y_k^s\ y_k^p$.

Assim, se o canal é sem memória, gaussiano e com modulação *Q-L-M-PSK*, então, a função densidade de probabilidade é dada por

$$\begin{aligned} p(y_k \mid x_k) &= \frac{1}{\pi\sqrt{N_0^s N_0^p}} \cdot \exp\left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0^s}\right) \cdot \exp\left(\frac{-\parallel y_k^p - x_k^p \parallel^2}{N_0^p}\right) \\ &= \exp\left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0^s}\right) \cdot \gamma_k^e(s', s), \end{aligned}$$

onde: x_k^s é o símbolo sistemático da palavra código transmitida x_k; x_k^p é o símbolo de paridade da palavra código transmitida x_k; y_k^s é o sinal recebido correspondente ao símbolo transmitido

x_k^s e y_k^p é o sinal recebido correspondente ao símbolo transmitido x_k^p; N_0^s é a densidade espectral de potência referente a modulação *Q-PSK* e N_0^P é a densidade espectral de potência referente a modulação *L-PSK*(quando for utilizado o decodificador **DEC-1**) ou referente a modulação *M-PSK*(quando for utilizado o decodificador **DEC-2**).

Note que os símbolos sistemáticos pertencentes ao anel $\mathbb{Z}_Q$ são transmitidos através da modulação *Q-PSK* e os símbolos de paridade pertencentes aos campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$ são transmitidos através das respectivas modulações *L-PSK* e *M-PSK* (*L-M-PSK*).

Portanto, a informação *a posteriori*, $L(u_k)$, calculada através do algoritmo MAP na saída de cada decodificador componente, pode ser dividida em três termos

$$L(u_k) = L^i(u_k) + \exp\left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0^s}\right) + L^e(u_k, \tag{4.8}$$

onde $L^i(u_k) = \ln(p(u_k))$ é a informação intrínseca, $\exp\left(\frac{-\parallel y_k^s - x_k^s \parallel^2}{N_0^s}\right)$ é a informação sistemática associada a modulação *Q-PSK*, e

$$L^e(u_k) = \ln\left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \beta_k(s) \cdot \gamma_k^e(s,s')\right), \tag{4.9}$$

é a informação extrínseca associada a modulação *L-M-PSK*. Estes termos serão usados no processo de decodificação iterativo como a seguir.

Processo Iterativo: Considere inicialmente o primeiro decodificador componente na primeira iteração. Este decodificador recebe a seqüência de canal, y^1 (ver Figura 4.10), e produz uma estimativa da informação *a posteriori* $L_{11}(u_k)$ dos símbolos de dados u_k, onde $k \in \{1, \cdots, N\}$ e N é o comprimento da seqüência de informação. Note que nesta primeira iteração a informação *a priori*, que o primeiro decodificador componente recebe, é $\ln(p(u_k = \theta)) = \ln(1/Q)$.

O segundo decodificador componente recebe a seqüência de canal y^2 junto com a informação extrínseca entrelaçada do primeiro decodificador componente e fornece uma estimativa da informação *a posteriori* $L_{12}(u_k)$, dos símbolos de dado u_k.

A informação extrínseca calculada na saída do primeiro decodificador, após ser entrelaçada, é usada como informação *a priori* na entrada do segundo decodificador. E de forma análoga, a informação extrínseca calculada na saída do segundo decodificador, após ser desentrelaçada, é usada como informação *a priori* na entrada do primeiro decodificador e assim por diante.

Na segunda iteração, o primeiro decodificador componente novamente processa sua seqüência recebida de canal y^1, mas dessa vez, ele também possui a informação *a priori* $L^i(u_k)$. Esta informação *a priori* é fornecida pela porção extrínseca da informação *a posteriori* $L_{12}(u_k)$, calculada pelo segundo decodificador componente, na primeira iteração. Portanto, este decodificador pode produzir uma melhor informação *a posteriori* $L_{21}(u_k)$, na saída do primeiro decodificador componente, na segunda iteração.

Este processo iterativo continua e, a cada iteração, em média a taxa de erro de bit diminui para uma mesma relação sinal ruído - SNR (*signal-to-noise ratio*).

4.5 Conclusão

Este capítulo estabelece um critério para analisar o desempenho do sistema de comunicação digital para os códigos turbo híbridos multiníveis. O tratamento teórico referente à codificação turbo *L-M-ária* e a modulação *Q-L-M-PSK* é para avaliar os códigos que proporcionam maior eficiência no processo de decodificação turbo. Já o tratamento teórico do algoritmo de decodificação MAP é essencial na implementação e compreensão do processo de decodificação iterativa turbo *L-M-ário*.

Nestes sistemas, analisaremos as características relacionadas a fonte de informação, a codificação, a modulação e a decodificação que nos permite estruturar um sistema turbo híbrido multinível mais eficiente que o sistema turbo multinível convencional.

Esta estrutura híbrida de codificação *L-M-ária* com modulação *Q-L-M-PSK*, proporciona comparações mais justas entre os sistemas turbo que possuem a mesma taxa de codificação e o mesmo número de estados no codificador turbo. Além disso, mantemos o perfeito mapeamento de bits para símbolos *Q-ários*, reduzimos a complexidade de codificação e decodificação, aumentamos a distância livre e o número de polinômios da matriz geradora do código *RSC*, tornando o sistema proposto mais eficiente que o sistema turbo convencional.

Capítulo 5

Resultados dos Códigos Turbo Híbridos Multiníveis

5.1 Introdução

Neste capítulo, apresentamos as curvas da taxa de erro de bit versus E_b/N_0, referentes as simulações realizadas para os códigos turbo híbridos multiníveis que possuem como entrada símbolos de informação proveniente de uma fonte definida sobre $\mathbb{Z}_Q$ ($Q = 2, 4, 8, \cdots$). As simulações visam avaliar de forma comparativa sistemas turbo *L-M-ário* com modulação do tipo *Q-L-M-PSK*. Os parâmetros, tais como o comprimento da seqüência de informação, a taxa de codificação turbo, o codificador RSC, as distâncias (D_{free} e D_{efet}), a cardinalidade do alfabeto e o número de iterações estão associados ao desempenho e a complexidade dos códigos turbo proposto.

5.2 Resultados

Inicialmente, organizamos os componentes do esquema turbo, analisando a contribuição dada por cada um deles nas simulações, e, em seguida, apresentamos os resultados obtidos por simulação.

