



**UNIVERSIDADE FEDERAL DE PERNAMBUCO**  
**CENTRO DE TECNOLOGIA E GEOCIÊNCIAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**CODIFICAÇÃO ITERATIVA PARA O**  
**CANAL ADITIVO COM DOIS**  
**USUÁRIOS BINÁRIOS**

Elaborado por:

Maria de Lourdes Melo Guedes Alcoforado

Aluna de doutorado do PPGEE

**Recife, Janeiro de 2005.**

**UNIVERSIDADE FEDERAL DE PERNAMBUCO**  
**CENTRO DE TECNOLOGIA E GEOCIÊNCIAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**CODIFICAÇÃO ITERATIVA PARA O CANAL  
ADITIVO COM DOIS USUÁRIOS BINÁRIOS**

por

**MARIA DE LOURDES MELO GUEDES ALCOFORADO**

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para a obtenção do grau de Doutor em Engenharia Elétrica.

**ORIENTADOR: VALDEMAR CARDOSO DA ROCHA Jr., PhD.**

Recife, Janeiro de 2005.

**A354      Alcoforado, Maria de Lourdes Melo Guedes**

Codificação iterativa para o canal aditivo com dois usuários binários / Maria de Lourdes Melo Guedes Alcoforado. – Recife: O Autor, 2005.

116 f. : il., fig., tabs., gráfs.

Inclui referências bibliográficas.

Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Depto. de Engenharia Elétrica, 2005.

1. Engenharia elétrica. 2. Telecomunicações – códigos corretores de erro e canais de acesso múltiplo. 3. Canais de comunicação. 4. Algoritmos (MATLAB) – implementação. 5. Códigos turbo. I. Título.

621 CDD (22 ed.)

UFPE/BCTG/2005-06



Universidade Federal de Pernambuco  
*Pós-Graduação em Engenharia Elétrica*

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE  
TESE DE DOUTORADO DE

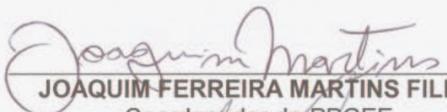
**MARIA DE LOURDES MELO GUEDES ALCOFORADO**

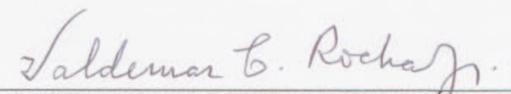
TÍTULO

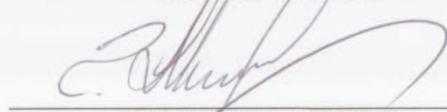
**"CODIFICAÇÃO ITERATIVA PARA O CANAL ADITIVO COM DOIS  
USUÁRIOS BINÁRIOS"**

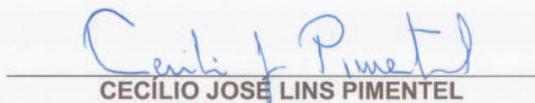
A comissão examinadora composta pelos professores:  
VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE, CECÍLIO  
JOSÉ LINS PIMENTEL, DES/UFPE, HÉLIO MAGALHÃES DE  
OLIVEIRA, DES/UFPE, GAREGIN MARKARIAN, Leeds (UK) E  
MARCELO SAMPAIO DE ALENCAR, DEE/UFCG, sob a presidência do  
prof. Joaquim Ferreira Martins Filho, Coordenador do Programa de Pós-  
Graduação em Engenharia Elétrica, consideram a candidata **MARIA DE  
LOURDES MELO GUEDES ALCOFORADO APROVADA.**

Recife, 11 de fevereiro de 2005.

  
\_\_\_\_\_  
JOAQUIM FERREIRA MARTINS FILHO  
Coordenador do PPGEE

  
\_\_\_\_\_  
VALDEMAR CARDOSO DA ROCHA JUNIOR  
Orientador e Membro Titular Interno

  
\_\_\_\_\_  
GAREGIN MARKARIAN  
Membro Titular Externo

  
\_\_\_\_\_  
CECÍLIO JOSÉ LINS PIMENTEL  
Membro Titular Interno

  
\_\_\_\_\_  
MARCELO SAMPAIO DE ALENCAR  
Membro Titular Externo

  
\_\_\_\_\_  
HÉLIO MAGALHÃES DE OLIVEIRA  
Membro Titular Interno

Aos meus pais, Ayrton e Maria de Lourdes, incansáveis na luta pela formação moral e intelectual dos seus filhos, aqueles que me apoiaram em todos os momentos e, principalmente, estimularam e motivaram a conclusão desta etapa de minha vida, incentivando-me a prosseguir e nunca desistir daquilo em que acredito, quaisquer que sejam os obstáculos.

Aos meus queridos pais, amigos em mais esta jornada, minha eterna gratidão. Dedico-lhes com meu reconhecimento, pois sem vocês a minha existência e deste trabalho não teriam sentido.

# Agradecimentos

Ao professor Dr. Valdemar Cardoso da Rocha Jr. pelo encorajamento, dedicação e segura orientação, demonstrados ao longo de todo este trabalho e principalmente, pelo constante apoio e amizade a mim dedicados, meus mais sinceros agradecimentos.

Ao professor Dr. Garik Markarian por compartilhar seus valiosos conhecimentos, durante o período passado por mim na *University of Leeds, UK*.

Aos meus queridos irmãos, Cláudia, Ayrton, Ricarda e Sérgio, por nossa grande amizade e companherismo, por sempre estarem torcendo por mim.

Ao meu querido marido Francisco, pelas ajudas de sempre, pela paciência, pelo permanente apoio e incentivo, por sua importância na minha vida e por compreender as minhas ausências.

© Maria de Lourdes Melo Guedes Alcoforado, 2005

Resumo da Tese apresentada à UFPE como parte dos requisitos necessários  
para a obtenção do grau de Doutor em Engenharia Elétrica.

# **CODIFICAÇÃO ITERATIVA PARA O CANAL ADITIVO COM DOIS USUÁRIOS BINÁRIOS**

**Maria de Lourdes Melo Guedes Alcoforado**

Janeiro/2005

Orientador: Valdemar Cardoso da Rocha Jr., PhD.

Área de Concentração: Comunicações.

Palavras-chave: Canais de acesso múltiplos, códigos turbo, CCMA.

Número de Páginas: 01.

Esta tese utiliza códigos turbo em sistemas de comunicações, para emprego em canais de acesso múltiplo. Tais canais permitem o acesso simultâneo a mais de um usuário, com a saída de cada canal sendo uma combinação dos sinais enviados pelos usuários ativos. Em particular é dada ênfase ao caso em que dois usuários binários podem transmitir simultaneamente em um canal aditivo para um único receptor. Este canal é chamado de canal aditivo com dois usuários binários (2-BAC). São feitas implementações de alguns sistemas codificados para a simulação de códigos e de decodificadores em presença de ruído branco Gaussiano aditivo, e é estabelecida uma condição de decodibilidade única para códigos de treliça usados no 2-BAC. A partir desta condição é possível uma visualização clara da utilização de códigos convolucionais (códigos lineares) ou de códigos de treliça (códigos não-lineares) no 2-BAC, até então não existentes na literatura técnica específica. É apresentada uma técnica para construção de códigos de treliça unicamente decodificáveis para o 2-BAC, bem como é introduzido um novo esquema de codificação colaborativa para acesso múltiplo (CCMA) que garante a decodibilidade única para o 2-BAC. Os esquemas introduzidos não limitam a taxa total a valores abaixo da capacidade. São implementados, por meio de simulação, codificadores e decodificadores para o 2-BAC usando códigos de treliça e esquemas CCMA. Através das condições de decodibilidade única encontradas, juntamente com os sistemas de codificação/decodificação turbo ficou demonstrada a viabilidade da utilização prática de códigos para o 2-BAC que possibilitam a obtenção de baixas probabilidades de erro na saída do decodificador, com equipamento de baixa complexidade. É proposta também uma aplicação de códigos para o 2-BAC para melhoria do desempenho da rede Ethernet e é estabelecida uma condição de decodibilidade única sobre uma classe de códigos para o 2-BAC quase-síncrono.

Abstract of Thesis presented to UFPE as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

# **ITERATIVE CODING FOR THE TWO USER BINARY ADDER CHANNEL**

**Maria de Lourdes Melo Guedes Alcoforado**

January /2005

Supervisor: Valdemar Cardoso da Rocha Jr., PhD.

Area of Concentration: Communications.

Keywords: Multiple access channel, turbo codes, CCMA.

Number of Pages: 01.

This thesis is concerned with an investigation of iterative coding schemes for the two-user binary adder channel (2-BAC).

A coding scheme is introduced for the 2-BAC using a single convolutional code which permits efficient decoding. The constructed two-user trellis codes have a significant practical advantage because iterative decoding techniques used in the single-user case, related to Viterbi decoding and to the maximum a posteriori probability rule, apply to the two-user case after minor modifications.

A novel coding scheme is presented for the 2-BAC using a two-dimensional block code (product code), also called collaborative coding multiple access, for practical application which uses turbo decoding for improving performance.

A new hierarchical coding scheme is introduced, which allows an increase of the overall traffic over existing Ethernet networks, with backward compatibility with existing users. In this scheme new users will be added with the proposed code, while the existing users will be able to continue receiving services without change of hardware.

Finally, a condition is introduced for unique decodability over a quasi-synchronous two-user binary adder channel for a class of codes, and simulation results are presented the over a synchronous and a quasi-synchronous two-user binary adder channel in the presence of white Gaussian noise.

# Sumário

<b>1</b>	<b>Motivação e Plano de Tese</b>	<b>2</b>
1.1	Introdução . . . . .	2
1.2	Plano de Tese . . . . .	4
<b>2</b>	<b>Canais de Acesso Múltiplo</b>	<b>6</b>
2.1	Teoria da Informação para Canais de Acesso Múltiplo . . . . .	7
2.2	Canal Aditivo com Dois Usuários Binários . . . . .	9
2.2.1	Decodibilidade Única . . . . .	11
2.2.2	Códigos Lineares para o 2-BAC . . . . .	12
2.2.3	Algumas Construções de Códigos para o 2-BAC . . . . .	13
<b>3</b>	<b>Códigos Convolucionais</b>	<b>15</b>
3.1	Conceitos Básicos . . . . .	15
3.2	Códigos Convolucionais não Recursivos ou de Resposta ao Impulso Finita - FIR	16
3.3	Diagrama de Estados . . . . .	19
3.4	Diagrama de Treliça . . . . .	20
3.5	Códigos Convolucionais Recursivos Sistemáticos - RSC . . . . .	22
3.6	Algoritmo BCJR . . . . .	25
3.6.1	BCJR . . . . .	25
3.6.2	BCJR Modificado . . . . .	32
3.7	Código Convolucional Perfurado . . . . .	35
<b>4</b>	<b>Códigos Turbo</b>	<b>38</b>
4.1	Entrelaçador . . . . .	38
4.1.1	Entrelaçador de Bloco . . . . .	40
4.1.2	Entrelaçador de Berrou-Glavieux . . . . .	41

4.2	O Codificador . . . . .	41
4.3	O Decodificador . . . . .	43
<b>5</b>	<b>Códigos de Treliça Unicamente Decodificáveis para o Canal Aditivo com Dois</b>	
	<b>Usuários Binários</b>	<b>46</b>
5.1	Treliça para o 2-BAC . . . . .	46
5.2	Arranjos de Treliça e Subarranjos . . . . .	47
5.3	Código de Treliça para o 2-BAC . . . . .	50
<b>6</b>	<b>Decodificação Iterativa para o Canal Aditivo com Dois Usuários Binários</b>	
	<b>Usando Códigos de Treliça</b>	<b>55</b>
6.1	O Codificador . . . . .	55
6.2	O Decodificador . . . . .	56
	6.2.1 Algoritmo BCJR para Dois Usuários . . . . .	56
	6.2.2 Decodificação Iterativa . . . . .	62
<b>7</b>	<b>Decodificação Iterativa para o Canal Aditivo com Dois Usuários Binários</b>	
	<b>Usando CCMA</b>	<b>70</b>
7.1	O Codificador . . . . .	70
	7.1.1 Construção 1 . . . . .	71
	7.1.2 Construção 2 . . . . .	72
7.2	Decodificação Iterativa . . . . .	73
	7.2.1 Álgebra de Log-verossimilhança . . . . .	73
	7.2.2 Decodificador Turbo . . . . .	75
7.3	Exemplos Tutoriais . . . . .	77
<b>8</b>	<b>Codificação Hierárquica para a Ethernet</b>	<b>86</b>
8.1	Código Manchester . . . . .	86
8.2	Codificação Hierárquica . . . . .	86
<b>9</b>	<b>Decodibilidade Única para uma Classe de Códigos Transmitindo por meio</b>	
	<b>do 2-BAC Quase-Síncrono</b>	<b>91</b>
9.1	O Caso para $n = 2$ . . . . .	92
9.2	O Caso Geral . . . . .	93
9.3	Resultado das Simulações . . . . .	95

<b>10 Conclusões</b>	<b>97</b>
10.1 Sugestões para Trabalhos Futuros . . . . .	98
<b>Bibliografia</b>	<b>100</b>

# Lista de Figuras

1.1	Sistema de comunicação ponto-a-ponto, também chamado modelo clássico de um sistema de comunicações. Neste modelo há apenas um transmissor e um receptor.	3
2.1	Sistema de comunicação de acesso múltiplo com $T$ usuários. A saída do canal $f(\mathbf{x})$ é uma combinação dos sinais $\mathbf{x}_i$ enviados pelos remetentes, onde $i \in \{1, \dots, T\}$ . Quando o sinal recebido $f(\mathbf{x})$ é corrompido por ruído dá origem ao sinal $\hat{f}(\mathbf{x})$ .	7
2.2	Canal de acesso múltiplo com dois usuários. Os símbolos $x_i$ e $w_i$ são enviados pelo primeiro e segundo transmissor respectivamente. A saída do canal é uma função de combinação dos símbolos de entrada, sendo representada pelos símbolos $y_i$ .	8
2.3	Região de capacidade para o canal de acesso múltiplo com dois usuários.	9
2.4	Região de capacidade para o canal de acesso múltiplo com dois usuários quando $I(X; Y) = 0$ .	9
2.5	Região de Capacidade para o canal de acesso múltiplo com dois usuários quando $I(X; Y) = I(X; Y W)$ .	9
2.6	Canal aditivo com dois usuários binários.	10
2.7	Canal aditivo sem ruído com dois usuários binários.	10
2.8	Canal aditivo ruidoso com dois usuários binários.	10
2.9	A região hachurada é a região de taxas atingíveis com probabilidade de erro nula quando ambos os códigos constituintes são lineares	12
2.10	Taxas atingíveis quando um dos códigos constituintes possui $k_1$ posições contendo todas as $2^{k_1}$ -uplas.	13
3.1	Código FIR não sistemático.	16
3.2	Código RSC sistemático.	16
3.3	Código RSC não sistemático.	16
3.4	Código FIR não sistemático.	16