Os codificadores convolucionais constituintes são codificadores *RSC*, não necessariamente idênticos, com taxa de codificação igual a 1/2 e com matriz geradora do tipo $g(D) = \left[1 \ \dfrac{g_2(D)}{g_1(D)}\right]$. Estes codificadores são definidos sobre os campos de números inteiros $\mathbb{Z}_L$ e $\mathbb{Z}_M$ e processam símbolos definidos sobre o anel $\mathbb{Z}_Q$. As distâncias livre e efetiva, definidas por Benedetto e Montorsi [26] para os sistemas binários, também são utilizadas na analise dos códigos *L-M-ários*. Estas distâncias efetiva e livre (D_{free} e D_{efet}) dos códigos turbo *L-M-ários* com entrada *Q-ária* são apresentadas nas Tabelas 5.1 e 5.2.

A Tabela 5.1 mostra o número de estados de cada codificador componente, a taxa de codificação turbo, os polinômios da matriz geradora dos codificadores componentes e as distâncias livre e efetiva dos códigos turbo **Binário**, **2-3-ário**, **Ternário** e **3-5-ário** com entrada binária.

C. Turbo	Estados	Taxa	$g_1(D)$	$g_2(D)$	D_{efet}	D_{free}
Binário	8	0.50	$1+D+D^3$	$1+D+D^2+D^3$	16	12
Binário	8	0.33	$1+D+D^3$	$1+D+D^2+D^3$	28	16
2-3-ário	*RSC-1*→8 *RSC-2*→ 9	0.43	$1+D+D^3$ $1+D+D^2$	$1+D+D^2+D^3$ $2+D^2$	12.15	11.64
Ternário	9	0.38	$1+D+D^2$	$2+D^2$	12.66	12.66
3-5-ário	*RSC-1*→9 *RSC-2*→ 25	0.33	$1+D+D^2$ $1+D+D^2$	$2+D^2$ $4+4D+3D^2$	13.05	11.76

Tabela 5.1: Codificadores com entrada binária.

A Tabela 5.2 mostra o número de estados de cada codificador componente, a taxa de codificação turbo, os polinômios da matriz geradora dos codificadores componentes e as distâncias livre e efetiva dos códigos turbo **Quaternário**, **5-ário**, **5-7-ário** e **7-ário** com entrada quaternária.

C. Turbo	Estados	Taxa	$g_1(D)$	$g_2(D)$	D_{efet}	D_{free}
Quaternário	16	0.5	$1+D+3D^2$	$2+D+2D^2$	9.65	8.0
5-ário	25	0.46	$1+D+D^2$	$4+4D+3D^2$	10.15	8.49
5-7-ário	*RSC-1*→25 *RSC-2*→ 49	0.43	$1+D+D^2$ $1+D+D^2$	$4+4D+3D^2$ $6+2D+4D^2$	10.59	9.12
7-ário	49	0.41	$1+D+D^2$	$6+5D+4D^2$	11.65	8.82

Tabela 5.2: Codificadores com entrada quaternária.

O entrelaçador s-aleatório utilizado foi escolhido por apresentar melhores propriedades aleatórias que os outros entrelaçadores apresentados na literatura.

O puncionador serve para apagar periodicamente símbolos de paridade de cada codificador componente, aumentando a taxa de codificação do codificador turbo. Variando o tamanho do alfabeto de entrada e da saída do codificador turbo, e utilizando ou não o puncionador, podemos obter taxas de codificação turbo mais flexíveis que as taxas convencionais 1/2 e 1/3.

Os símbolos de informação *Q-ários* e de paridade *L-ários* e *M-ários* são transmitidos através das respectivas modulações *Q-PSK*, *L-PSK* e *M-PSK*. O canal de comunicação é o *AWGN*.

Os decodificadores são de máximo *a posteriori*, MAP, organizados de acordo com o esquema de decodificação iterativa turbo *L-M-ário* da Figura 4.10. Estes decodificadores estão adaptados para processar a informação associada à modulação *Q-L-M-PSK*. O processo de decodificação iterativa retira a informação extrínseca, referente ao símbolos *L-ários* ou *M-ários* do decodificador prévio, e a usa como informação a priori, referente ao símbolos *Q-ários* no decodificador presente.

Nas simulações deste capítulo, mantemos o perfeito mapeamento de bits para símbolos e analisamos a capacidade de correção de erro, a complexidade de decodificação e a cardinalidade do alfabeto com relação à variação do número de estados e da taxa de codificação turbo de cada sistema comparado. As comparações nem sempre são justas, mas são um indicativo das contribuições destes esquemas de codificação turbo multiníveis.

5.2.1 Esquemas Binário, *2-3-ário*, Ternário e *3-5-ário* com Entrada Binária

Nesta subseção, apresentamos as curvas da probabilidade $P_e(b)$ de erro de bit versus E_b/N_0 dos esquemas de codificação turbo com símbolos de informação definidos sobre o campo $\mathbb{Z}_2$. Ou seja, são esquemas de codificação sistemáticos *2-3-ário*, ternário e *3-5-ário*, com seqüência de entrada binária, que utilizam as respectivas modulações *2-2-3-PSK*, *2-3-3-PSK* e *2-3-5-PSK*. No termo *2-3-5-PSK*, por exemplo, 2 é o tamanho da modulação *2-PSK* usada na transmissão dos bits de informação; 3 refere-se a modulação *3-PSK* usada na transmissão dos símbolos ternários do codificador *RSC-1* e 5 refere-se a modulação *5-PSK* usada na transmissão dos símbolos *5-ários* do codificador *RSC-2*.

Para comparação entre os sistemas turbo, fixamos o número de iterações em 8 e o comprimento da seqüência de informação em $N = 2000$ bits. A Figura 5.1 apresenta o desempenho dos esquemas turbo:

1. Binário com modulação *2-PSK*, taxa de codificação turbo 1/2 e matriz geradora dos codificadores *RSC-1* e *RSC-2*, $g(D) = \begin{bmatrix} 1 & \dfrac{1+D+D^2+D^3}{1+D+D^3} \end{bmatrix}$, com 8 estados. O codificador turbo possui 16 estados na treliça;

2. *2-3-ário* com modulação *2-2-3-PSK*, taxa 0.43, codificador *RSC-1* com matriz geradora $g(D) = \begin{bmatrix} 1 & \dfrac{1+D+D^2+D^3}{1+D+D^3} \end{bmatrix}$ (8 estados) e codificador *RSC-2* com matriz geradora $g(D) = \begin{bmatrix} 1 & \dfrac{1+2D^2}{1+D+D^2} \end{bmatrix}$ (9 estados). O codificador turbo possui 17 estados;

3. Ternário com modulação *2-3-3-PSK*, taxa 0.38 e codificadores *RSC-1* e *RSC-2* com $g(D) = \begin{bmatrix} 1 & \dfrac{1+2D^2}{1+D+D^2} \end{bmatrix}$ (9 estados). O codificador turbo possui 18 estados;

4. Binário com modulação *2-PSK*, taxa 1/3, codificadores *RSC-1* e *RSC-2* com $g(D) = \begin{bmatrix} 1 & \dfrac{1+D+D^2+D^3}{1+D+D^3} \end{bmatrix}$ (8 estados). O codificador turbo possui 16 estados.

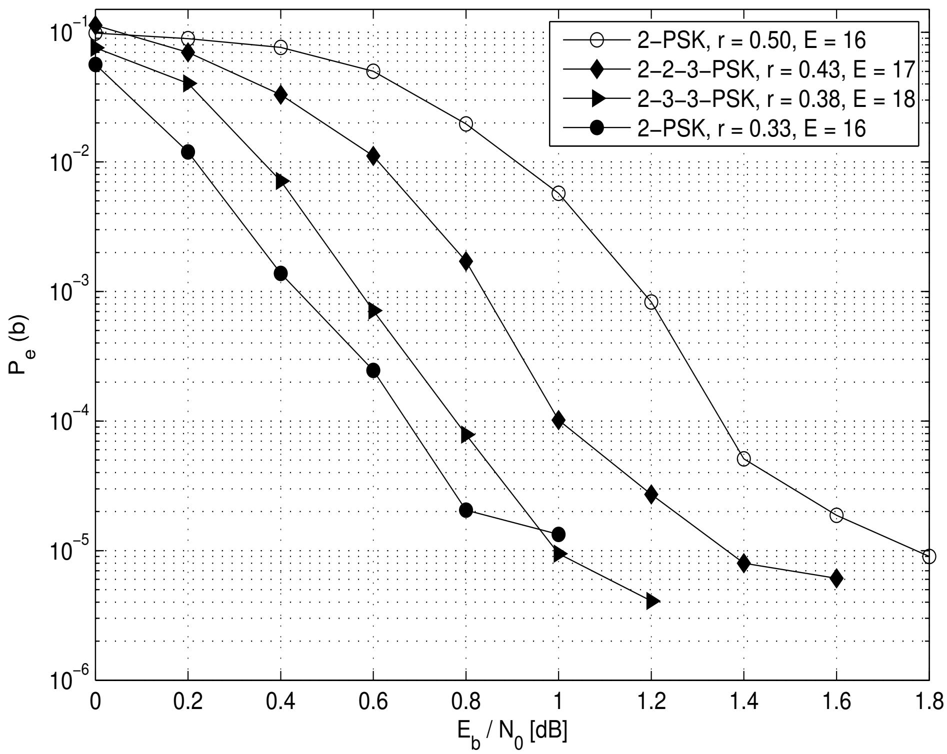


Figura 5.1: Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas binário com taxas 1/2 e 1/3, *2-3-ário* com taxa 0.43, ternário com taxa 0.3868.

De acordo com os parâmetros acima e as curvas mostradas na Figura 5.1, os respectivos esquemas *2-3-ário* com modulação *2-2-3-PSK* e ternário com modulação *2-3-3-PSK* apresentam ganhos de aproximadamente 0.4 e 0.8 dB, para probabilidade de erro de bit $P_e(b) \leq 10^{-5}$, em relação ao esquema binário de taxa 1/2. Além disso, apenas os esquemas com modulações *2-2-3-PSK* e *2-3-3-PSK* alcançam probabilidades de erro $P_e(b) < 10^{-5}$.

Observando os esquemas da Figura 5.1 para probabilidade de erro de bit $P_e(b) \leq 2 \times 10^{-5}$ notamos que: do esquema binário de taxa 0.5 para o esquema *2-3-ário* de taxa 0.436 perdemos 0.0637 na taxa de codificação e ganhamos 0.35 dB na relação sinal ruído; do esquema *2-3-ário* de taxa 0.436 para o esquema ternário de taxa 0.386 perdemos 0.0493 na taxa e ganhamos 0.33 dB; do esquema ternário de taxa 0.386 para o esquema binário de taxa 0.333 perdemos 0.0536 na taxa e ganhamos 0.12 dB. Como os tempos gastos na decodificação dos esquemas *2-3-ário*, ternário, binário taxa 1/2 e binário taxa 1/3 são praticamente os mesmos, os esquemas com modulações *2-2-3-PSK* e *2-3-3-PSK* são mais eficientes que os esquemas binários para qualquer relação sinal ruído E_b/N_0 $[dB]$.

A Figura 5.2 apresenta o desempenho dos esquemas turbo:

1. *2-3-ário* com modulação *2-2-3-PSK*, taxa 0.43, matriz geradora do codificador *RSC-1*, $g(D) = \left[1 \quad \dfrac{1+D^2}{1+D+D^2}\right]$, com 4 estados e matriz geradora do codificador *RSC-2*, $g(D) = \left[1 \quad \dfrac{1+2D^2}{1+D+D^2}\right]$, com 9 estados. O codificador turbo possui 13 estados;

2. *2-3-ário* com modulação *2-2-3-PSK*, taxa 0.43, matriz geradora do codificador *RSC-1*, $g(D) = \left[1 \quad \dfrac{1+D+D^2+D^3}{1+D+D^3}\right]$, com 8 estados e matriz geradora do codificador *RSC-2*, $g(D) = \left[1 \quad \dfrac{1+2D^2}{1+D+D^2}\right]$, com 9 estados. O codificador turbo possui 17 estados;

3. *2-3-ário* com modulação *2-2-3-PSK*, taxa 0.43, matriz geradora do codificador *RSC-1*, $g(D) = \left[1 \quad \dfrac{1+D+D^2+D^3+D^4}{1+D+D^4}\right]$, com 16 estados e matriz geradora do codificador *RSC-2*, $g(D) = \left[1 \quad \dfrac{1+2D^2}{1+D+D^2}\right]$, com 9 estados. O codificador turbo possui 25 estados;

Na Figura 5.2, o esquema *2-3-ário* com 25 estados apresenta ganhos de aproximadamente 0.4 e 0.2 dB, para probabilidade de erro de bit $P_e(b) \leq 10^{-5}$, em relação ao esquema *2-3-ário* com 13 e 17 estados, respectivamente. Além disso, para taxa $E_b/N_0 \geq 1.4$ $[dB]$, o esquema com 25 estados apresenta probabilidade de erro de bit 11 vezes menor que o esquema de 13 e 8 vezes menor que o esquema de 17 estados (*error floor*). Como o tempo de decodificação do esquema com 25 estados é, em média, 1.62 e 1.37 vezes maior que nos respectivos esquemas com 13 e 17 estados e estes sistemas são considerados de baixa complexidade, o esquema com 25 estados é mais eficiente que os esquemas de 13 e 17 estados para $P_e(b) \leq 10^{-4}$.