3.5	Diagrama de estados do codificador ilustrado na Figura 3.4, no qual os rótulos indicados nos ramos representam os pares entrada/saída. . . . .	20
3.6	Código FIR não sistemático, usado na construção da treliça da Figura 3.7. . . . .	21
3.7	Diagrama de treliça do codificador da Figura 3.6. Aqui os estados estão respresentados por 00, 01, 10, 11. Os rótulos indicados nos ramos representam as saídas do codificador. Os ramos pontilhadas e os ramos contínuos representam símbolos de entrada iguais a 1 e a 0 respectivamente. Os números em negrito abaixo da treliça representam os intervalos de tempo considerados ou a profundidade da treliça. . . . .	21
3.8	A seqüência de rótulos (11 10 01 10 00 01 11) do caminho destacado, corresponde à seqüência de sub-blocos resultante da informação (1 1 1 0 1). . . . .	22
3.9	Uma função de transferência racional da forma $\frac{f_0+f_1D+\dots+f_mD^m}{1+q_1D+\dots+q_mD^m}$ , pode ser implementada por meio deste codificador. . . . .	24
3.10	Treliça construída a partir da matriz geradora polinomial $G(D) = [1 \frac{1}{1+D}]$ . . . . .	30
3.11	Diagrama de treliça para o código perfurado gerado a partir do código com matriz geradora $G(D) = [1 + D^2 \quad 1 + D + D^2]$ . A cada quatro símbolos de saída um é perfurado (extraído), o codificador produz três símbolos na saída para cada dois símbolos de informação. O novo código portanto terá taxa assintótica 2/3. . . . .	35
3.12	codificador de um código com taxa 2/3. . . . .	36
3.13	Treliça resultante do codificador ilustrado na Figura 3.12. . . . .	36
4.1	Entrelaçador $I$ com seqüência de dados na entrada $\mathbf{u} = (u_1, u_2, u_3, \dots, u_N)$ e seqüência de dados na saída $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \dots, \tilde{u}_N)$ . . . . .	39
4.2	Função de mapeamento correspondente para entrelaçador com seqüência de entrada $\mathbf{u} = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8)$ e seqüência entrelaçada $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4, \tilde{u}_5, \tilde{u}_6, \tilde{u}_7, \tilde{u}_8) = (u_2, u_4, u_1, u_6, u_3, u_8, u_5, u_7)$ . . . . .	39
4.3	Diagrama de blocos de um codificador turbo com taxa 1/3. . . . .	42
4.4	Codificador turbo com taxa 1/3, utilizando dois códigos RSC idênticos com parâmetros (2, 1, 4). . . . .	43
4.5	O decodificador turbo consiste de dois decodificadores componentes concatenados em série. . . . .	44

5.1	Treliça para dois usuários em que, para cada usuário, é usado um mesmo código convolucional com matriz geradora $G(D) = \begin{bmatrix} 1 & \frac{1}{1+D} \end{bmatrix}$ . Os rótulos nos ramos $(u_k, d_k/\text{saída})$ correspondem respectivamente ao par de símbolos de informação do usuário 1 e usuário 2 e à saída do 2-BAC sem ruído. . . . .	47
5.2	Arranjo de treliça para o par de códigos com matriz geradora polinomial dada em (5.2) e seqüências binárias $a(s_i, s_p)$ e $b(s_r, s_v)$ constituídas por apenas $N = 1$ sub-bloco. . . . .	49
5.3	Arranjo de treliça para o par de códigos com matriz geradora polinomial dada em (5.3) e seqüências binárias $a(s_i, s_p)$ e $b(s_r, s_v)$ constituídas pela concatenação de $N = 2$ sub-blocos. . . . .	49
5.4	Modelo de construção de código unicamente decodificável para o 2-BAC. . . . .	51
5.5	Treliça resultante da concatenação de $\mathcal{C}$ e $\mathbf{C}_1$ (código do usuário 1). . . . .	52
5.6	Treliça resultante da concatenação de $\mathcal{C}$ e $\mathbf{C}_2$ (código do usuário 2). . . . .	52
5.7	Treliça resultante para dois usuários após o uso da concatenação serial ilustrada na Figura 5.4. . . . .	52
5.8	Arranjo de treliça perfurado, encontrado após eliminação de algumas linhas e colunas do arranjo de treliça ilustrado na Figura 5.2, devido ao uso da concatenação serial ilustrada em 5.4, tendo $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$ dado em (5.2) e $\mathbf{C}_1 = \{00, 11\}$ e $\mathbf{C}_2 = \{00, 01, 10\}$ . . . . .	53
5.9	Treliça para dois usuários após o uso do esquema de concatenação serial, tendo $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$ dado em (5.2) e $\mathbf{C}_1 = \{00, 11\}$ e $\mathbf{C}_2 = \{00, 01, 10\}$ . . . . .	54
5.10	Arranjo de treliça perfurado, encontrado após eliminação de algumas linhas e colunas do arranjo de treliça ilustrado na Figura 5.3, devido ao uso do esquema de concatenação serial (Figura 5.4), tendo $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$ dado em (5.3) e $\mathbf{C}_1 = \{00, 11\}$ e $\mathbf{C}_2 = \{00, 01, 10\}$ . . . . .	54
6.1	Esquema de codificação paralela para o codificador de $\mathcal{C}_1$ . . . . .	56
6.2	Esquema de codificação paralela para o codificador de $\mathcal{C}_2$ . . . . .	56
6.3	O decodificador empregado utiliza a decodificação iterativa para estimar a seqüência ternária mais provável e em seguida usa o decodificador 2-BAC para separar a informação relativa ao usuário 1 e ao usuário 2. . . . .	56

6.4	Decodificador turbo para dois usuários. O decodificador utiliza o princípio da decodificação iterativa e consiste de dois decodificadores componentes concatenados em série. É usado para estimar a seqüência ternária mais provável. . . . .	63
6.5	A- Usuário 1(1 iteração), B- Usuário 1(2 iterações), C- Usuário 1(3 iterações), D- Usuário 2 (1 iteração), E- Usuário 2 (2 iterações), F- Usuário 2 (3 iterações), G - Usuário 1 (Algoritmo BCJR), H - Usuário 2 (Algoritmo BCJR). . . . .	66
6.6	A- Usuário 1 (1 iteração), B- Usuário 1 (2 iterações), C- Usuário 1 (3 iterações), D- Usuário 2 (1 iteração), E- Usuário 2 (2 iterações), F- Usuário 2 (3 iterações), G - Usuário 1 (Algoritmo BCJR), H - Usuário 2 (Algoritmo BCJR). . . . .	67
6.7	Curvas relacionadas ao usuário 1. Os casos para os quais $\mathcal{C}_1^- = \mathcal{C}_1^  = \mathcal{C}_2^- = \mathcal{C}_2^ $ têm matrizes geradoras polinomiais $G(D) = \left[1 \quad \frac{1+D+D^2}{1+D^2}\right]$ estão ilustrados em: A- 1 iteração, B- 2 iterações e C- 3 iterações; Os casos para os quais $\mathcal{C}_1^- = \mathcal{C}_1^  = \mathcal{C}_2^- = \mathcal{C}_2^ $ têm matrizes geradoras polinomiais $G(D) = \left[1 \quad \frac{1+D+D^3}{1+D^2+D^3}\right]$ estão ilustrados em: D- 1 iteração, E- 2 iterações e F- 3 iterações; O uso do algoritmo BCJR, sem a decodificação iterativa está ilustrado em: G - matriz geradora do codificador convolucional é $G(D) = \left[1 \quad \frac{1+D+D^2}{1+D^2}\right]$ , H- matriz geradora do codificador convolucional é $G(D) = \left[1 \quad \frac{1+D+D^3}{1+D^2+D^3}\right]$ . . . . .	68
6.8	Curvas relacionadas ao usuário 2. Os casos para os quais $\mathcal{C}_1^- = \mathcal{C}_1^  = \mathcal{C}_2^- = \mathcal{C}_2^ $ têm matrizes geradoras polinomiais $G(D) = \left[1 \quad \frac{1+D+D^2}{1+D^2}\right]$ estão ilustrados em: A- 1 iteração, B- 2 iterações e C- 3 iterações; Os casos para os quais $\mathcal{C}_1^- = \mathcal{C}_1^  = \mathcal{C}_2^- = \mathcal{C}_2^ $ têm matrizes geradoras polinomiais $G(D) = \left[1 \quad \frac{1+D+D^3}{1+D^2+D^3}\right]$ estão ilustrados em: D- 1 iteração, E- 2 iterações e F- 3 iterações; O uso do algoritmo BCJR, sem a decodificação iterativa está ilustrado em: G - matriz geradora do codificador convolucional é $G(D) = \left[1 \quad \frac{1+D+D^2}{1+D^2}\right]$ , H- matriz geradora do codificador convolucional é $G(D) = \left[1 \quad \frac{1+D+D^3}{1+D^2+D^3}\right]$ . . . . .	69
7.1	Arranjo representando palavra código de $\mathcal{C}_1$ . . . . .	72
7.2	Arranjo representando palavra código de $\mathcal{C}_2$ . . . . .	72
7.3	Arranjo representando palavra código de $\mathcal{C}_T$ . . . . .	72
7.4	Esquema de concatenação serial empregando um par de códigos de bloco unicamente decodificáveis e um par de códigos produto. . . . .	72
7.5	Arranjo representando palavra código do código ternário $\mathcal{C}_T^*$ . . . . .	73

7.6	Esquema de concatenação serial empregando um par de códigos de bloco unicamente decodificáveis para o 2-BAC e um código de bloco ternário. . . . .	73
7.7	O decodificador com entrada suave/saída suave usa os valores a priori $\Lambda_1(x_k)$ e $\Lambda_2(x_k)$ para todos os símbolos de informação, se disponível, e os valores do canal $\frac{4r+4}{2\sigma^2}$ e $\frac{8r}{2\sigma^2}$ para todos os símbolos codificados. Ele também entrega as saídas suaves $\Lambda_1(\hat{x}_k)$ e $\Lambda_2(\hat{x}_k)$ para todos os símbolos de informação e as informações extrínsecas $\Lambda_{1e}(\hat{x}_k)$ e $\Lambda_{2e}(\hat{x}_k)$ . . . . .	75
7.8	Palavra código do código produto binário $\mathcal{C}_1$ . . . . .	78
7.9	Palavra código do código produto binário $\mathcal{C}_2$ . . . . .	78
7.10	Palavra código do código ternário $\mathcal{C}_T$ . . . . .	78
7.11	Palavra código para o código binário $\mathcal{C}_1$ . . . . .	79
7.12	Palavra código para o código binário $\mathcal{C}_2$ . . . . .	79
7.13	Palavra código para o código ternário $\mathcal{C}_T$ . . . . .	79
7.14	Valores recebidos $\frac{8r}{2\sigma^2}$ . . . . .	79
7.15	Valores recebidos $\frac{4r+4}{2\sigma^2}$ . . . . .	79
7.16	Informação extrínseca $\Lambda_{1e}^-$ após a primeira decodificação horizontal. . . . .	80
7.17	Informação extrínseca $\Lambda_{2e}^-$ após a primeira decodificação horizontal. . . . .	80
7.18	Informação extrínseca $\Lambda_{1e}^ $ após a primeira decodificação vertical. . . . .	80
7.19	Informação extrínseca $\Lambda_{2e}^ $ após a primeira decodificação vertical. . . . .	80
7.20	Saída suave $\Lambda_1(\hat{x})$ após a primeira decodificação horizontal e vertical. . . . .	81
7.21	Saída suave $\Lambda_2(\hat{x})$ após a primeira decodificação horizontal e vertical. . . . .	81
7.22	Probabilidades relativas ao símbolo -2. . . . .	81
7.23	Probabilidades relativas ao símbolo 2. . . . .	81
7.24	Probabilidades relativas ao símbolo 0. . . . .	81
7.25	Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação do desempenho alcançado após uma e duas iterações para ambos usuários usando a construção 1. . . . .	81
7.26	Palavra código para o código ternário $\mathcal{C}_T^*$ . . . . .	82
7.27	Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação do desempenho alcançado após uma e duas iterações para ambos usuários usando a construção 2. . . . .	84

7.28	Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação de desempenho alcançado usando a construção 1 e a construção 2 para o usuário 1. . . . .	84
7.29	Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação de desempenho alcançado usando a construção 1 e a construção 2 para o usuário 2. . . . .	85
8.1	Inclusão de novos usuários em redes <i>Ethernet</i> já existentes. A especificação original utiliza o código Manchester. . . . .	87
8.2	Esquema de codificação hierárquica com dois usuários. . . . .	88
8.3	A - Usuário 1 (codificação hierárquica), B- Usuário 2 (codificação hierárquica), C- Código Manchester, D- PAM ternário. . . . .	89
8.4	Concatenação do esquema de codificação hierárquica com código turbo. . . . .	90
8.5	Diagrama de blocos representando o codificador turbo com taxa 1/3. . . . .	90
8.6	A - Usuário 1 (codificação hierárquica), B - Usuário 1 com turbo (1 iteração), C - Usuário 1 com turbo (2 iterações), D- Usuário 1 com turbo (3 iterações) E - Usuário 2 (codificação hierárquica), F - Usuário 2 com turbo (1 iteração), G - Usuário 2 com turbo (2 iterações), H- Usuário 2 com turbo (3 iterações). . . . .	90
9.1	2-BAC quase-síncrono. A diferença de fase $s$ entre duas palavras-código de dois códigos de comprimento de bloco $n$ é chamada de <i>slippage</i> . . . . .	92
9.2	2-BAC quase-síncrono para código com comprimento de bloco 2. A diferença de fase entre duas palavras-código dos dois códigos é $s = 1$ . O par $(v_k, v_{k+1})$ representa uma palavra código do usuário 1. Os pares $(w_{k-1}, w_k)$ e $(w_{k+1}, w_{k+2})$ representam duas palavras-código do usuário 2. . . . .	93
9.3	Todos os possíveis vetores recebidos $(r_k, r_{k+1})$ no decodificador e as palavras código correspondentes do codificador 1 $(v_k, v_{k+1})$ e os pares correspondentes $(w_k, w_{k+1})$ do codificador 2. . . . .	94
9.4	Código de treliça para $C_2$ com sub-blocos de comprimento 2. . . . .	94
9.5	Código de treliça para $C_2$ com comprimento de bloco 3. . . . .	95

9.6	2-BAC síncrono $\times$ 2-BAC quase-síncrono. Curvas da probabilidade de erro versus relação sinal ruído para comparação do caso em que $C_1 = \{00, 11\}$ e $C_2 = \{00, 01, 10\}$ para um 2-BAC quase-síncrono e o caso em que $C_1 = \{00, 11\}$ e o codificador para $C_2$ tem a estrutura de treliça mostrada na Figura 9.4 para um 2-BAC síncrono e um 2-BAC quase- síncrono. . . . .	96
10.1	Esquema de concatenação serial para uso em canal aditivo com $T$ usuários binários.	99

# Lista de Tabelas

2.1	Saídas possíveis do 2-BAC quando $C_1 = \{00, 11\}$ e $C_2 = \{00, 01, 10\}$ . . . . .	11
3.1	Tabela contendo os parâmetros $\alpha, \beta, \gamma$ e $\Lambda$ resultantes da aplicação do algoritmo BCJR em um sistema codificado por um RSC (2,1,1) com matriz geradora polinomial $G(D) = [1 \frac{1}{1+D}]$ . . . . .	31
3.2	Tabela contendo os parâmetros $\alpha, \beta, \gamma$ e $\Lambda$ resultantes da aplicação do algoritmo BCJR modificado em um sistema codificado por um RSC (2,1,1) com matriz geradora polinomial $G(D) = [1 \frac{1}{1+D}]$ . . . . .	34
4.1	Bloco $3 \times 3$ , usado para exemplificar os tipos de entrelaçadores de bloco. . . . .	40
8.1	Par de códigos unicamente decodificáveis para o 2-BAC. . . . .	87
8.2	Mapeamento do código 3B2T no código <i>pseudo Manchester</i> . . . . .	88
8.3	Par de códigos unicamente decodificáveis para o 2-BAC com taxa 1,25. . . . .	88
9.1	$C_1 = \{00, 11\}$ e $C_2 = \{00, 01, 10\}$ é um par de códigos de bloco unicamente decodificáveis para o 2-BAC síncrono. . . . .	92

# Lista de Abreviaturas Utilizadas

<b>ICC</b>	Conferência Internacional de Comunicações
<b>BCJR</b>	Algoritmo de Bahl, Cocke, Jelinek e Raviv
<b>2-BAC</b>	Canal Aditivo com Dois Usuários Binários
<b>CCMA</b>	Codificação Colaborativa para Acesso Múltiplo
<b>T-BAC</b>	Canal Aditivo com $T$ Usuários Binários
<b>BCE</b>	Codificador Convolutacional Binário
<b>BCC</b>	Código Convolutacional Binário
<b>FIR</b>	Resposta ao Impulso Finita
<b>IIR</b>	Resposta ao Impulso Infinita
<b>RSC</b>	Código Convolutacional Recursivo Sistemático
<b>BPSK</b>	Modulação Binária por Deslocamento em Fase
<b>SNR</b>	Relação Sinal Ruído
<b>MAP</b>	Máximo <i>a Posteriori</i>
<b>MC</b>	Código Manchester
<b>3B2T</b>	Código 3 Binário 2 Ternário
<b>3PAM</b>	Modulação de Amplitude de Pulso Ternário
<b>QSUD</b>	Quase Síncrono Unicamente Decodificável
<b>CDMA</b>	Acesso Múltiplo por Divisão de Canal
<b>COFDM</b>	Multiplexação por Divisão em Frequência Ortogonal

# Capítulo 1

## Motivação e Plano de Tese

### 1.1 Introdução

Em 1948 Claude Shannon publicou um artigo intitulado “A Mathematical Theory of Communication” [1]. Neste artigo, foi desenvolvida uma teoria matemática que estabeleceu os limites e as possibilidades para o modelo do sistema de comunicações mostrado na Figura 1.1. Os sistemas representados por este modelo, chamados de sistemas ponto-a-ponto, compreendem apenas um transmissor e um receptor, que trocam informações por meio de um canal ruidoso, isto é, um canal que pode modificar aleatoriamente as mensagens transmitidas. Para este modelo, Shannon mostrou que, apesar da presença de ruído, o canal permitia a transmissão de seqüências de símbolos com uma probabilidade de erro tão pequena quanto se quisesse, desde que se escolhesse o conjunto de seqüências convenientemente e que a taxa de transmissão não ultrapassasse um determinado valor, dependente apenas do canal, denominado por ele de capacidade do canal. Desde então, um grande esforço por parte dos pesquisadores tem sido empreendido para implementar métodos eficientes de codificação e decodificação a fim de garantir desempenho em ambientes ruidosos. Após numerosos trabalhos na tentativa de aproximação ao limite de Shannon, duas grandes famílias de códigos emergiram. Os códigos de bloco [2] [3] e os códigos convolucionais [2]-[6].