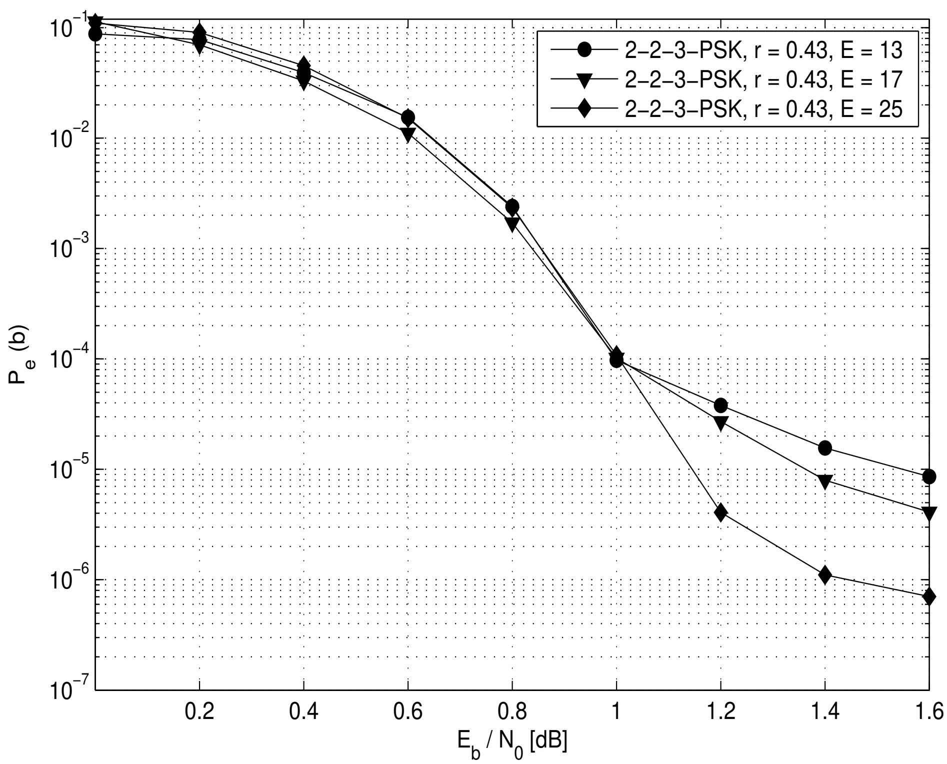


Figura 5.2: Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas *2-3-ário* com taxa 0.43 e o número de estados assumindo os valores 13, 17 e 25.

A Figura 5.3 apresenta o desempenho dos esquemas turbo:

1. binário com modulação *2-PSK*, taxa 0.333 e matriz geradora dos codificadores *RSC-1* e *RSC-2*, $g(D) = \begin{bmatrix} 1 & \dfrac{1+D^2+D^3+D^4}{1+D+D^4} \end{bmatrix}$, com 16 estados. O codificador turbo possui 32 estados;

2. *3-5-ário* com modulação *2-3-5-PSK*, taxa 0.335, matriz geradora do codificador *RSC-1*, $g(D) = \begin{bmatrix} 1 & \dfrac{1+2D^2}{1+D+D^2} \end{bmatrix}$, com 27 estados e matriz geradora do codificador *RSC-2*, $g(D) = \begin{bmatrix} 1 & \dfrac{4+4D+3D^2}{1+D+D^2} \end{bmatrix}$, com 5 estados. O codificador turbo possui 32 estados.

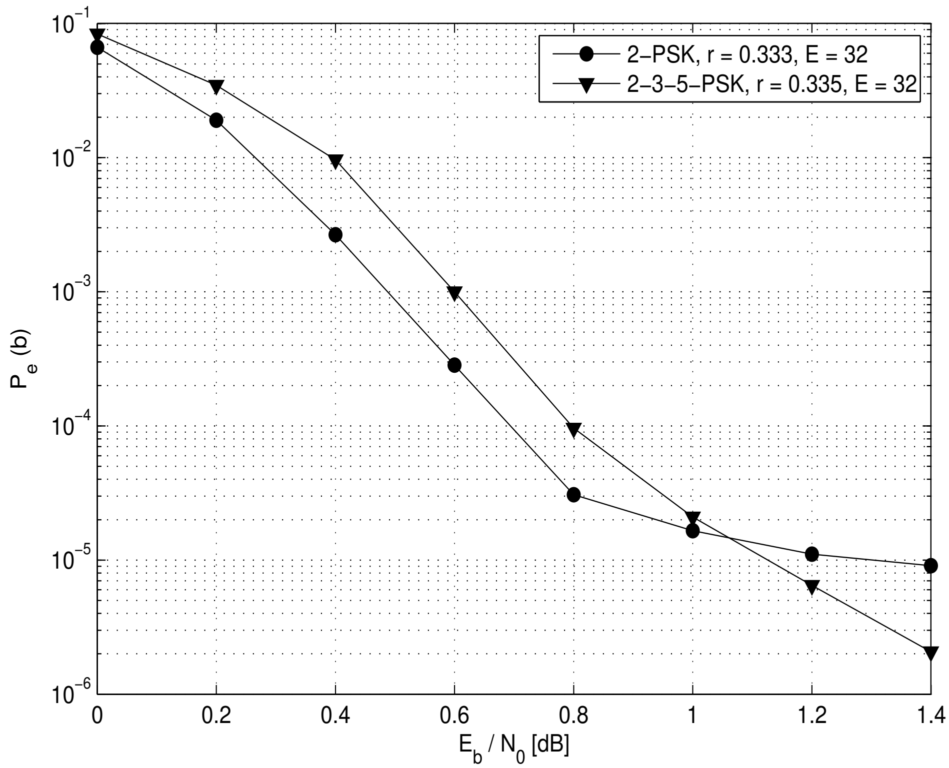


Figura 5.3: Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas binário com taxa 0.333 e *3-5-ário* com taxa 0.335 e 32 estados.

Na Figura 5.3, o esquema *3-5-ário* com modulação *2-3-5-PSK* apresenta melhor desempenho que o esquema binário com modulação *2-PSK* para $E_b/N_0 \geq 1.0$ $[dB]$. Como o tempo gasto na decodificação do esquema *3-5-ário* é em média 1.11 vezes maior que no esquema binário e estes sistemas são considerados de baixa complexidade, para sistemas turbo que necessitem de probabilidade de erro de bit $P_e(b) \leq 2 \times 10^{-5}$, o esquema *3-5-ário* é mais eficiente que o esquema binário.

De acordo com as figuras 5.1 e 5.3 os esquemas híbridos com entrada binária são mais eficientes que os esquemas turbo binários padrão com taxas 1/2 e 1/3.

5.2.2 Esquemas Quaternário, *5-ário*, *5-7-ário* e *7-ário* com Entrada Quaternária

Nesta subseção, apresentamos as curvas da probabilidade $P_e(b)$ de erro de bit versus E_b/N_0 dos esquemas turbo com símbolos de informação definidos sobre o anel $\mathbb{Z}_4$. Ou seja, são esquemas do tipo: quaternário, *5-ário*, *5-7-ário* e *7-ário*, com entrada quaternária, que utilizam as modulações *4-PSK*, *4-5-5-PSK*, *4-5-7-PSK* e *4-7-7-PSK*, respectivamente.

Nos sistemas turbo com entradas quaternárias, fixamos o número de iterações em 8 e o comprimento da seqüência de informação em $N = 1000$ símbolos.

A Figura 5.4 apresenta o desempenho dos esquemas turbo:

1. Quaternário com modulação *4-PSK*, taxa de codificação turbo igual a 1/2 e matriz geradora dos codificadores *RSC-1* e *RSC-2*, $g(D) = \left[1 \;\; \dfrac{2+D+2D^2}{1+D+3D^3}\right]$, com 16 estados. O codificador turbo possui portanto 32 estados;

2. *5-ário* com modulação *4-5-5-PSK*, taxa de codificação igual a 0.46 e matriz geradora dos codificadores *RSC-1* e *RSC-2*, $g(D) = \left[1 \;\; \dfrac{4+4D+3D^2}{1+D+D^3}\right]$, com 25 estados. O codificador turbo possui 50 estados;

3. *5-7-ário* com modulação *4-5-7-PSK*, taxa 0.43, matriz geradora do codificador *RSC-1*, $g(D) = \left[1 \;\; \dfrac{4+4D+3D^2}{1+D+D^2}\right]$, com 25 estados e matriz geradora do codificador *RSC-2*, $g(D) = \left[1 \;\; \dfrac{6+2D+4D^2}{1+D+D^2}\right]$, com 49 estados. O codificador turbo possui portanto um total de 74 estados;

4. *7-ário* com modulação *4-7-7-PSK*, taxa 0.41 e matriz geradora dos codificadores *RSC-1* e *RSC-2*, $g(D) = \left[1 \;\; \dfrac{6+2D+4D^2}{1+D+D^3}\right]$, com 49 estados. O codificador turbo possui 98 estados;

De acordo com os parâmetros acima e as curvas mostradas na Figura 5.4, o esquema *5-7-ário* com taxa 0.43 apresenta: um ganho maior que 0.3 dB para probabilidade de erro de bit $P_e(b) \leq 8 \times 10^{-6}$, em relação ao esquema quaternário com taxa 1/2; um ganho maior que 0.2 dB para probabilidade de erro de bit $P_e(b) \leq 6 \times 10^{-7}$, em relação ao esquema *5-ário* com taxa 0.46; um ganho maior que 0.15 dB para probabilidade de erro de bit $P_e(b) = 6 \times 10^{-7}$, em relação ao esquema *7-ário* com taxa 0.41. Além disso, este esquema *5-7-ário* apresenta patamar de saturação da $P_e(b)$ ou"error floor" 15, 5 e 0 (zero) vezes menor que o "error floor" dos esquemas quaternário, *5-ário* e *7-ário*, respectivamente.

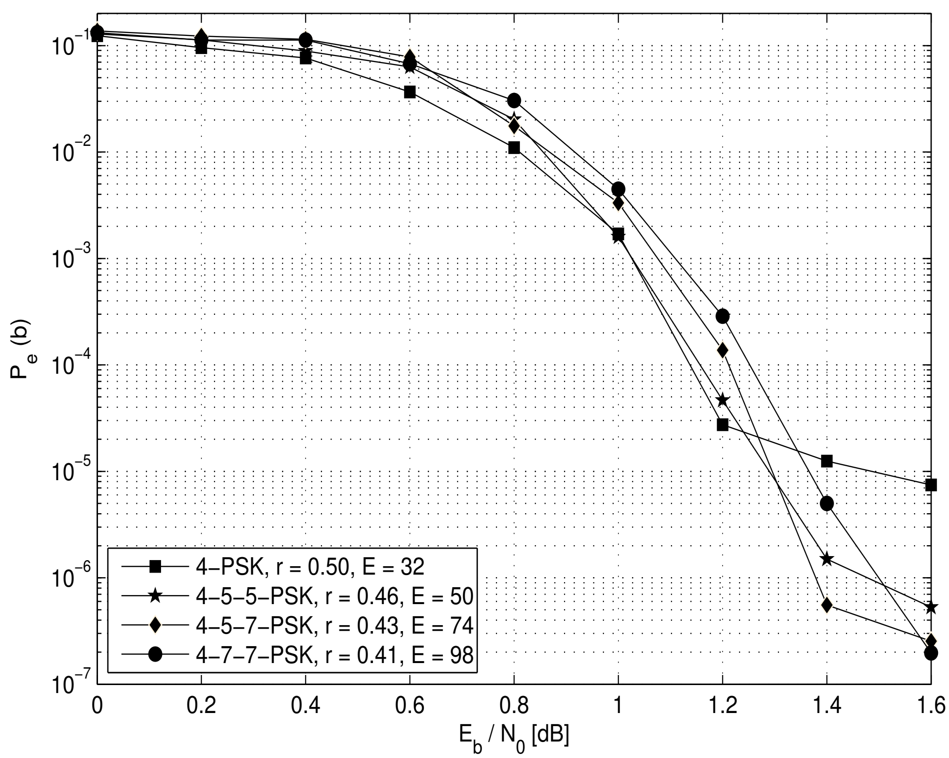


Figura 5.4: Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas quaternário, *5-ário*, *5-7-ário* e *7-ário* com taxas de codificação iguais a 0.5, 0.46, 0.43 e 0.41, respectivamente.