Apesar dos sistemas ponto-a-ponto abrangerem um número grande de situações reais, existem casos que não são tratáveis com este modelo. Pode-se citar o caso em que vários usuários desejam se comunicar com um mesmo destinatário fazendo uso simultâneo de um mesmo canal de comunicações. Exemplos de acesso múltiplo são a transmissão de vários canais de voz por meio de um único cabo coaxial em uma rede telefônica, a transmissão de vários programas de

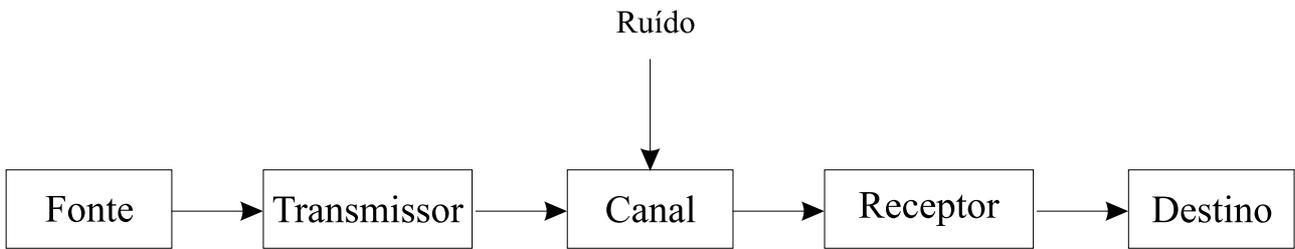


Figura 1.1: Sistema de comunicação ponto-a-ponto, também chamado modelo clássico de um sistema de comunicações. Neste modelo há apenas um transmissor e um receptor.

televisão via satélite e a troca de mensagens e informações entre um computador central e seus vários terminais em uma rede de computadores. Sistemas de comunicações de acesso múltiplo foram estudados primeiramente por Shannon em 1961 [7] e desde então têm sido investigados por diversos autores [8]-[32].

Para tirar melhor proveito das técnicas de codificação de informação, um esquema de codificação concatenada foi proposto por Forney [34]. Esse método foi proposto com o objetivo de construção de códigos longos a partir de códigos curtos, tendo o decodificador, para esses códigos longos, menor complexidade do que o teria sem a concatenação. O mais popular desses esquemas consiste em um código de Reed-Solomon seguido por um código convolucional [35].

A decodificação de códigos convolucionais é geralmente feita com o uso do algoritmo de Viterbi [36] [37], que é um método de decodificação de máxima verossimilhança [2] que minimiza a probabilidade de erro por grupo de sub-blocos. Em 1989, Hagenauer apresentou um artigo intitulado “A Viterbi algorithm with soft-decision outputs” [38] no qual ele procurou modificar o algoritmo de Viterbi com a finalidade de estimar não apenas a seqüência correspondente ao caminho mais provável em uma máquina de estados finitos, mas também a probabilidade *a posteriori* para cada bit ou um valor de confiabilidade desta estimativa.

Códigos turbo foram apresentados à comunidade científica em 1993 por Berrou, Glaviex e Thitimajshima na Conferência Internacional de Comunicações (ICC), em Genebra, por meio de um artigo intitulado “Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes” [39]. Neste artigo eles mostraram que uma combinação de concatenações paralelas e decodificação iterativa, com uso de entrelaçadores, pode prover comunicações confiáveis, com o desempenho, em termos de correção de erros, próximo ao limite de Shannon. Para isto, os entrelaçadores devem ter comprimentos suficientemente grandes. A descoberta de *turbo codes* reanimou alguns conceitos e algoritmos adormecidos, como por exemplo o algoritmo proposto por Bahl *et al* [40] (BCJR), e os combinou com várias idéias novas.

Esta tese de doutorado em engenharia elétrica na área de comunicações utiliza *turbo codes* para uso em sistemas de comunicações que utilizam canal de acesso múltiplo, isto é, meio de transmissão no qual mais de um usuário pode acessá-lo simultaneamente, com a saída deste canal sendo uma combinação dos sinais enviados pelos usuários ativos. Em particular será dada ênfase ao caso em que dois usuários binários podem transmitir simultaneamente em um canal aditivo para um único receptor. Este canal é chamado de canal aditivo com dois usuários binários (2-BAC)[7]-[9] [20]-[28].

Foram feitas implementações de sistemas codificados para simulação de códigos e decodificadores em presença de ruído branco gaussiano e estabelecidas condições para decodibilidade única para códigos de treliça adaptados ao canal 2-BAC. Foram desenvolvidos métodos de decodificação iterativa para o 2-BAC usando códigos de treliça bem como esquemas CCMA (codificação colaborativa para acesso múltiplo). Propõe-se também uma aplicação de códigos para o 2-BAC para melhoria do desempenho da rede Ethernet e foi estabelecida uma condição para decodibilidade única sobre uma classe de códigos para o 2-BAC quase-síncrono. Todas as implementações foram feitas através de simulações com a utilização da ferramenta MATLAB 6.5 [33].

Esta tese de doutorado foi desenvolvida no Departamento de Eletrônica e Sistemas da Universidade Federal de Pernambuco, tendo a autora viajado à Inglaterra para cumprir um período de quatro meses, a fim de concluir simulações de decodificação iterativa, bem como desenvolver o projeto intitulado *Hierarchical Coding for Enhanced Performance ETHERNET*, em conjunto com o grupo de pesquisas do Prof. Garik Markarian do *Institute of Integrated Information Systems* na *University of Leeds* - United Kingdom.

## 1.2 Plano de Tese

Este trabalho está organizado em 10 capítulos, incluindo as conclusões. O Capítulo 1 é introdutório, com uma revisão bibliográfica e uma exposição do plano de tese.

No Capítulo 2 é revisado o conceito de canais de acesso múltiplo dando ênfase ao canal aditivo com dois usuários binários (2-BAC).

No capítulo 3 introduz-se a notação a ser utilizada com uma revisão dos conceitos relacionados a códigos convolucionais como os códigos convolucionais recursivos sistemáticos, diagramas de estados e treliça, perfuradores e algoritmo de decodificação BCJR.

O Capítulo 4 contém uma revisão dos conceitos relacionados a códigos turbo e entrelaçadores.

No Capítulo 5 é revisa-se o conceito de treliça para dois usuários. Introduce-se uma condição para decodibilidade única em códigos de treliça no 2-BAC e descreve-se um novo esquema de codificação para obtenção de tais códigos com o uso da concatenação serial de um par de códigos de bloco unicamente decodificáveis para o 2-BAC com um par de códigos convolucionais sistemáticos. Este esquema de codificação permite decodificação eficiente no 2-BAC.

No Capítulo 6 é introduzida a decodificação iterativa para o 2-BAC com o uso do algoritmo BCJR para dois usuários e de códigos de treliça para dois usuários. Resultados de simulações são apresentados em forma de curvas para ilustração da melhoria de desempenho com o uso da decodificação iterativa.

No Capítulo 7 um novo esquema de codificação CCMA é descrito, que permite o uso da decodificação iterativa para o 2-BAC. Resultados de simulações são mostrados para comparação do desempenho das duas construções de códigos apresentadas.

No Capítulo 8 propõe-se um esquema de codificação que permite um acréscimo do tráfego total em redes *Ethernet*, mantendo compatibilidade com os usuários já existentes.

No Capítulo 9 uma condição de decodibilidade única para uma classe de códigos  $(C_1, C_2)$ , em que  $C_1 = \{0^n, 1^n\}$ , transmitindo por meio de um 2-BAC quase-síncrono, é introduzida e finalmente o Capítulo 10 encerra o trabalho com as conclusões e perspectivas de trabalhos futuros.

# Capítulo 2

## Canais de Acesso Múltiplo

A origem da Teoria da Informação data de 1948 quando Claude E. Shannon publicou um artigo no *Bell System Technical Journal*, que ele intitulou “A Mathematical Theory of Communication” [1]. Desde então, muitos outros pesquisadores dedicaram-se a estudar e expandir esta teoria. O modelo clássico proposto por Shannon representa sistemas de comunicação ponto-a-ponto, isto é, sistemas em que há apenas um remetente enviando informações por intermédio de um canal ruidoso para um único destinatário. Existem situações, entretanto, em que vários remetentes desejam se comunicar com um único destinatário, por meio de um mesmo canal. É o caso, por exemplo, de várias estações de rádio na Terra, querendo se comunicar com um único satélite. Este sistema de comunicações é conhecido como sistema de acesso múltiplo.

O esquema de acesso múltiplo de captura de canal, tem como proposta principal reduzir o tempo de transmissão. Nesta abordagem, o remetente pode usar o canal em sua plena capacidade, no instante em que desejar, como se este fosse exclusivamente seu. É o caso do sistema ALOHA, projetado nos anos 70, por Abramson [42] na Universidade do Hawaii. Abramson confrontou-se com o problema de conectar vários terminais de computadores, espalhados pelas diversas ilhas do arquipélago, ao computador central da Universidade. A solução encontrada por Abramson, chamada de ALOHA PURO [42] [43], consistia em estabelecer que os terminais dividissem as mensagens em blocos de duração fixa, denominados pacotes, e que os transmitissem assim que ficassem disponíveis. Se durante o intervalo de tempo de transmissão de um pacote de um usuário nenhum outro usuário utilizasse o canal, então o receptor seria capaz de receber corretamente o pacote, avisando ao transmissor de que assim foi feito. Entretanto, se um segundo usuário tentasse enviar um pacote enquanto o primeiro estivesse transmitindo, o receptor detectaria esta colisão e desprezaria os dois (ou mais) pacotes que colidiram, avisando

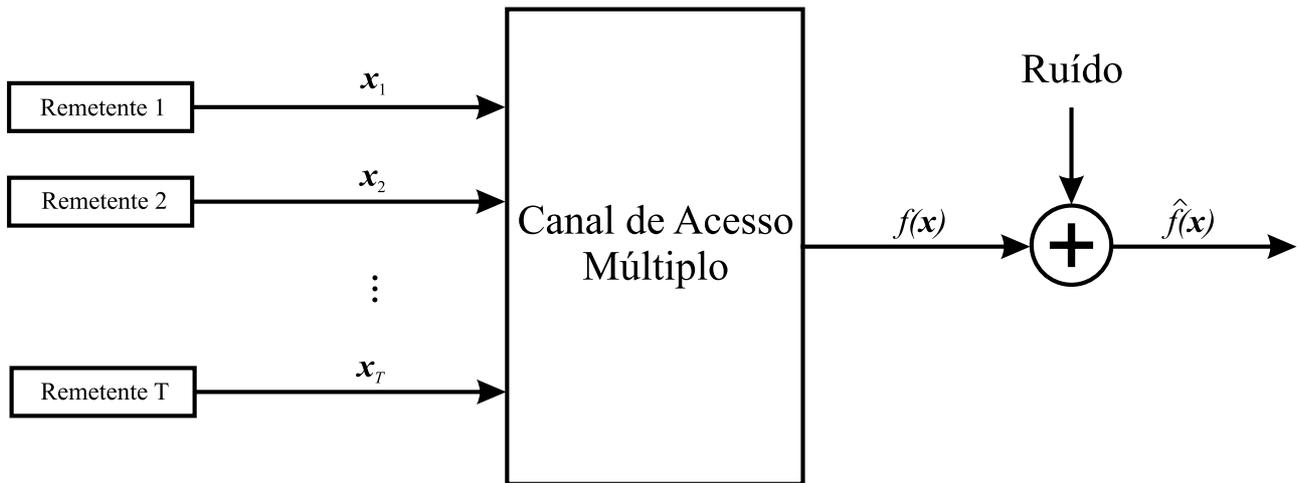


Figura 2.1: Sistema de comunicação de acesso múltiplo com  $T$  usuários. A saída do canal  $f(\mathbf{x})$  é uma combinação dos sinais  $\mathbf{x}_i$  enviados pelos remetentes, onde  $i \in \{1, \dots, T\}$ . Quando o sinal recebido  $f(\mathbf{x})$  é corrompido por ruído dá origem ao sinal  $\hat{f}(\mathbf{x})$ .

a todos os terminais de que uma colisão havia ocorrido. Os terminais afetados pela colisão deveriam esperar um tempo, escolhido alatoriamente, para retransmitir os seus pacotes.

Pode-se citar também como esquema de acesso múltiplo de captura de canal, o algoritmo de pilha projetado por Capetanakis [44], então estudante de doutorado juntamente com o Prof. R. Gallager e, independentemente, pelos pesquisadores soviéticos B. Tsybakov e V. Mikhailov [45].

Nos anos 60, pesquisadores [46]-[48] apresentaram o modelo matemático de um sistema de transmissão no qual mais de um remetente pode acessá-lo simultaneamente. Neste modelo, a saída do canal  $f(\mathbf{x})$  é uma combinação dos sinais  $\mathbf{x}_i$  enviados pelos remetentes, em que  $i \in \{1, \dots, T\}$ . Esse modelo de transmissão é chamado, em geral, de canal de acesso múltiplo com  $T$  usuários e pode-se visualizá-lo na Figura 2.1. Quando o sinal recebido  $f(\mathbf{x})$  é corrompido por ruído dá origem ao sinal  $\hat{f}(\mathbf{x})$ . A função de combinação de maior interesse ao longo desta tese é a adição sobre os reais. Este modelo é chamado de canal aditivo com  $T$  usuários binários (T-BAC). Será dada ênfase ao caso em que há apenas dois usuários (2-BAC).

## 2.1 Teoria da Informação para Canais de Acesso Múltiplo

Pode-se definir formalmente o canal de acesso múltiplo [8]-[14] como um modelo matemático de transmissão no qual mais de um remetente pode acessá-lo simultaneamente, com a saída deste canal sendo uma combinação dos sinais enviados pelos remetentes como ilustrado na Figura 2.1.

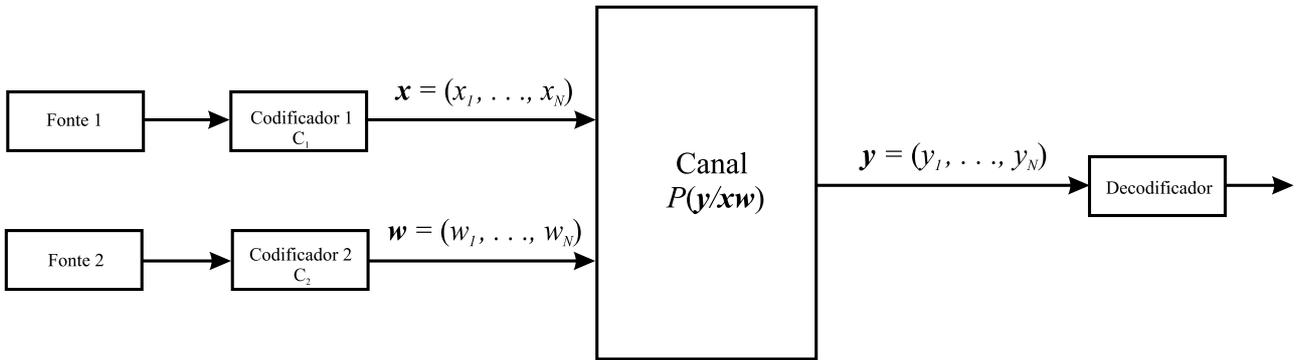


Figura 2.2: Canal de acesso múltiplo com dois usuários. Os símbolos  $x_i$  e  $w_i$  são enviados pelo primeiro e segundo transmissor respectivamente. A saída do canal é uma função de combinação dos símbolos de entrada, sendo representada pelos símbolos  $y_i$ .

Um determinado tipo de canal que será de grande interesse neste trabalho é o canal de acesso múltiplo com dois usuários [7]-[9] [20]-[28], ilustrado na Figura 2.2. Neste modelo de transmissão, existem duas fontes independentes, discretas, codificadas independentemente em duas entradas para o canal.

Representa-se por  $(C_1, C_2)$  um par de códigos de bloco, em que cada código tem comprimento  $N$ . O código constituinte  $C_1$ , usado para o transmissor 1, tem  $M$  palavras  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$  e o código constituinte  $C_2$ , usado para o transmissor 2, tem  $L$  palavras  $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_L\}$ . Em cada unidade de tempo, o primeiro transmissor envia símbolos  $x$  pertencentes a um determinado alfabeto  $X$  e o segundo transmissor envia símbolos  $w$  pertencentes a um determinado alfabeto  $W$ . A saída do canal, é uma determinada função de combinação dos símbolos de entrada, representada pelos símbolos  $y$  pertencentes a um determinado alfabeto de saída  $Y$ . O canal é sem memória, sendo que  $\mathbf{x} = (x_1, \dots, x_N)$  e  $\mathbf{w} = (w_1, \dots, w_N)$  representam as entradas no canal dos remetentes 1 e 2 respectivamente e  $\mathbf{y} = (y_1, \dots, y_N)$  representa a saída do canal em  $N$  instantes de tempo sucessivos. Sendo o canal de tempo discreto e sem memória, com entradas e saídas pertencentes a alfabetos também discretos, a probabilidade condicional  $P\{\mathbf{y}|\mathbf{x}\mathbf{w}\}$  representando a matriz de transição do canal é

$$P\{\mathbf{y}|\mathbf{x}\mathbf{w}\} = \prod_{i=1}^N P\{y_i|x_iw_i\}.$$

A taxa conjunta  $R$  do par  $(C_1, C_2)$  é dada por:

$$R = R_1 + R_2 = \frac{\log(M)}{N} + \frac{\log(L)}{N}.$$

Um importante resultado acerca destes canais é o teorema provado por Ahlswede [11] e Liao [12] [13], enunciado a seguir, cuja demonstração pode ser vista em [14].

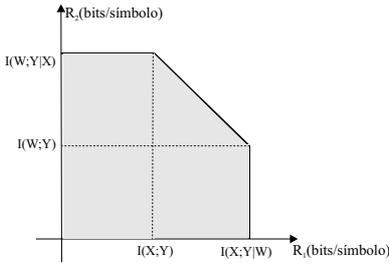


Figura 2.3: Região de capacidade para o canal de acesso múltiplo com dois usuários.

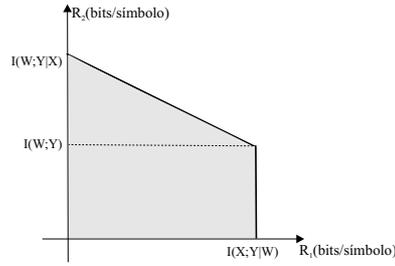


Figura 2.4: Região de capacidade para o canal de acesso múltiplo com dois usuários quando  $I(X;Y) = 0$ .

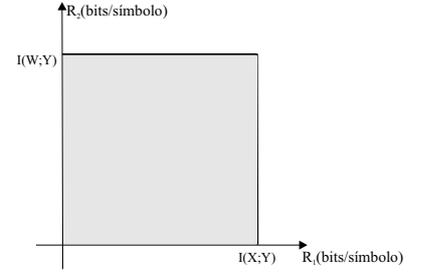


Figura 2.5: Região de Capacidade para o canal de acesso múltiplo com dois usuários quando  $I(X;Y) = I(X;Y|W)$ .

**Teorema 2.1.1** *As taxas de transmissão possíveis  $R$  para o tipo de canal representado acima formam a região convexa sobre o conjunto dos pares das taxas de transmissão  $(R_1, R_2)$  satisfazendo as seguintes desigualdades:*

$$\begin{aligned} R_1 + R_2 &\leq I(XW; Y), \\ 0 \leq R_1 &\leq I(X; Y|W), \\ 0 \leq R_2 &\leq I(W; Y|X), \end{aligned}$$

em que  $I(A; B)$  denota a informação mútua e  $I(A; B|C)$  denota a informação mútua entre  $A$  e  $B$  condicionada à variável aleatória  $C$ .

As restrições impostas no Teorema acima são ilustradas por meio dos gráficos das Figuras 2.3, 2.4 e 2.5.

## 2.2 Canal Aditivo com Dois Usuários Binários

O canal aditivo com dois usuários binários (2-BAC) [8]-[10] [20]-[28] está ilustrado na Figura 2.6. Neste sistema, dois remetentes geograficamente separados tentam enviar dados binários por um mesmo canal de comunicações. A função de combinação utilizada neste canal é a soma sobre os reais. Desta forma, cada remetente tem como alfabeto de entrada o conjunto  $F_2 = \{0, 1\}$ . A entrada do canal consiste de duplas binárias  $x_i, w_i$  e a saída  $y_i = x_i + w_i$ , será formada por símbolos do alfabeto  $\{0, 1, 2\}$ . O remetente 1 envia palavras-código de um código de bloco  $C_1$  e o remetente 2 envia palavras-código de um código de bloco  $C_2$ . Considera-se que:

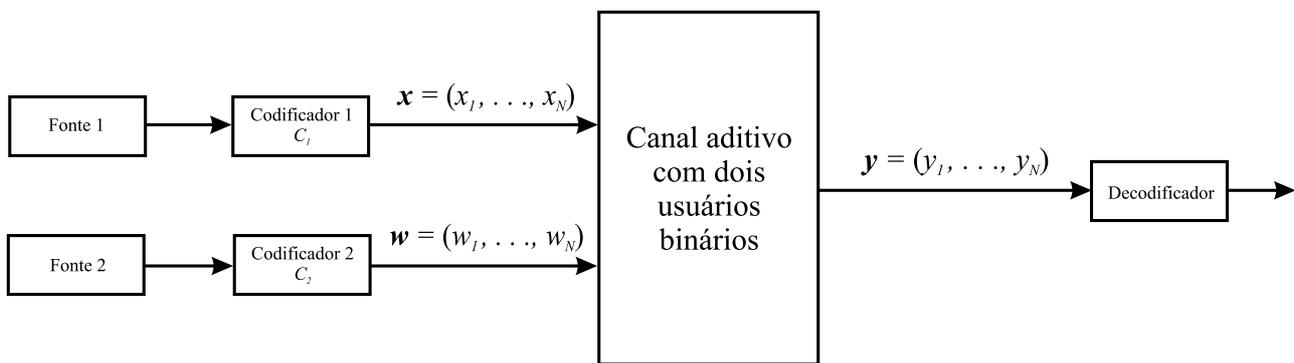


Figura 2.6: Canal aditivo com dois usuários binários.

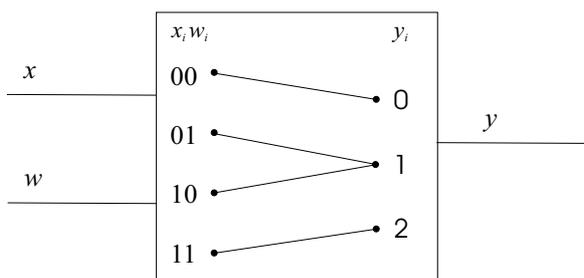


Figura 2.7: Canal aditivo sem ruído com dois usuários binários.

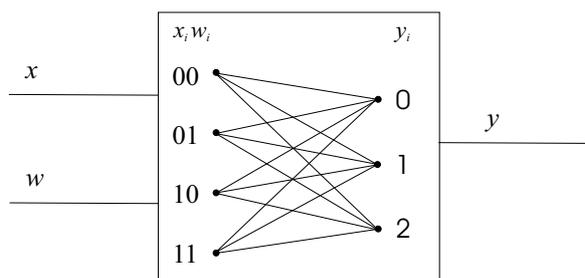


Figura 2.8: Canal aditivo ruidoso com dois usuários binários.

- Os dois remetentes operam na mesma banda de frequência;
- Os dois remetentes transmitem ao mesmo tempo;
- Os dois remetentes utilizam códigos binários de mesmo comprimento  $N$ ;
- É mantido sincronismo na transmissão dos símbolos das palavras;
- Existe um único decodificador no receptor;
- Os dois usuários escolhem independentemente as respectivas palavras-código a serem transmitidas.

Os dois códigos  $C_1$ ,  $C_2$  juntos são denominados de códigos colaborativos para dois usuários [29]-[32], e cada componente é denominado de código constituinte.

A seguir verifica-se os dois modelos de canais mostrados nas Figuras 2.7 e 2.8. O modelo de canal da Figura 2.7 é chamado canal aditivo sem ruído com dois usuários binários. Neste modelo, a saída do canal é a soma aritmética dos símbolos de entrada, assim se os dois símbolos transmitidos pelos dois remetentes são 0's, um 0 é transmitido pelo canal para o receptor. Se

Tabela 2.1: Saídas possíveis do 2-BAC quando  $C_1 = \{00, 11\}$  e  $C_2 = \{00, 01, 10\}$ .

$C_2 \downarrow \setminus C_1 \rightarrow$	00	11
00	00	11
01	01	12
10	10	21

os dois símbolos transmitidos pelos dois remetentes são 1's, um 2 é transmitido pelo canal para o receptor, se o símbolo transmitido por um remetente é 0 e pelo outro remetente é 1, um 1 é transmitido pelo canal para o receptor. A região de capacidade para este tipo de canal foi estudada na seção anterior e está representada na Figura 2.3. O modelo de canal da Figura 2.8 é chamado de canal aditivo ruidoso com dois usuários binários, neste tipo de canal todas as transições são possíveis e existe uma lei de probabilidade relacionando os símbolos de entrada com o de saída  $P\{y|x_i, w_i\}$ .

### 2.2.1 Decodibilidade Única

O interesse inicial é construir um par de códigos  $C_1$  e  $C_2$ , para o canal aditivo com dois usuários binários, de modo que o decodificador seja capaz de decodificar o vetor  $\mathbf{y}$  recebido, sem ambigüidade nas duas palavras-código que foram transmitidas pelos remetentes 1 e 2. Isto é, se para quaisquer  $x_1, x_2 \in C_1$  e  $w_1, w_2 \in C_2$  tais que  $x_1 \neq x_2$  e  $w_1 \neq w_2$  então  $x_1 + w_1 \neq x_2 + w_2$ . Um par de códigos  $(C_1, C_2)$  que possui esta propriedade é dito ser unicamente decodificável. Há o interesse também em que as taxas de transmissão  $(R_1, R_2)$  de  $C_1$  e  $C_2$  respectivamente estejam em um ponto dentro da região de capacidade e tão próximas à fronteira quanto possível.

**Exemplo 2.2.1** Para  $N = 2$ , é possível construir um par unicamente decodificável com  $C_1 = \{00, 11\}$  e  $C_2 = \{00, 01, 10\}$ . A taxa de transmissão para  $C_1$  e  $C_2$  é  $(\frac{\log 2}{2}, \frac{\log 3}{2})$ , que é um ponto dentro da região de capacidade (Figura 2.3). As possíveis saídas para o 2-BAC sem ruído são vistas na Tabela 2.1, na qual a primeira linha representa as palavras-código de  $C_1$ , a primeira coluna representa as palavras-código de  $C_2$  e as saídas são as somas aritméticas, símbolo a símbolo, das linhas com as colunas. Observando a Tabela 2.1 verificamos que o vetor  $\mathbf{y}$  pode ser decodificado sem ambigüidade em duas palavras-código, uma em  $C_1$  e a outra em  $C_2$ .

Este código, de grande interesse para o trabalho, pode ser generalizado para  $N > 2$ , simplesmente tomando  $C_1$  como  $\{0^N, 1^N\}$ , em que  $0^N$  e  $1^N$  representam as  $N$ -uplas binárias com

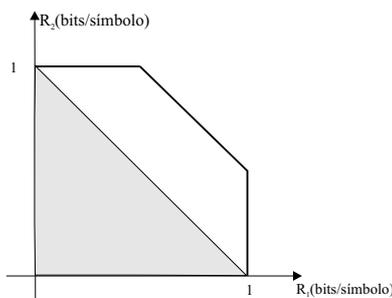


Figura 2.9: A região hachurada é a região de taxas atingíveis com probabilidade de erro nula quando ambos os códigos constituintes são lineares .

todos os elementos iguais a 0 e 1, respectivamente, e  $C_2$  como o conjunto de todas as  $N$ -uplas binárias com exceção de  $1^N$ . As taxas de transmissão de  $C_1$  e  $C_2$  são respectivamente,

$$R_1 = \frac{\log 2}{N}, \quad (2.1)$$

$$R_2 = \frac{\log(2^N - 1)}{N}, \quad (2.2)$$

portanto a taxa conjunta para este código será:

$$R = R_1 + R_2 = \frac{\log(2^N - 1) + 1}{N}, \quad (2.3)$$

que tende para 1, quando  $N \rightarrow \infty$ .

## 2.2.2 Códigos Lineares para o 2-BAC

Pode-se analisar o comportamento das taxas de transmissão quando ambos os códigos constituintes  $C_1$  e  $C_2$  são lineares ou quando apenas um dos dois códigos constituintes é linear. Quando  $C_1$  e  $C_2$  forem códigos binários lineares com parâmetros  $(N, k_1)$  e  $(N, k_2)$  respectivamente, para que  $(C_1, C_2)$  seja unicamente decodificável é preciso que eles não tenham mais de uma palavra-código em comum. Para que isto ocorra tem-se que  $k_1 + k_2 \leq N$  [8] e portanto

$$R = R_1 + R_2 = \frac{\log M_1 + \log M_2}{N} = \frac{k_1 + k_2}{N} \leq 1,$$

em que  $M_i$  é o número de palavras-código de  $C_i$ . Desta forma vê-se que a taxa de transmissão conjunta  $R$  não poderá ultrapassar 1, como mostra o gráfico da Figura 2.9.