O tempo gasto na decodificação do esquema *5-7-ário* é em média 2.47 vezes maior que o tempo gasto no esquema quaternário, 1.63 vezes maior que o tempo gasto no esquema *5-ário* e 0.72 vezes menor que o tempo gasto no esquema *7-ário*. Portanto, o esquema *5-7-ário* é mais eficiente que os esquemas quaternário, *5-ário*, e *7-ário* para qualquer probabilidade de erro de bit $P_e(b) \leq 10^{-5}$ $[dB]$. As comparações podem ser consideradas injustas, pois o número de estados e a taxa de codificação são diferentes para cada sistema. Todavia, estas comparações indicam o bom desempenho dos sistemas quaternário, *5-ário*, *5-7-ário* e *7-ário* com entrada quaternária.

A Figura 5.5 apresenta o desempenho dos esquemas turbo:

1. Binário com modulação *2-PSK*, taxa 0.333 e matriz geradora dos codificadores *RSC-1* e *RSC-2*, $g(D) = \left[1 \quad \dfrac{1+D+D^2+D^3+D^4}{1+D+D^4}\right]$, com 16 estados. O codificador turbo possui 32 estados e o comprimento da seqüência de informação é $N = 2000$ bits;

2. *3-5-ário* com modulação *2-3-5-PSK*, taxa 0.335, matriz geradora do codificador *RSC-1*, $g(D) = \left[1 \quad \dfrac{2+D^2+2D^3}{1+D+D^2+D^3}\right]$, com 27 estados e matriz geradora do codificador *RSC-2*, $g(D) = \left[1 \quad \dfrac{3+3D}{1+3D}\right]$, com 5 estados. O codificador turbo possui 32 estados e o comprimento da seqüência de informação é $N = 2000$ bits;

3. *5-7-ário* com modulação *4-5-7-PSK*, taxa 0.338, matriz geradora do codificador *RSC-1*, $g(D) = \left[1 \quad \dfrac{1+D+D^2}{4+4D+3D^2}\right]$, com 25 estados e matriz geradora do codificador *RSC-2*, $g(D) = \left[1 \quad \dfrac{4+3D}{1+5D}\right]$, com 7 estados. O codificador turbo possui 32 estados e o comprimento da seqüência de informação é $N = 2000$ bits. Esta taxa 0.338 foi obtida por meio de um puncionador que apaga apenas os símbolos de paridade ímpares do codificador *RSC-1* e mantém todos os símbolos de paridade do codificador *RSC-2*.

Na Figura 5.5, os esquemas binário e *3-5-ário* apresentam ganhos maiores que 0.2 e 0.1 dB, para probabilidade de erro de bit entre 10^{-2} e 2×10^{-5}, em relação ao esquema *5-7-ário*. Para probabilidade de erro de bit $P_e(b) \leq 10^{-5}$, os esquemas *3-5-ário* e *5-7-ário* apresentam melhor desempenho que o esquema binário . Como o tempo de decodificação dos esquemas binário e *3-5-ário* são 2.33 e 2.11 vezes maiores que o tempo do esquema *5-7-ário* e estes sistemas são considerados de média complexidade, para sistemas que possuam largura de faixa limitada, conclui-se que o esquema *5-7-ário* é mais eficiente que os esquemas binário e *3-5-ário*.

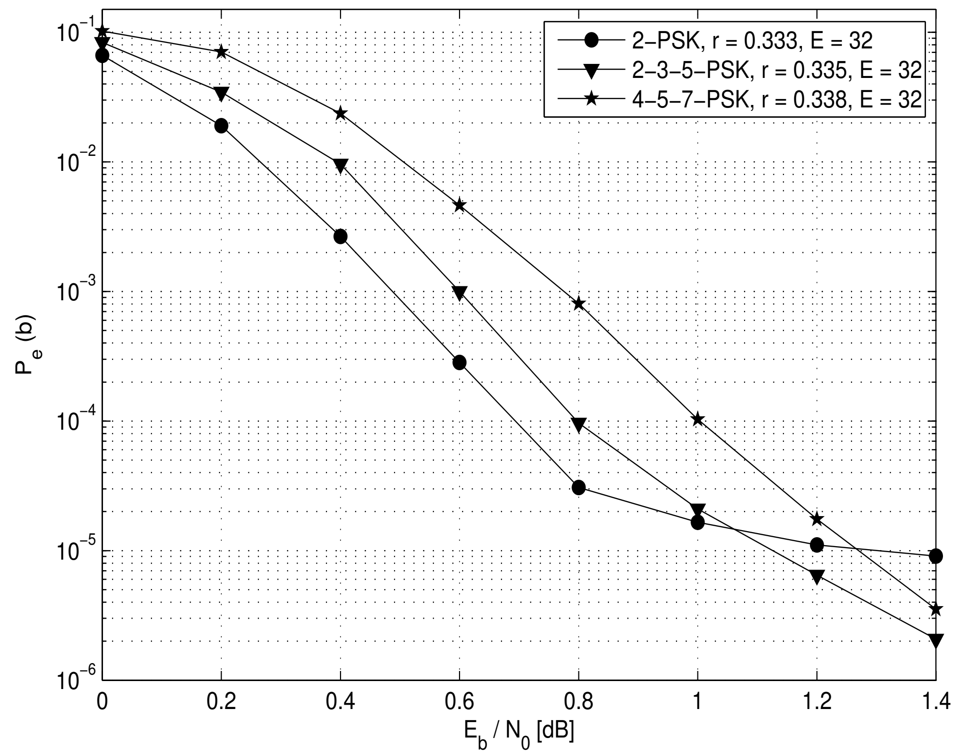


Figura 5.5: Desempenho da $P_b(e) \times E_b/N_0$ dos esquemas binários, *3-5-ário* e *5-7-ário* com taxas iguais a 0.33 e 32 estados.

Note que nos códigos turbo do tipo *L-M-ário* com entrada quaternária, apresentados nesta subseção, existe uma maior variação no número de estados e, portanto, a comparação entre os sistemas apresentados não são feitas fixando todos os parâmetros dos códigos de forma idêntica entre os diversos esquemas. Por outro lado, os códigos com entradas quaternárias processam 1000 símbolos em vez de 2000 bits e ocupam a mesma largura de faixa dos códigos com entradas binárias.

Os códigos turbo híbridos com entrada quaternária apresentam um patamar de saturação para $P_e(b)$ "error floor" mais baixo e possuem baixa complexidade de decodificação (para um o número de estados menor do que 50) quando comparados com os códigos com entradas binárias. Logo, os esquemas com entrada quaternária podem ser considerados mais eficientes que os esquemas com entrada binária por apresentar melhor desempenho e ocupar mesma largura de faixa.