Outro caso de interesse é aquele para o qual apenas um dos códigos constituintes é linear.

**Definição 2.2.1** *Um código  $(C_1, C_2)$  de comprimento  $N$  para o 2-BAC é dito ser um código linear se um dos códigos constituintes for um código linear de parâmetros  $(N, k)$ .*

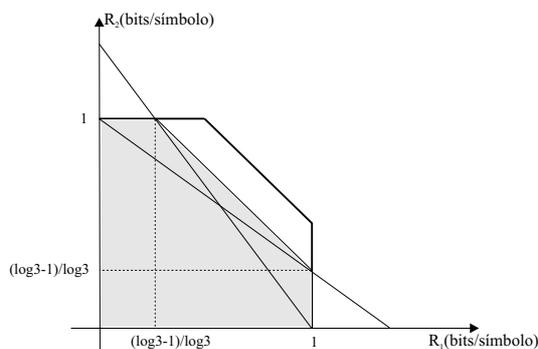


Figura 2.10: Taxas atingíveis quando um dos códigos constituintes possui  $k_1$  posições contendo todas as  $2^{k_1}$ -uplas.

Códigos lineares para o 2-BAC possuem a desvantagem de não atingirem a capacidade quando o código constituinte linear tiver taxa menor que  $\frac{\log 3 - 1}{\log 3}$ , como decorre do Teorema abaixo provado por Weldon [8] [19].

**Teorema 2.2.1** *Se  $C_1$  é um código constituinte linear com parâmetros  $(N, k_1)$ , então a região de capacidade para um par  $(C_1, C_2)$  unicamente decodificável é limitada superiormente por*

$$(R_1, R_2) \leq \left( \frac{k_1}{N}, \left(1 - \frac{k_1}{N}\right) \log 3 \right).$$

Rocha [20] observou que na demonstração do teorema 2.2.1 não é feito uso da linearidade de  $C_1$  e, portanto, o resultado é válido quando  $C_1$  é apenas um código que possua  $k_1$  posições contendo todas as  $2^{k_1}$ -uplas. Assim ele pode ser sistemático e não linear. Pode-se ilustrar esse Teorema com o auxílio do gráfico da Figura 2.10.

### 2.2.3 Algumas Construções de Códigos para o 2-BAC

Pode-se citar diversos trabalhos cuja finalidade é a construção de códigos para o 2-BAC, por exemplo [16] [17] [21] [22]. Kasami e Lin [16] apresentaram um método para construção de pares de códigos  $\delta$ -decodificáveis. Em publicação posterior Kasami e Lin [17] apresentaram um esquema para decodificação de códigos  $\delta$ -decodificáveis para o 2-BAC com ruído, levando em conta a linearidade e corrigindo no máximo  $\lfloor (\delta - 1)/2 \rfloor$  erros de transmissão, em que  $\lfloor (\delta - 1)/2 \rfloor$  denota o maior inteiro igual ou menor que  $(\delta - 1)/2$ . Ahswede e Balakirsky [22], apresentaram um método de construção de códigos binários unicamente decodificáveis  $(C_1, C_2)$  para o 2-BAC, de comprimento  $tN$ , em que  $t$  e  $N$  são inteiros fixos, e nem  $C_1$  nem  $C_2$  são lineares. Rocha e Massey [21] estabeleceram condições suficientes para determinação de códigos binários

unicamente decodificáveis  $(C_1, C_2)$  de peso constante para o 2-BAC, com ruído e sem ruído, obtendo a decodibilidade única.

Em 1994, Cabral [8] em sua dissertação de mestrado, apresentou um algoritmo para, a partir de um código  $C_1$  linear, produzir o código  $C_2$  com o maior número possível de palavras-código. Em outras palavras, ele verificou que condições dois conjuntos de vetores binários devem satisfazer para garantir sua decodibilidade única sobre o 2-BAC, dando ênfase ao caso em que um dos conjuntos é um código de bloco linear. Cabral provou teoremas que permitem dividir a busca de vetores para  $C_2$  no espaço  $F_2^N$  em buscas em subconjuntos de menor cardinalidade e independentes entre si. Em 1999, Alcoforado [9] deu continuidade ao trabalho acima mencionado e, em particular, implementou o algoritmo de Cabral para obtenção de códigos unicamente decodificáveis no 2-BAC.

# Capítulo 3

## Códigos Convolucionais

Códigos convolucionais foram introduzidos na literatura por Elias [49] e desde então muitos pesquisadores dedicaram-se ao estudo da sua estrutura algébrica, como é o caso de Forney [50]. Neste capítulo apresentamos alguns conceitos de códigos convolucionais [2] [4] [5] [6] necessários para implementação dos códigos turbo.

### 3.1 Conceitos Básicos

Um codificador convolucional binário (BCE), é um dispositivo cuja saída, formada por blocos de  $n$  símbolos binários, não depende só dos blocos de  $k$  símbolos binários presentes na entrada, mas também dos  $m$  blocos de mensagens anteriores. Por ter memória, o codificador convolucional de parâmetros  $(n, k, m)$  pode ser implementado como um circuito sequencial lógico com  $k$  entradas,  $n$  saídas e memória  $m$ . Normalmente,  $n$  e  $k$  são valores inteiros positivos, pequenos, com  $k < n$ . Quanto maior o valor da memória  $m$ , usualmente obtém-se uma menor probabilidade de erro após a decodificação. O valor  $k/n$  é chamado de taxa assintótica do código. Um código convolucional binário (BCC) é o conjunto de todas as palavras código passíveis de serem produzidas à saída do codificador convolucional binário.

As Figuras 3.1, 3.2 e 3.3 mostram vários tipos de BCE's. Um BCE pode apresentar resposta ao impulso finita (FIR), também chamado de não recursivo, ou apresentar resposta ao impulso infinita (IIR), são os códigos conhecidos como recursivos ou com realimentação (RSC). O BCE também pode ser sistemático ou não sistemático. O BCE é sistemático se  $k$  símbolos presentes em cada bloco de saída de comprimento  $n$  são efetivamente iguais aos  $k$  símbolos de cada bloco de entrada.

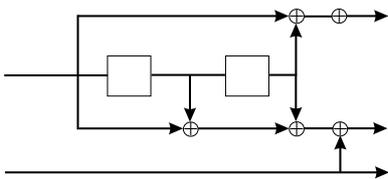


Figura 3.1: Código FIR não sistemático.

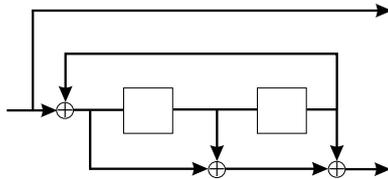


Figura 3.2: Código RSC sistemático.

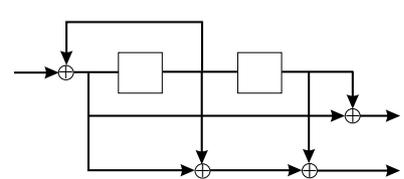


Figura 3.3: Código RSC não sistemático.

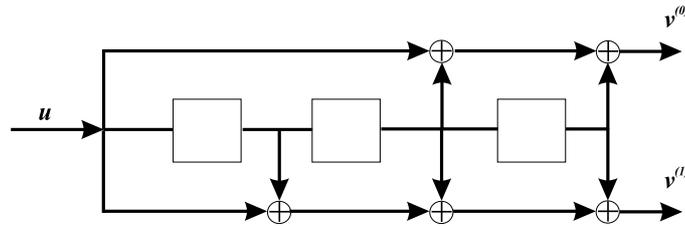


Figura 3.4: Código FIR não sistemático.

## 3.2 Códigos Convolucionais não Recursivos ou de Resposta ao Impulso Finita - FIR

Um codificador é dito ser FIR, se e somente se, a sua saída pode ser calculada como uma combinação linear das entradas presentes e de um número finito de entradas passadas. A combinação linear é expressa em termos de símbolos de entrada e das seqüências geradoras para os codificadores. Vamos analisar como exemplo o codificador ilustrado na Figura 3.4.

Este codificador tem parâmetros  $(2, 1, 3)$  e taxa assintótica  $1/2$ . A entrada consiste de blocos de 1 símbolo e a saída de blocos de 2 símbolos. A seqüência de informação  $\mathbf{u} = (u_0, u_1, u_2, \dots)$  entra bit a bit no codificador. Como o codificador é um sistema linear, as duas seqüências de saída do codificador  $\mathbf{v}^{(0)} = (v_0^{(0)}, v_1^{(0)}, v_2^{(0)}, \dots)$  e  $\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \dots)$  podem ser obtidas pela convolução da seqüência de entrada  $\mathbf{u}$  com as duas respostas ao impulso do codificador. As respostas ao impulso são obtidas quando a entrada é  $\mathbf{u} = (100\dots)$  e observam-se as duas seqüências obtidas na saída. Como o codificador tem memória  $m$ , as respostas ao impulso são  $\mathbf{g}^{(0)} = (g_0^{(0)}, g_1^{(0)}, g_2^{(0)}, \dots, g_m^{(0)})$  e  $\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, g_2^{(1)}, \dots, g_m^{(1)})$ . Para o codificador da Figura 3.4 tem-se:

$$\mathbf{g}^{(0)} = (1011),$$

$$\mathbf{g}^{(1)} = (1111),$$

as respostas ao impulso são chamadas de seqüências geradoras do código.

As equações de codificação podem ser escritas como:

$$\mathbf{v}^{(0)} = \mathbf{u} * \mathbf{g}^{(0)},$$

$$\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)},$$

em que  $*$  denota a convolução discreta e todas as operações são modulo-2. A convolução implica que para todo  $l \geq 0$ ,

$$v_l^{(j)} = \sum_{i=0}^m u_{l-i} g_i^{(j)} = u_l g_0^{(j)} + u_{l-1} g_1^{(j)} + \dots + u_{l-m} g_m^{(j)}, \quad j = 0, 1,$$

em que  $u_{l-i} = 0$  para todo  $l < i$ . Portanto, para o codificador da Figura 3.4 tem-se:

$$v_l^{(0)} = u_l + u_{l-2} + u_{l-3},$$

$$v_l^{(1)} = u_l + u_{l-1} + u_{l-2} + u_{l-3},$$

que podem ser obtidos de imediato por inspeção direta no circuito de codificação. Depois da codificação, as duas seqüências de saída são multiplexadas em uma única seqüência, chamada de palavra-código, para a transmissão no canal. A palavra-código é:

$$\mathbf{v} = (v_0^{(0)} v_0^{(1)}, v_1^{(0)} v_1^{(1)}, v_2^{(0)} v_2^{(1)}, \dots).$$

Em qualquer sistema linear, operações no domínio do tempo envolvendo convoluções podem ser substituídas por operações polinomiais. Como um codificador convolucional é um sistema linear, cada seqüência nas equações de codificação pode ser substituída por uma correspondência linear e a operação de convolução substituída por uma multiplicação polinomial. Por exemplo, para um código  $(2, 1, m)$  as equações de codificação podem se expressas em função do operador de deslocamento  $D$ .

$$\mathbf{v}^{(0)}(D) = \mathbf{u}(D) \mathbf{g}^{(0)}(D),$$

$$\mathbf{v}^{(1)}(D) = \mathbf{u}(D) \mathbf{g}^{(1)}(D),$$

em que

$$\mathbf{u}(D) = u_0 + u_1 D + u_2 D^2 + \dots,$$

é a seqüência de informação,

$$\mathbf{v}^{(0)}(D) = v_0^{(0)} + v_1^{(0)} D + v_2^{(0)} D^2 + \dots,$$

$$\mathbf{v}^{(1)}(D) = v_0^{(1)} + v_1^{(1)} D + v_2^{(1)} D^2 + \dots,$$

são as seqüências codificadas e

$$\begin{aligned}\mathbf{g}^{(0)}(D) &= g_0^{(0)} + g_1^{(0)}D + g_2^{(0)}D^2 + \dots + g_m^{(0)}D^m, \\ \mathbf{g}^{(1)}(D) &= g_0^{(1)} + g_1^{(1)}D + g_2^{(1)}D^2 + \dots + g_m^{(1)}D^m,\end{aligned}$$

são os polinômios geradores do código.

Como o codificador é um sistema linear e  $\mathbf{u}^{(i-1)}(D)$  representa a  $i$ -ésima seqüência de entrada e  $\mathbf{v}^{(j-1)}(D)$  representa a  $j$ -ésima seqüência de saída, o polinômio gerador pode ser interpretado como a função de transferência relacionando a entrada  $i$  e a saída  $j$ . Considerando um sistema com  $k$  entradas e  $n$  saídas, existe um total de  $k.n$  funções de transferência, que podem ser representadas por uma matriz de funções de transferência também chamada matriz geradora polinomial  $\mathbf{G}(D)$ .

$$\mathbf{G}(D) = \begin{bmatrix} \mathbf{g}_0^{(0)}(D) & \mathbf{g}_0^{(1)}(D) & \dots & \mathbf{g}_0^{(n-1)}(D) \\ \mathbf{g}_1^{(0)}(D) & \mathbf{g}_1^{(1)}(D) & \dots & \mathbf{g}_1^{(n-1)}(D) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{g}_{k-1}^{(0)}(D) & \mathbf{g}_{k-1}^{(1)}(D) & \dots & \mathbf{g}_{k-1}^{(n-1)}(D) \end{bmatrix}, \quad (3.1)$$

em que  $\mathbf{g}_{i-1}^{(j-1)}(D)$  é o polinômio gerador relacionando a  $i$ -ésima entrada e a  $j$ -ésima saída.

Usando a matriz geradora polinomial, as equações de codificação para um código  $(n, k, m)$  podem ser expressas como:

$$\mathbf{V}(D) = \mathbf{U}(D)\mathbf{G}(D), \quad (3.2)$$

em que  $\mathbf{U}(D) = [\mathbf{u}^{(0)}(D), \mathbf{u}^{(1)}(D), \dots, \mathbf{u}^{(k-1)}(D)]$  é uma  $k$ -upla de seqüências de entrada e  $\mathbf{V}(D) = [\mathbf{v}^{(0)}(D), \mathbf{v}^{(1)}(D), \dots, \mathbf{v}^{(n-1)}(D)]$ .

Para os codificadores FIR das Figuras 3.1 e 3.4 tem-se

$$\begin{aligned}\mathbf{G}(D) &= \begin{bmatrix} 1 & 1 & 1 \\ 1 + D^2 & 1 + D + D^2 & 0 \end{bmatrix}, \\ \mathbf{G}(D) &= [1 + D^2 + D^3 \quad 1 + D + D^2 + D^3],\end{aligned}$$

respectivamente.

### 3.3 Diagrama de Estados

Considere um codificador convolucional  $(n, k, m)$  com matriz geradora polinomial dada em (3.1). O comprimento de restrição para a  $i$ -ésima entrada de uma matriz geradora polinomial é definido como

$$\nu_i = \text{máx}_{(0 \leq j \leq n-1)} \{\text{grau } \mathbf{g}_i^{(j)}\} \quad 0 \leq i \leq k-1.$$

A memória  $m$  da matriz geradora polinomial é definida como o comprimento de restrição máximo, isto é

$$m = \text{máx}_{(0 \leq i \leq k-1)} \{\nu_i\}.$$

O comprimento de restrição total é definido como a soma dos comprimentos de restrição para todas as  $i$ -ésimas entradas

$$\nu = \sum_{i=0}^{k-1} \nu_i.$$

Em outras palavras, queremos dizer que, para um código  $(n, k, m)$  com  $k > 1$ , o  $i$ -ésimo registrador de deslocamento contém  $\nu_i$  símbolos de informação anteriores. Isto significa que os elementos do vetor  $(\nu_1, \nu_2, \dots, \nu_k)$  representam os comprimentos de cada um dos  $k$  registradores de deslocamento do codificador, isto é, o  $i$ -ésimo registrador de deslocamento tem  $\nu_i$  elementos de memória. O estado do codificador no instante de tempo  $l$ , quando as entradas do codificador são

$$u_l^{(0)} u_l^{(1)} \dots u_l^{(k-1)},$$

é a  $m$ -upla binária

$$(u_{l-1}^{(0)} u_{l-2}^{(0)} \dots u_{l-m_1}^{(0)} u_{l-1}^{(1)} u_{l-2}^{(1)} \dots u_{l-m_2}^{(1)} \dots u_{l-1}^{(k-1)} u_{l-2}^{(k-1)} \dots u_{l-m_k}^{(k-1)}).$$

Existe um total de  $2^\nu$  estados. Para um código  $(n, 1, m)$  tem-se que  $\nu = \nu_1$  e o estado do codificador no instante de tempo  $l$  é:

$$(u_{l-1} u_{l-2} \dots u_{l-m}).$$

O diagrama de estados é um grafo que consiste de nós, representando os estados do codificador, e de linhas com setas representando as transições entre os estados. Cada linha é rotulada por um par de entrada/saída. Dado o estado atual do codificador, a seqüência de informação na entrada determina o caminho pelo diagrama de estados e a seqüência de saída.

Cada novo bloco de  $k$  entradas leva a uma transição para um novo estado. Portanto existem  $2^k$  ramos saindo de cada estado, correspondendo a cada bloco de entrada diferente. Para um

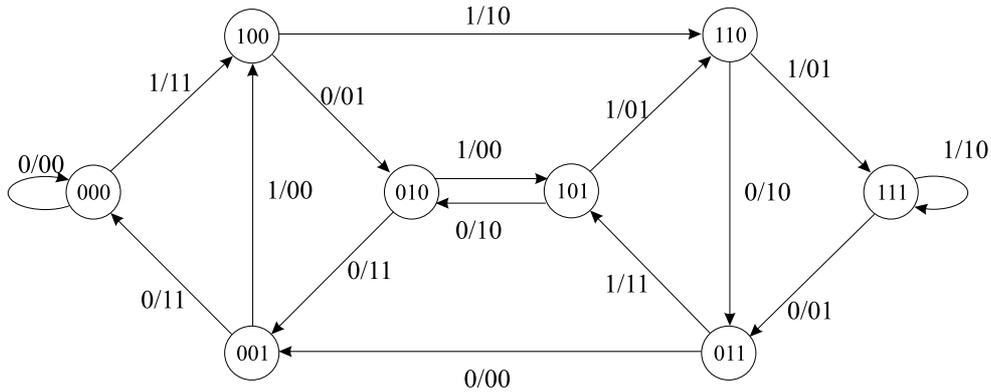


Figura 3.5: Diagrama de estados do codificador ilustrado na Figura 3.4, no qual os rótulos indicados nos ramos representam os pares entrada/saída.

código  $(n, 1, m)$  existem somente dois ramos saindo de cada estado. Cada ramo é rotulado por um par de entrada/saída.

O estado atual e a saída do codificador são unicamente determinados pelo estado anterior e a entrada atual. O codificador realiza uma transição de um estado para outro quando um bloco de mensagens é deslocado no codificador.

Considere por exemplo, o codificador mostrado na Figura 3.4. O diagrama de estados para este codificador é mostrado na Figura 3.5. O codificador tem oito estados rotulados como 000, 001, 010, 011, 100, 101, 110 e 111, em que cada estado representa o conteúdo do registrador de deslocamento. Existem dois possíveis ramos saindo de cada estado, correspondendo a dois possíveis valores do símbolo de mensagem.

### 3.4 Diagrama de Treliça

Um diagrama de treliça é obtido a partir de um diagrama de estados traçando todas as possíveis transições de estados e as respectivas seqüências de entrada/saída no decorrer do tempo.

Como um exemplo, considere o diagrama de treliça (Figura 3.7) do código ilustrado na Figura 3.6 com matriz geradora

$$\mathbf{G}(D) = [1 + D^2 \quad 1 + D + D^2], \tag{3.3}$$

e uma seqüência de informação de comprimento  $L = 5$ . O diagrama de treliça contém  $L + m + 1$  unidades de tempo ou níveis e estes são rotulados de 0 a  $L + m$  na Figura 3.7. Considerando que o estado inicial do codificador é 00 e que o estado final é também 00, as primeiras  $m$  unidades de tempo correspondem à partida do codificador do estado 00 e as últimas  $m$  unidades de tempo

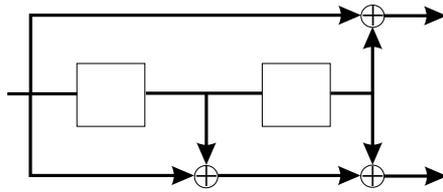


Figura 3.6: Código FIR não sistemático, usado na construção da treliça da Figura 3.7.

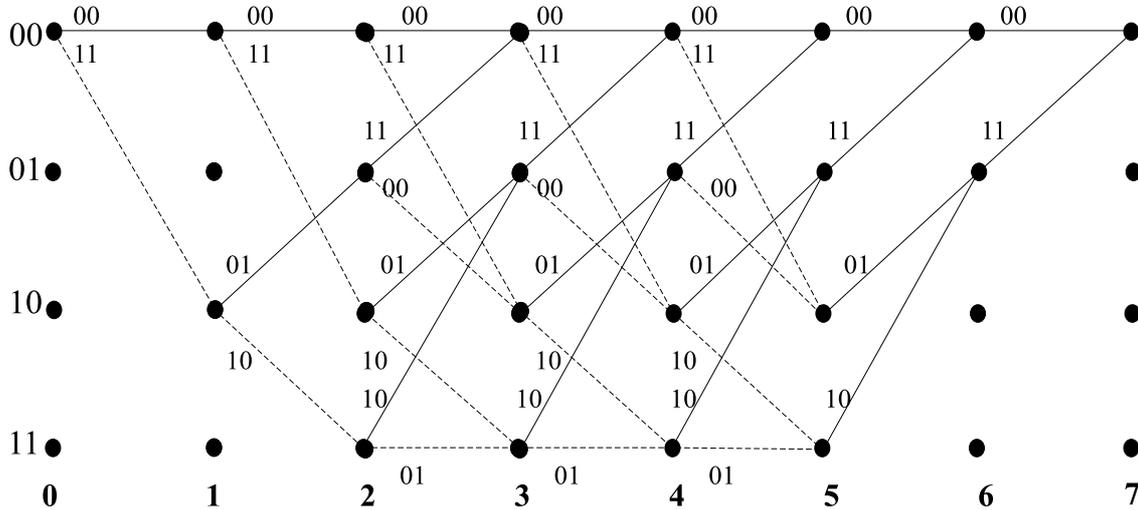


Figura 3.7: Diagrama de treliça do codificador da Figura 3.6. Aqui os estados estão respresentados por 00, 01, 10, 11. Os rótulos indicados nos ramos representam as saídas do codificador. Os ramos pontilhadas e os ramos contínuos representam símbolos de entrada iguais a 1 e a 0 respectivamente. Os números em negrito abaixo da treliça representam os intervalos de tempo considerados ou a profundidade da treliça.

correspondem ao retorno do codificador ao estado 00. Isto significa que nem todos os estados podem ser alcançados nas primeiras  $m$  ou nas últimas  $m$  unidades de tempo. Contudo, na porção central da treliça, todos os estados são possíveis, e cada unidade de tempo contém uma réplica do diagrama de estados. Existem dois ramos saindo e entrando em cada estado. Os ramos pontilhados correspondem à entrada do símbolo de informação igual a 1 e os ramos contínuos correspondem à entrada de um símbolo de informação igual a 0. Cada ramo é rotulado com as saídas  $v_i$  correspondentes e cada uma das  $2^L$  palavras-código de comprimento  $N = n(L + m)$  é representada por um único caminho na treliça. Por exemplo, a palavra-código correspondendo à seqüência de informação  $\mathbf{u} = (11101)$ , tem seu caminho destacado na Figura 3.8. Considerando o caso geral de um código  $(n, k, m)$  e uma seqüência de informação de comprimento  $kL$ , existem  $2^k$  ramos saindo e entrando em cada estado e  $2^{kL}$  caminhos distintos na treliça correspondendo a  $2^{kL}$  palavras-código.

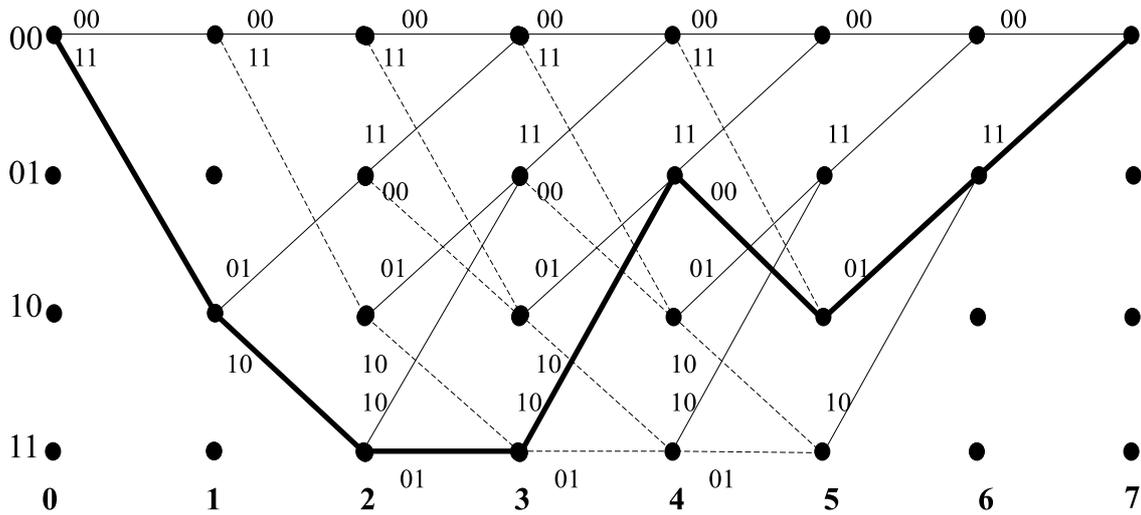


Figura 3.8: A seqüência de rótulos (11 10 01 10 00 01 11) do caminho destacado, corresponde à seqüência de sub-blocos resultante da informação (1 1 1 0 1).

### 3.5 Códigos Convolucionais Recursivos Sistemáticos - RSC

Os códigos convolucionais recursivos sistemáticos (RSC) [4]-[6] [52], também chamados de códigos de resposta ao impulso infinita (IIR), podem ser obtidos a partir de códigos com resposta ao impulso finita (FIR) como mostrado nesta seção.

Em um código convolucional sistemático  $(n, k, m)$  as primeiras  $k$  seqüências de saída são réplicas exatas das seqüências de informação de entrada. A matriz geradora polinomial de um código convolucional sistemático tem a forma

$$\mathbf{G}(D) = [\mathbf{I} \quad \mathbf{P}(D)],$$

em que  $\mathbf{I}$  é uma matriz identidade  $k \times k$  e  $\mathbf{P}(D)$  é uma matriz  $k \times (n - k)$ .

Para obter a forma sistemática para uma determinada matriz geradora polinomial de um código convolucional é necessário primeiro definir matriz geradora polinomial equivalente. Duas matrizes geradoras polinomiais são equivalentes se elas geram o mesmo código convolucional.

**Exemplo 3.5.1** Considere a matriz geradora polinomial para o codificador da Figura 3.6.

$$\mathbf{G}(D) = [1 + D^2 \quad 1 + D + D^2],$$

a seqüência código deste codificador é dada por

$$\mathbf{V}(D) = \mathbf{U}(D)\mathbf{G}(D), \quad (3.4)$$

$$\mathbf{V}(D) = \mathbf{U}(D) \begin{bmatrix} 1 + D^2 & 1 + D + D^2 \end{bmatrix}, \quad (3.5)$$

$$\mathbf{V}(D) = \mathbf{U}(D)(1 + D^2) \begin{bmatrix} 1 & \frac{1 + D + D^2}{1 + D^2} \end{bmatrix}, \quad (3.6)$$

$$\mathbf{V}(D) = \mathbf{U}'(D) \begin{bmatrix} 1 & \frac{1 + D + D^2}{1 + D^2} \end{bmatrix}, \quad (3.7)$$

$$\mathbf{V}(D) = \mathbf{U}'(D)\mathbf{G}_1(D), \quad (3.8)$$

em que

$$\mathbf{G}_1(D) = \begin{bmatrix} 1 & \frac{1 + D + D^2}{1 + D^2} \end{bmatrix},$$

$$\mathbf{U}'(D) = \mathbf{U}(D)\mathbf{T}(D),$$

e

$$\mathbf{T}(D) = (1 + D^2).$$

O conjunto de seqüências de saída  $\mathbf{V}(D)$  é produzido pelo produto do conjunto de seqüências  $\mathbf{U}'(D)$  pela matriz geradora  $\mathbf{G}_1(D)$ , bem como pelo produto do conjunto de seqüências  $\mathbf{U}(D)$ , pela matriz geradora original  $\mathbf{G}(D)$  em que

$$\mathbf{G}(D) = \mathbf{T}(D)\mathbf{G}_1(D).$$

Diz-se que duas matrizes geradoras  $\mathbf{G}(D) = \mathbf{T}(D)\mathbf{G}_1(D)$  e  $\mathbf{G}_1(D)$  são equivalentes se a matriz  $\mathbf{T}(D)$  é inversível. A matriz  $\mathbf{G}_1(D)$  no exemplo anterior está na forma sistemática. Contudo, as suas entradas não são polinômios, mas funções racionais. A seqüência de paridade na saída de (3.6) é obtida pela multiplicação da seqüência de entrada pelo polinômio  $(1 + D + D^2)$  e divisão pelo polinômio  $(1 + D^2)$ . As operações de multiplicação podem ser representadas por um registrador de deslocamento sem realimentação e a divisão por um registrador de deslocamento com realimentação. Isto é, no caso do codificador RSC, os elementos da matriz geradora são funções racionais na variável  $D$  com coeficientes binários, em outras palavras é uma razão de polinômios com coeficientes binários.