5.3 Conclusão

Neste capítulo apresentamos as curvas de taxa de erro de bit versus E_b/N_0 referente às simulações realizadas para os códigos turbo híbridos multiníveis com modulação *Q-L-M-PSK*. Estes sistemas híbridos possuem codificadores definidos sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$ e seqüência de entrada definida sobre $\mathbb{Z}_Q$ $(Q = 2, 4, 8, \cdots)$. Esta estrutura de codificação híbrida proporciona ao sistema turbo:

1. perfeito mapeamento dos símbolos do alfabeto *Q-ário* em bits, pois a cardinalidade do anel $\mathbb{Z}_Q$ é uma potência de 2 ($\mathbb{Z}_4$, $\mathbb{Z}_8$, $\mathbb{Z}_{16}$, $\cdots$);

2. maior número de polinômios para a matriz geradora do codificador definido sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$;

3. maiores distâncias (D_{free} e D_{efet}) do que no esquema turbo multinível convencional;

4. maior variação da taxa de codificação turbo;

5. menor complexidade de codificação e decodificação que o esquema turbo convencional definido sobre o campo $\mathbb{Z}_M$;

Logo, para sistemas que necessitem perfeito mapeamento de símbolos *Q-ário* em bits, este sistema é mais vantajoso que o sistema convencional por apresentar melhor desempenho, erro floor mais baixo e complexidade de codificação e decodificação próximas a complexidade do sistema turbo binário convencional.

Capítulo 6

Conclusões

6.1 Resumo da tese

Neste trabalho, apresentamos as estruturas de codificação e decodificação para minimização da interferência do ruído em sistemas turbo híbridos multiníveis, definidos sobre campos e anéis de inteiros módulo-M, $\mathbb{Z}_M$. A estrutura de codificação é formada por um esquema de codificação entrelaçado que visa reduzir a correlação entre a informação extrínseca e a entrada de dados a cada iteração do processo de decodificação. Já a estrutura de decodificação é formada por dois decodificadores componentes que atuam de forma conjunta, aumentando a confiabilidade sobre os símbolos decodificados.

Para todos os casos considerados, a escolha dos componentes de codificação e decodificação permite destacar os sistemas mais imunes à interferência do ruído, ou seja, são sistemas bastante eficientes, que apresentam grande capacidade de correção de erro e baixa complexidade de decodificação, sem sacrificar a taxa de codificação, a potência do sinal e a largura de banda.

Na primeira parte desta tese apresentamos os códigos turbo multiníveis definidos sobre os anéis e campos de números inteiros $\mathbb{Z}_M$. No sistema *M-ário* definido sobre o anel de inteiros $\mathbb{Z}_M$, existe um número reduzido de polinômios para a matriz geradora do codificador *RSC*, pois nem todo elemento do anel $\mathbb{Z}_M$ possui inverso multiplicativo. Por outro lado, os anéis cuja cardinalidade é um potência positiva de 2, por exemplo $\mathbb{Z}_4$, $\mathbb{Z}_8$, $\mathbb{Z}_{16}$, $\cdots$, proporcionam perfeito mapeamento dos símbolos *M-ários* em bits.

No sistema *M-ário* definido sobre um campo $\mathbb{Z}_M$, pode-se obter um maior número de polinômios para a matriz geradora do codificador *RSC*, mas não existe um perfeito mapeamento da fonte binária nos símbolos de $\mathbb{Z}_M$.

As curvas simuladas mostram que os códigos turbo multiníveis apresentam bom desempenho com baixa relação sinal ruído e, em geral, quanto maior o tamanho do alfabeto maior será a eficiência espectral do sistema turbo *M-ário*, pois 1 símbolo transmitido corresponde a $log_2 M$ bits transmitidos no sistema *M-ário*. Além disso, para cada sistema turbo *M-ário* existe uma faixa de valores para os parâmetros (tamanho do alfabeto, comprimento da seqüência de informação, número de iterações, taxa de codificação e codificador *RSC*) para os quais o sistema turbo apresenta melhor desempenho.

Os resultados simulados mostram que o esquema turbo ternário, cujos codificadores componentes possuem 9 estados, apresenta melhor desempenho e menor complexidade de decodificação que os demais esquemas turbo *M-ários* analisados.

Na segunda parte desta tese apresentamos uma nova metodologia para analisar os códigos turbo híbridos multiníveis com modulação do tipo *Q-L-M-PSK*. O tratamento teórico referente ao esquema de codificação e decodificação estabelece procedimentos que permitem avaliar a contribuição do codificador, do entrelaçador, do puncionador, do modulador e do decodificador no desempenho do sistema *L-M-ário*.

A relação fechada que existe entre os símbolos de informação definidos sobre o anel de inteiros $\mathbb{Z}_Q$ e a modulação *Q-PSK* e os símbolos de paridade definidos sobre os campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$ e as respectivas modulações *L-PSK* e *M-PSK*, conserva os aspectos de distância usados na escolha dos melhores codificadores componentes do esquema turbo. Este sistema fornece perfeito mapeamento de bits para símbolos, maiores distâncias livre e efetiva, maior variação na taxa de codificação turbo, maior número de polinômios na matriz geradora e menor complexidade de codificação e de decodificação que no esquema turbo padrão definido sobre o campo

$\mathbb{Z}_M$.

As curvas da taxa de erro de bit versus E_b/N_0 referentes as simulações realizadas para os esquemas *2-3-ário*, ternário e *3-5-ário* com entrada binária comprovam que estes sistemas são mais eficientes que o sistema turbo binário padrão. Já as curvas referentes aos esquemas *5-ário*, *5-7-ário* e *7-ário* com entrada quaternária comprovam que estes sistemas híbridos são mais eficientes que o sistema turbo quaternário padrão. Além disso, os sistemas com entrada quaternária ocupam a mesma largura de faixa e processa a metade da quantidade de elementos processada pelos sistemas com entrada binária, sendo assim mais eficientes que os sistemas com entrada binária.

6.2 Contribuições da tese

Principais contribuições deste trabalho:

- Apresentamos os códigos turbo multiníveis definidos sobre os anéis e campos de números inteiros módulo-M, $\mathbb{Z}_M$;

- Descrevemos toda estrutura de codificação e decodificação, obtivemos as expressões do algoritmo de decodificação iterativo e destacamos os componentes dos esquemas de codificação e decodificação que proporcionam melhor desempenho nos sistemas turbo multiníveis;

- Realizamos as simulações e analisamos os resultados de forma comparativa para obter as conclusões sobre o desempenho de cada sistema *M-ário*;

- Propusemos os códigos turbo híbridos multiníveis que proporcionam perfeito mapeamento de bits para símbolos;

- Apresentamos a estrutura de codificação que processa símbolos de informação do anel $\mathbb{Z}_Q$ e produz símbolos de paridades pertencentes aos campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$ e o esquema de modulação usado na transmissão destes símbolos provenientes do codificador turbo;

- Obtivemos as expressões do algoritmo de decodificação e destacamos a contribuição da informação extrínseca referente aos símbolos de paridade dos campos $\mathbb{Z}_L$ e $\mathbb{Z}_M$ no processo de decodificação iterativo turbo *L-M-ário*;

- Mostramos o desempenho dos códigos turbo *L-M-ários* em relação aos códigos turbo *M-ários* e descrevemos os sistema mais eficientes em termos da probabilidade de erro de bit pela relação sinal ruído.