Uma função de transferência racional da forma

$$\frac{f_0 + f_1D + \dots + f_mD^m}{1 + q_1D + \dots + q_mD^m},$$

pode ser implementada pelo codificador mostrado na Figura 3.9 [5]. A saída é uma função linear da entrada e do conteúdo dos registradores de deslocamento. A entrada do registrador

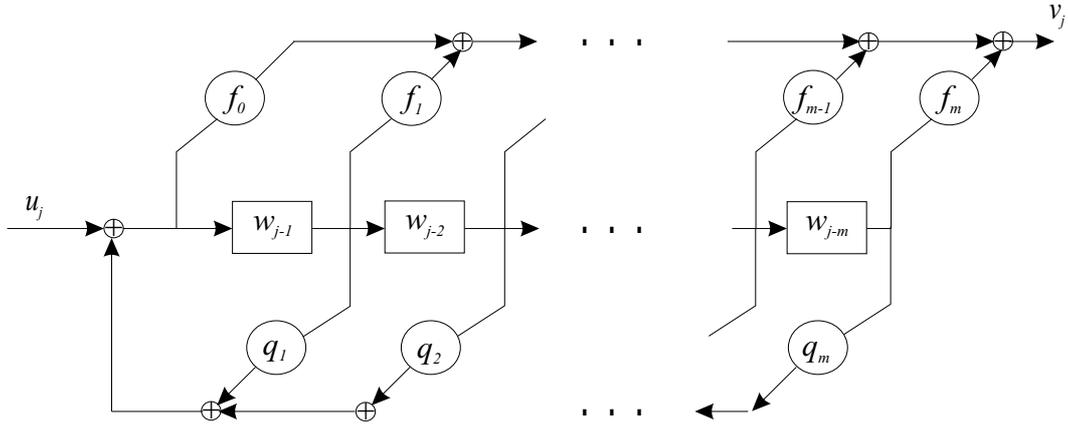


Figura 3.9: Uma função de transferência racional da forma  $\frac{f_0+f_1D+\dots+f_mD^m}{1+q_1D+\dots+q_mD^m}$ , pode ser implementada por meio deste codificador.

de deslocamento é uma função linear da entrada e do conteúdo do registrador de deslocamento. No instante de tempo  $j$  a saída será:

$$v_j = \sum_{i=0}^m f_i w_{j-i}.$$

Utilizando a representação polinomial tem-se

$$v(D) = \sum_{j=-\infty}^{\infty} v_j D^j = \sum_{j=-\infty}^{\infty} \sum_{i=0}^m f_i w_{j-i} D^j \quad (3.9)$$

$$= \sum_{k=-\infty}^{\infty} \left( \sum_{i=0}^m f_i D^i \right) w_k D^k = f(D)w(D), \quad (3.10)$$

em que substituiu-se  $j - i$  por  $k$ , no qual

$$f(D) = f_0 + f_1 D + \dots + f_m D^m,$$

e

$$w(D) = \sum_{k=-\infty}^{\infty} w_k D^k.$$

Da Figura 3.9 segue

$$w_j = u_j + \sum_{i=1}^m q_i w_{j-i}.$$

Supondo  $q_0 = 1$  pode-se escrever

$$u_j = \sum_{i=0}^m q_i w_{j-i},$$

ou repetindo os passos de (3.10)

$$u(D) = q(D)w(D), \quad (3.11)$$

em que

$$u(D) = \sum_{j=-\infty}^{\infty} u_j D^j,$$

e

$$q(D) = 1 + q_1 D + \dots + q_m D^m.$$

Combinando (3.10) e (3.11) tem-se

$$v(D) = u(D) \frac{f(D)}{q(D)} = u(D) \frac{f_0 + f_1 D + \dots + f_m D^m}{1 + q_1 D + \dots + q_m D^m}. \quad (3.12)$$

Seja

$$g(D) = \frac{f(D)}{q(D)},$$

então

$$v(D) = u(D)g(D).$$

Diz-se que  $g(D)$  é uma função de transferência racional. Em geral a matriz  $G(D)$  cujas entradas são funções racionais é chamada de matriz de funções de transferência racionais. As matrizes de funções de transferência racionais relativas aos codificadores das Figuras 3.2 e 3.3 são respectivamente,

$$\begin{aligned} \mathbf{G}(D) &= \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix}, \\ \mathbf{G}(D) &= \begin{bmatrix} \frac{1+D^2}{1+D} & \frac{1+D+D^2}{1+D} \end{bmatrix}. \end{aligned}$$

## 3.6 Algoritmo BCJR

O algoritmo de Viterbi [2] [3] [5] é um método de decodificação ótimo, no sentido que minimiza a probabilidade de erro por bloco de dígitos para códigos convolucionais. Este algoritmo porém não minimiza necessariamente a probabilidade de erro por símbolo. Um algoritmo utilizado para esta finalidade foi proposto em 1974 por Bahl *et al* [40] e é geralmente chamado de BCJR. Este algoritmo foi modificado para ser utilizado em codificadores convolucionais recursivos [39]. Esta seção aborda o algoritmo BCJR [40] e o algoritmo BCJR modificado [39] [52] [55].

### 3.6.1 BCJR

Sem perda de generalidade, considere um codificador convolucional sistemático com taxa de transmissão assintótica  $1/2$  e  $M$  estados, tem-se que a seqüência de símbolos de informação é representada por

$$\mathbf{u} = \mathbf{u}_1^N = \{u_1, u_2, \dots, u_k, \dots, u_N\}.$$

A seqüência código associada é

$$\mathbf{v} = \mathbf{v}_1^N = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k, \dots, \mathbf{v}_N\},$$

em que  $\mathbf{v}_k = (v_k^{(1)}, v_k^{(2)}) = (u_k, v_k^{(2)})$ , é a saída associada a cada símbolo de informação e  $\mathbf{v}_1^N$  é a entrada para um canal contaminado com ruído branco gaussiano, sem memória, cuja saída é a seqüência recebida,

$$\mathbf{r} = \mathbf{r}_1^N = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k, \dots, \mathbf{r}_N\},$$

em que  $\mathbf{r}_k = (x_k, y_k)$ . As variáveis aleatórias  $x_k$  e  $y_k$  no instante de tempo  $k$ , são definidas pelas seguintes relações:

$$x_k = (2u_k - 1) + i_k \quad (3.13)$$

$$y_k = (2v_k^{(2)} - 1) + q_k \quad (3.14)$$

em que  $i_k$  e  $q_k$  são dois ruídos independentes com a mesma variância  $\sigma^2$  e  $v_k^{(2)}$  é a saída não sistemática do codificador.

O algoritmo BCJR calcula a razão de log-verossimilhança  $\Lambda(u_k)$  associada com cada símbolo de informação  $u_k$ .

$$\Lambda(u_k) = \log \frac{P\{u_k = 1 | \mathbf{r}\}}{P\{u_k = 0 | \mathbf{r}\}}, \quad (3.15)$$

em que  $P\{u_k = i | \mathbf{r}\}$ ,  $i = 0, 1$  é a probabilidade *a posteriori* do símbolo de informação  $u_k$ . Considerando  $1 \leq k \leq N$ , em que  $N$  é o comprimento da seqüência recebida, o decodificador decide  $\hat{u}_k = 1$ , se  $P\{u_k = 1 | \text{observação}\} \geq P\{u_k = 0 | \text{observação}\}$ , caso contrário ele decide  $\hat{u}_k = 0$ . A decisão portanto será feita da seguinte forma:

$$\Lambda(u_k) \geq 0 : \hat{u}_k = 1,$$

$$\Lambda(u_k) < 0 : \hat{u}_k = 0.$$

O módulo de  $\Lambda(u_k)$  representa a informação suave associada com o valor abrupto (0 ou 1) estimado de  $u_k$ .

O estado do codificador no  $k$ -ésimo intervalo de tempo é dado por  $S_k$ . A probabilidade *a posteriori* de cada símbolo de informação pode ser extraída da probabilidade conjunta  $\lambda_k^i(m)$  definida por

$$\lambda_k^i(m) = P\{u_k = i, S_k = m | \mathbf{r}_1^N\} \quad (3.16)$$

$$= \frac{P\{u_k = i, S_k = m, \mathbf{r}_1^N\}}{P\{\mathbf{r}_1^N\}}, \quad (3.17)$$

em que  $p\{\mathbf{r}_1^N\}$  representa a função densidade de probabilidade de  $\mathbf{r}_1^N$ .

Portanto, a probabilidade *a posteriori* do símbolo de informação  $u_k$  é

$$P\{u_k = i | \mathbf{r}_1^N\} = \sum_{m=0}^{M-1} \lambda_k^i(m), \quad i = 0, 1.$$

A relação (3.15) portanto pode ser escrita da seguinte forma:

$$\Lambda(u_k) = \log \frac{\sum_{m=0}^{M-1} \lambda_k^1(m)}{\sum_{m=0}^{M-1} \lambda_k^0(m)}. \quad (3.18)$$

De (3.17) e (3.18) segue que

$$\Lambda(u_k) = \log \frac{\sum_m \sum_{m'} p\{u_k = 1, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}{\sum_m \sum_{m'} p\{u_k = 0, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}. \quad (3.19)$$

Levando em consideração que eventos depois do instante de tempo  $k$  não são influenciados pela observação  $\mathbf{r}_1^k$  e pelo bit  $u_k$  se o estado  $S_k$  é conhecido tem-se que

$$\Lambda(u_k) = \log \frac{\sum_m \sum_{m'} p\{\mathbf{r}_{k+1}^N | S_k = m\} p\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} p\{u_k = 1, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}}{\sum_m \sum_{m'} p\{\mathbf{r}_{k+1}^N | S_k = m\} p\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} p\{u_k = 0, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}}. \quad (3.20)$$

Introduzindo as funções de probabilidade abaixo:

$$\alpha_k(m) = p\{S_k = m, \mathbf{r}_1^k\}, \quad (3.21)$$

$$\beta_k(m) = p\{\mathbf{r}_{k+1}^N | S_k = m\}, \quad (3.22)$$

$$\gamma_i(\mathbf{r}_k, m', m) = p\{u_k = i, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}, \quad (3.23)$$

tem-se que:

$$\Lambda(u_k) = \log \frac{\sum_m \sum_{m'} \gamma_1(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \gamma_0(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}, \quad (3.24)$$

em que  $\alpha_k(m)$ , para  $k = 1, 2, \dots, N$ , pode ser calculado de maneira recursiva por meio de:

$$\begin{aligned} \alpha_k(m) &= \sum_{m'=0}^{M-1} p\{S_{k-1} = m', S_k = m, \mathbf{r}_1^k\} \\ &= \sum_{m'=0}^{M-1} p\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} p\{S_k = m, \mathbf{r}_k | S_{k-1} = m', \mathbf{r}_1^{k-1}\} \\ &= \sum_{m'=0}^{M-1} p\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} p\{S_k = m, \mathbf{r}_k | S_{k-1} = m'\} \\ &= \sum_{i=0}^1 \sum_{m'=0}^{M-1} p\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} p\{u_k = i, S_k = m, \mathbf{r}_k | S_{k-1} = m'\} \\ &= \sum_{i=0}^1 \sum_{m'=0}^{M-1} \alpha_{k-1}(m') \gamma_i(\mathbf{r}_k, m', m). \end{aligned} \quad (3.25)$$

A terceira igualdade segue do fato que eventos após o tempo  $k - 1$  não são influenciados por  $\mathbf{r}_1^{k-1}$  se  $S_{k-1}$  é conhecido. Considerando que a treliça é inicializada no estado  $S_0 = \mathbf{0}$ , então as condições de contorno são as seguintes:

$$\alpha_0(\mathbf{0}) = 1 \text{ e } \alpha_0(m) = 0, \text{ para } m \neq 0. \quad (3.26)$$

Similarmente, para  $k = 1, 2, \dots, N - 1$ ,  $\beta_k(m)$  pode ser calculado de maneira recursiva

$$\begin{aligned} \beta_k(m) &= \sum_{m'=0}^{M-1} \text{p}\{S_{k+1} = m'; \mathbf{r}_{k+1}^N | S_k = m\} \\ &= \sum_{m'=0}^{M-1} \text{p}\{S_{k+1} = m'; \mathbf{r}_{k+1} | S_k = m\} \text{p}\{\mathbf{r}_{k+2}^N | S_k(m), S_{k+1} = m', \mathbf{r}_{k+1}\} \\ &= \sum_{m'=0}^{M-1} \text{p}\{S_{k+1} = m'; \mathbf{r}_{k+1} | S_k = m\} \text{p}\{\mathbf{r}_{k+2}^N | S_{k+1} = m'\} \\ &= \sum_{i=0}^1 \sum_{m'=0}^{M-1} \text{p}\{u_{k+1} = i, S_{k+1} = m'; \mathbf{r}_{k+1} | S_k = m\} \text{p}\{\mathbf{r}_{k+2}^N | S_{k+1} = m'\} \\ &= \sum_{i=0}^1 \sum_{m'=0}^{M-1} \gamma_i(\mathbf{r}_{k+1}, m, m') \beta_{k+1}(m'). \end{aligned} \quad (3.27)$$

As condições de contorno apropriadas quando o codificador é levado a terminar no estado  $\mathbf{0}$  isto é,  $S_N = \mathbf{0}$  são:

$$\beta_N(\mathbf{0}) = 1 \text{ e } \beta_N(m) = 0, \text{ para } m \neq 0. \quad (3.28)$$

Quando não se sabe o estado final do codificador, as condições de contorno apropriadas são [39]:

$$\beta_N(\mathbf{0}) = 1/M \text{ e } \beta_N(m) = 0, \text{ para } m \neq 0. \quad (3.29)$$

As probabilidades  $\gamma_i(\mathbf{r}_k, m', m)$  podem ser determinadas a partir das probabilidades de transição do canal contaminado com ruído branco gaussiano e das probabilidades de transição da treliça do codificador

$$\begin{aligned} \gamma_i(\mathbf{r}_k, m', m) &= \text{P}\{S_k = m | S_{k-1} = m'\} \text{p}\{\mathbf{r}_k | u_k = i, S_k = m, S_{k-1} = m'\} \\ &\quad \text{P}\{u_k = i | S_k = m, S_{k-1} = m'\}. \end{aligned} \quad (3.30)$$

As probabilidades de transição  $\text{P}\{S_k = m | S_{k-1} = m'\}$  são definidas pelas probabilidades *a priori* dos bits de entrada. Quando os bits de entrada são equiprováveis  $\text{P}\{u_k = 1\} = \text{P}\{u_k = 0\} = 1/2$  então  $\text{P}\{S_k = m | S_{k-1} = m'\} = 1/2$ . A probabilidade  $\text{P}\{\mathbf{r}_k | u_k = i, S_k = m, S_{k-1} =$

$m'$  é a probabilidade de transição do canal contaminado com ruído branco gaussiano, podendo ser escrita da seguinte forma

$$p\{\mathbf{r}_k | u_k = i, S_k = m, S_{k-1} = m'\} = p\{\mathbf{r}_k | \mathbf{v}_k\}, \quad (3.31)$$

em que

$$p\{\mathbf{r}_k | \mathbf{v}_k\} = \prod_{j=0}^{n-1} p\{r_k^{(j)} | v_k^{(j)}\},$$

e

$$p\{r_k^{(j)} | v_k^{(j)}\} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_k^{(j)} - v_k^{(j)})^2}{2\sigma^2}\right), \quad (3.32)$$

assim,

$$p\{r_k^{(j)} | v_k^{(j)} = -1\} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_k^{(j)} + 1)^2}{2\sigma^2}\right),$$

$$p\{r_k^{(j)} | v_k^{(j)} = +1\} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_k^{(j)} - 1)^2}{2\sigma^2}\right).$$

A probabilidade  $P\{u_k = i | S_k = m, S_{k-1} = m'\}$  é igual a 0 ou 1 uma vez que o codificador convolucional é uma máquina determinística.

Pode-se expressar  $\gamma_i(\mathbf{r}_k, m', m)$  da seguinte forma:

$$\gamma_i(\mathbf{r}_k, m', m) = \begin{cases} P\{u_k = i\} \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - v_{k,i}^{(j)})^2}{2\sigma^2}\right) & \text{para } m \text{ e } m' \in B_k^i \\ 0 & \text{outros,} \end{cases} \quad (3.33)$$

em que  $v_{k,i}^{(j)}$  é a saída do codificador associada com a transição  $S_{k-1} = m'$  para  $S_k = m$  e entrada  $u_k = i$  e  $B_k^i$  é o conjunto de transições  $S_{k-1} = m'$  para  $S_k = m$  que são causadas pelo bit de entrada  $u_k = i$ . Substituindo (3.33) em (3.24) e assumindo que em (3.34) e (3.35) o numerador contempla todas as transições  $S_{k-1} = m'$  para  $S_k = m$  que são causadas pelo bit de entrada  $u_k = 0$  ( $m, m' \in B_k^0$ ) e o denominador contempla todas as transições  $S_{k-1} = m'$  para  $S_k = m$  que são causadas pelo bit de entrada  $u_k = 1$  ( $m, m' \in B_k^1$ ) tem-se:

$$\Lambda(u_k) = \log \frac{\sum_m \sum_{m'} P\{u_k = 1\} \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - v_{k,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} P\{u_k = 0\} \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - v_{k,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}. \quad (3.34)$$

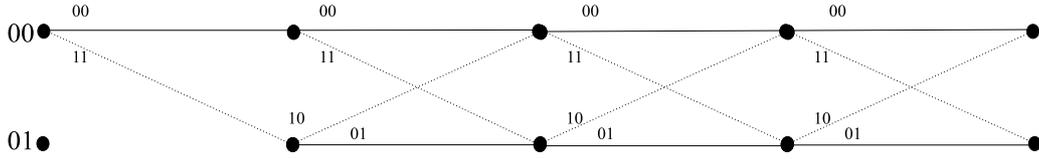


Figura 3.10: Treliça construída a partir da matriz geradora polinomial  $G(D) = [1 \frac{1}{1+D}]$ .