6.3 Trabalhos Futuros

Com base nos códigos turbo multiníveis convencionais e nos códigos turbo híbridos multiníveis com entrada sobre $\mathbb{Z}_Q$ ($\mathbb{Z}_2$, $\mathbb{Z}_4$, $\mathbb{Z}_8, \cdots$), sugerimos os trabalhos:

- Códigos multiníveis de baixa densidade de paridade (*LDPC*);

- Adaptação dos códigos turbo multiníveis para codificação espaço temporal.

6.4 Publicações

- J. Barros, R. Baldini, "Códigos Turbo Quaternários," In:XXII Simpósio Brasileiro de Telecomunicações, pp. 10, Set. 2005.

- J. Barros, R. Baldini, "5-Ary Turbo Codes With 4-5-PSK Modulation," In:VI International Telecommunications Symposium (ITS2006), pp. 180-183, Set. 2006.

- J. Barros, R. Baldini, "Códigos Turbo *M-ários* com Modulação *M-PSK*," In:XXV Simpósio Brasileiro de Telecomunicações, pp. 16-17, Set. 2007.

Bibliografia

[1] C. Shannon, “A Mathematical Theory of Communication,” The Bell System Technical Journal, p. 379–423, 1948.

[2] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes,” Proc. Int. Conf. Comm., pp. 1064–1070, May 1993.

[3] C. Berrou, A. Glavieux, “Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes,” IEEE Trans. Comm., vol. 44, n. 10, pp. 1261–1271, Oct. 1996.

[4] J. Barros, R. Baldini, “Códigos Turbo Quaternários,” In:XXII Simpósio Brasileiro de Telecomunicações, pp. 10, Set. 2005.

[5] J. Barros, R. Baldini, “5-Ary Turbo Codes With 4-5-PSK Modulation,” In:VI International Telecommunications Symposium (ITS2006), pp. 180–183, Set. 2006.

[6] J. Barros, R. Baldini, “Códigos Turbo M-ários com Modulação M-PSK,” In:XXV Simpósio Brasileiro de Telecomunicações, Set. 2007.

[7] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” IEEE Trans. Inform. Theory, vol. 42, pp. 429–445, Mar. 1996.

[8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” IEEE Trans. Inform. Theory, pp. 284–287, Mar. 1974.

[9] G. Ungerboeck, “Channel coding with multilevel/phase signals,” IEEE Trans. Inform. Theory, pp. 55–67, Jan. 1982

[10] P. Robertson, “Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes,” Proc. Globecom. pp. 1298–1303, 1994.

[11] D. Divsalar and F. Pollara, “Turbo codes for PCS application,” Proc. 1995 Int. Conf. Comm., pp. 54–59.

[12] K. Sayood and J. C. Borkenhagen, “Use of residual redundancy in the design of joint source and channel coders,” IEEE Trans. Comm., vol. 39, pp. 838–846, June 1991.

[13] K. Sayood, F. Liu, and J. D. Gibson, “A constrained joint source/channel coder design,” IEEE J. Select. Areas Commun., vol. 12, pp. 1584–1593, Dec. 1994.

[14] R. Baldini F. and P. G. Farrell, “Coded Modulation Based on Rings of Integers Modulo-q, Parte-I: Block Codes,” IEEE Proc. Commun., pp. 129–136, Vol. 141, n. 3, Jun. 1994.

[15] R. Baldini F. and P. G. Farrell, “Coded Modulation Based on Rings of Integers Modulo-q, Parte-II: Convolutional Codes,” IEEE Proc. Commun., pp. 137–142, Vol. 141, n. 3, Jun. 1994.

[16] A. Ruscitto and E. M. Biglieri, “Joint Source and Channel Coding using Turbo Codes over Rings,” IEEE Trans. on Comms., vol. 46, n. 8, Aug. 1998.

[17] J. Berkmann, “A Symbol-by-Symbol Map Decoding Rule for Linear Codes over Rings using the Dual Codes,” Proc. of ISIT’, Cambridge, MA, USA, Aug. 1998, pp. 16–21.

[18] Jason P. Woodard and Lajos Hanzo, “Comparative Study of Turbo Decoding Techniques: An Overview,” IEEE Trans. Veh. Technol, Vol. 49, n. 6, pp. 2208–2233, Nov. 2000.

[19] W. E. Ryan, “A Turbo Code Tutorial,”Proc. IEEE Globecom’98, 1998.

[20] Rolf Johannesson, "Some Structural Properties of Convolutional Codes over Rings," IEEE Trans. Inform. Theory, pp. 839–845, Mar. 1998.

[21] M. F. Atiyah and I. G. MacDonald, "Commutative Algebra," Reading, MA: Addison - Wesley, 1969.

[22] Chris Heegard and Stephen B. Wicker, "Turbo Coding," Kluwer Academic Publishers pp. 10–63.

[23] K. S. Andrews, C. Heegard and D. Kozen, "Interleaver Desingn Methods for Turbo Codes", Proceedings of the 1998. International Symposium on Information Theory, pg. 420.

[24] S. Dolinar and D. Divsalar, "Weight distributions for Turbo Codes using random and nonrandom permutations", TDA Progress Report 42–121, JPL, August 1995.

[25] F. Fagnini and S. Zampieri, "System-Theoretic Properties of Convolutional Codes Over Rings," IEEE Trans. Inform. Theory, vol. 47, N^o. 6, Sept. 2001, pp. 2256–2274.

[26] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," IEEE Trans. Comm., vol. 44, N^o. 5, May. 1996, pp. 591–600.

[27] Andrew C. Reid, T. Aaron Gulliver and Desmond P. Taylor, "Rate-1/2 Component Codes for Nonbinary Turbo Codes," IEEE Trans. Commun., pp. 1417–1422, Vol. 53, n. 9, Sept. 2005.

[28] Rankin, David M., "Single Parity Check Product Codes and Iterative Decoding". Ph.D. Thesis, University of Canterbury, Christchurch, New Zealand, May/2001