Para simplicidade de notação pode-se fazer  $P\{u_k = i\} = p_k(i)$ :

$$\Lambda(u_k) = \log \frac{\sum_m \sum_{m'} p_k(1) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - v_{k,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} p_k(0) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - v_{k,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}, \quad m, m' \in B_k^i. \quad (3.35)$$

O algoritmo BCJR portanto segue da seguinte forma:

1. As condições iniciais de  $\alpha_0(m)$  e  $\beta_N(m)$  são vistas em (3.26) e (3.28) ou (3.29);
2. Quando  $\mathbf{r}_k$  é recebido, o decodificador calcula  $\gamma_i(\mathbf{r}_k, m', m)$  usando (3.33) e  $\alpha_k(m)$  usando (3.25). Os valores de  $\alpha_k(m)$  são armazenados para todo  $k$  e  $m$ ;
3. Depois que a seqüência completa  $\mathbf{r}_1^N$  é recebida, o decodificador calcula de forma recursiva  $\beta_k(m)$  usando (3.27). O valor de  $\beta_k(m)$  calculado pode ser multiplicado pelo  $\alpha_k(m)$  e  $\gamma_i(\mathbf{r}_k, m', m)$  apropriado para obtenção de (3.35).

**Exemplo 3.6.1** Considere um sistema codificado por um RSC  $(2,1,1)$  com matriz geradora polinomial

$$\mathbf{G}(D) = \left[ 1 \quad \frac{1}{1+D} \right].$$

A treliça está ilustrada na Figura 3.10. A modulação é BPSK e  $\frac{E_b}{N_0} = 2\text{dB}$ , em que  $E_b$  é a energia por bit do sinal e  $N_0$  é a potência do ruído. A seqüência de informação transmitida é:

$$\mathbf{u} = (001101), \quad (3.36)$$

a seqüência código recebida é:

$$\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \mathbf{r}_7) \quad (3.37)$$

e os valores para  $x_k$  e  $y_k$  dados em (3.13) e (3.14) são mostrados na Tabela 3.1. O objetivo é determinar a saída abrupta e a saída suave do decodificador BCJR.

No final da seqüência de entrada é adicionado 1 bit no codificador assim, com uma seqüência de informação de 6 bits, tem-se uma seqüência de saída de 14 bits. Este bit acrescentado não

Tabela 3.1: Tabela contendo os parâmetros  $\alpha, \beta, \gamma$  e  $\Lambda$  resultantes da aplicação do algoritmo BCJR em um sistema codificado por um RSC (2,1,1) com matriz geradora polinomial  $G(D) = [1 \frac{1}{1+D}]$ .

$k$	1	2	3	4	5	6	7
$u_k$	0	0	1	1	0	1	
$x_k$	-0,02360320	-2,10844567	0,71635523	1,42707271	-1,46886132	0,78865002	0,26712210
$y_k$	-1,58697213	-1,06256874	0,27619064	0,27708397	-1,72649526	1,56885652	-2,03069514
$\gamma_0(\mathbf{r}_k, 0, 0)$	0,20706472	0,21645866	0,02236281	0,00302110	0,30090375	0,00064353	0,08165009
$\gamma_0(\mathbf{r}_k, 1, 1)$	0,02776780	0,01206662	0,04736095	0,00641376	0,00276205	0,04568346	0,00032796
$\gamma_1(\mathbf{r}_k, 0, 1)$	0,00260430	0,00003924	0,33165872	0,30974359	0,00005105	0,38934535	0,00067766
$\gamma_1(\mathbf{r}_k, 1, 0)$	0,19420270	0,00070387	0,15660201	0,14589962	0,00556198	0,00548461	0,16871365
$\alpha_k(0)$	0,20706472	0,04482279	0,00100856	0,00217224	0,00065590	0,00000043	0,00004313
$\alpha_k(1)$	0,00260430	0,00003955	0,01486774	0,00040775	0,00000124	0,00025543	0,00000008
$\beta_k(0)$	0,00010413	0,00048104	0,00009000	0,00989098	0,03287021	0,04082504	0,5
$\beta_k(1)$	0,00000133	0,00008250	0,00144433	0,00019409	0,00407762	0,08435683	0
$\Lambda_k(u_k)$	-8,73325520	-9,53915606	5,44279592	6,09200944	-5,65785959	6,88676993	
$\hat{u}_k$	0	0	1	1	0	1	

teve a intenção de levar a treliça o estado final 0, assim as condições de contorno utilizadas para  $\beta_N(m)$  foram:

$$\beta_7(\mathbf{0}) = 0.5 \quad e \quad \beta_7(m) = 0, \quad \text{para } m \neq 0.$$

A taxa do código é

$$R = \frac{5}{10 + 2} = 0.416.$$

A variância do ruído branco gaussiano é

$$\sigma^2 = \frac{N_0}{2}.$$

A relação sinal ruído é

$$SNR = \frac{E_s}{N_0},$$

em que  $E_s$  é a energia por bloco do sinal,

$$E_s = RE_b.$$

A variância pode ser escrita da seguinte forma

$$\sigma^2 = \frac{E_s}{2SNR} = \frac{0.5}{R \frac{E_b}{N_0}}.$$

Substituindo os valores e colocando  $\frac{E_b}{N_0}$  em dB tem-se:

$$\sigma^2 = \frac{0.5}{0.416(10^{0.2})}.$$

Os valores encontrados em cada instante de tempo para  $\gamma_i(\mathbf{r}_k, m', m)$ ,  $\alpha_k(m)$ ,  $\beta_k(m)$  e  $\Lambda_k(u_k)$  estão ilustrados na Tabela 3.1.

### 3.6.2 BCJR Modificado

Quando se utiliza o algoritmo BCJR com o RSC, o valor de  $\alpha_k(m)$  decresce rapidamente à medida que o valor de  $k$  aumenta e o valor de  $\beta_k(m)$  decresce rapidamente à medida que o valor de  $k$  diminui, havendo dificuldades no cálculo de (3.24). Com o intuito de resolver este problema, utiliza-se então o algoritmo BCJR modificado [39] [52] [55].

Partindo de (3.20) e introduzindo as funções de probabilidade:

$$\alpha_k(m) = P\{S_k = m | \mathbf{r}_1^k\}, \quad (3.38)$$

$$\beta_k(m) = \frac{P\{\mathbf{r}_{k+1}^N | S_k = m\}}{P\{\mathbf{r}_{k+1}^N | \mathbf{r}_1^k\}}, \quad (3.39)$$

$$\gamma_i(\mathbf{r}_k, m', m) = P\{u_k = i, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}. \quad (3.40)$$

encontra-se (3.24).

Pode-se ver agora como (3.38) e (3.39) podem ser calculadas de maneira recursiva. De (3.38) tem-se:

$$\begin{aligned} \alpha_k(m) &= P\{S_k = m | \mathbf{r}_1^k\} \\ &= \frac{P\{S_k = m, \mathbf{r}_1^k\}}{P\{\mathbf{r}_1^k\}} = \frac{P\{S_k = m, \mathbf{r}_1^{k-1}, \mathbf{r}_k\}}{P\{\mathbf{r}_1^{k-1}, \mathbf{r}_k\}} \\ &= \frac{P\{S_k = m, \mathbf{r}_k | \mathbf{r}_1^{k-1}\}}{P\{\mathbf{r}_k | \mathbf{r}_1^{k-1}\}}. \end{aligned} \quad (3.41)$$

Pode-se expressar o numerador de (3.41) em função do bit  $u_k$  e do estado  $S_{k-1}$  da seguinte forma:

$$\begin{aligned} P\{S_k = m, \mathbf{r}_k | \mathbf{r}_1^{k-1}\} &= \sum_{m'} \sum_{i=0}^1 P\{u_k = i, S_{k-1} = m', S_k = m, \mathbf{r}_k | \mathbf{r}_1^{k-1}\} \\ &= \sum_{m'} \sum_{i=0}^1 P\{u_k = i, S_k = m, \mathbf{r}_k | S_{k-1} = m', \mathbf{r}_1^{k-1}\} P\{S_{k-1} = m', | \mathbf{r}_1^{k-1}\} \\ &= \sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_k, m', m) \alpha_{k-1}(m'). \end{aligned} \quad (3.42)$$

A Equação (3.42) vem do fato que eventos após  $(k - 1)$  são independentes da observação  $\mathbf{r}_1^{k-1}$ , se  $S_{k-1}$  é conhecido. O denominador de (3.41) pode ser expresso da seguinte forma

$$\begin{aligned}
\mathbb{P}\{\mathbf{r}_k | \mathbf{r}_1^{k-1}\} &= \sum_m \sum_{m'} \sum_{i=0}^1 \mathbb{P}\{u_k = i, S_k = m, S_{k-1} = m', \mathbf{r}_k | \mathbf{r}_1^{k-1}\} \\
&= \sum_m \sum_{m'} \sum_{i=0}^1 \mathbb{P}\{u_k = i, S_k = m, \mathbf{r}_k | S_{k-1} = m', \mathbf{r}_1^{k-1}\} \mathbb{P}\{S_{k-1} = m' | \mathbf{r}_1^{k-1}\} \\
&= \sum_m \sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_k, m', m) \alpha_{k-1}(m'). \tag{3.43}
\end{aligned}$$

Finalmente, a probabilidade  $\alpha_k(m)$  pode ser expressa a partir da probabilidade  $\alpha_{k-1}(m)$  utilizando (3.42) e (3.43),

$$\alpha_k(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_k, m', m) \alpha_{k-1}(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_k, m', m) \alpha_{k-1}(m')}. \tag{3.44}$$

De maneira similar,  $\beta_k(m)$  pode ser calculado recursivamente. De (3.39) tem-se:

$$\begin{aligned}
\beta_k(m) &= \frac{\mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\}}{\mathbb{P}\{\mathbf{r}_{k+1}^N | \mathbf{r}_1^k\}} \\
&= \frac{\sum_{m'} \sum_{i=0}^1 \mathbb{P}\{u_{k+1} = i, S_{k+1} = m', \mathbf{r}_{k+2}^N, \mathbf{r}_{k+1} | S_k = m\}}{\mathbb{P}\{\mathbf{r}_{k+1}^N | \mathbf{r}_1^k\}}. \tag{3.45}
\end{aligned}$$

Pode-se reescrever o numerador de (3.45) da seguinte forma

$$\begin{aligned}
\mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\} &= \sum_{m'} \sum_{i=0}^1 \mathbb{P}\{\mathbf{r}_{k+2}^N | S_{k+1} = m'\} \mathbb{P}\{u_{k+1} = i, S_{k+1} = m', \mathbf{r}_{k+1} | S_k = m\} \\
&= \sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_{k+1}, m, m') \beta_{k+1}(m') \mathbb{P}\{\mathbf{r}_{k+2}^N | \mathbf{r}_1^{k+1}\}. \tag{3.46}
\end{aligned}$$

É possível escrever  $\beta_k(m)$  da seguinte forma

$$\beta_k(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_{k+1}, m, m') \beta_{k+1}(m')}{\mathbb{P}\{\mathbf{r}_{k+1}^N | \mathbf{r}_1^k\}}. \tag{3.47}$$

Substituindo  $k$  por  $k + 1$  em (3.43) o denominador de (3.47) é dado por

$$\mathbb{P}\{\mathbf{r}_{k+1}^N | \mathbf{r}_1^k\} = \sum_m \sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_{k+1}, m', m) \alpha_k(m'). \tag{3.48}$$

Finalmente, a probabilidade  $\beta_k(m)$  pode ser expressa a partir da probabilidade  $\beta_{k+1}(m')$  pela seguinte relação:

$$\beta_k(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_{k+1}, m, m') \beta_{k+1}(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \gamma_i(\mathbf{r}_{k+1}, m', m) \alpha_k(m')}. \tag{3.49}$$

Os passos do algoritmo BCJR modificado seguem similarmente ao do algoritmo BCJR:

Tabela 3.2: Tabela contendo os parâmetros  $\alpha, \beta, \gamma$  e  $\Lambda$  resultantes da aplicação do algoritmo BCJR modificado em um sistema codificado por um RSC (2,1,1) com matriz geradora polinomial  $G(D) = \left[1 \frac{1}{1+D}\right]$ .

$k$	1	2	3	4	5	6	7
$u_k$	0	0	1	1	0	1	
$x_k$	-0,02360320	-2,10844567	0,71635523	1,42707271	-1,46886132	0,78865002	0,26712210
$y_k$	-1,58697213	-1,06256874	0,27619064	0,27708397	-1,72649526	1,56885652	-2,03069514
$\gamma_0(\mathbf{r}_k, 0, 0)$	0,20706472	0,21645866	0,02236281	0,00302110	0,30090375	0,00064353	0,08165009
$\gamma_0(\mathbf{r}_k, 1, 1)$	0,02776780	0,01206662	0,04736095	0,00641376	0,00276205	0,04568346	0,00032796
$\gamma_1(\mathbf{r}_k, 0, 1)$	0,00260430	0,00003924	0,33165872	0,30974359	0,00005105	0,38934535	0,00067766
$\gamma_1(\mathbf{r}_k, 1, 0)$	0,19420270	0,00070387	0,15660201	0,14589962	0,00556198	0,00548461	0,16871365
$\alpha_k(0)$	0,98757901	0,99911842	0,06352596	0,84195632	0,99811740	0,00167624	0,99805477
$\alpha_k(1)$	0,01242099	0,00088158	0,93647404	0,15804368	0,00188260	0,99832376	0,00194523
$\beta_k(0)$	1,01044470	0,99878427	0,06612931	1,18104948	0,99970333	0,48343339	0,5
$\beta_k(1)$	0,01294564	0,17129327	1,06127217	0,02317516	0,12401539	0,99891884	0
$\Lambda_k(u_k)$	-8,73325520	-9,53915606	5,44279592	6,09200944	-5,65785959	6,88676993	
$\hat{u}_k$	0	0	1	1	0	1	

1. As condições iniciais de  $\alpha_0(m)$  e  $\beta_N(m)$  são vistas em (3.26) e (3.28) ou (3.29) ;
2. Quando  $\mathbf{r}_k$  é recebido, o decodificador calcula  $\gamma_i(\mathbf{r}_k, m', m)$  usando (3.30) e  $\alpha_k(m)$  usando (3.44). Os valores de  $\alpha_k(m)$  são armazenados para todo  $k$  e  $m$ ;
3. Depois que a seqüência completa  $\mathbf{r}_1^N$  é recebida, o decodificador calcula de forma recursiva  $\beta_k(m)$  usando (3.49). O valor de  $\beta_k(m)$  calculado pode ser multiplicado pelo  $\alpha_k(m)$  e  $\gamma_i(\mathbf{r}_k, m', m)$  apropriado para obtenção de (3.24).

**Exemplo 3.6.2** *Pode-se resolver o exemplo anterior utilizando o algoritmo BCJR modificado. O código utilizado é um RSC (2, 1, 1) com matriz geradora polinomial*

$$\mathbf{G}(D) = \left[ 1 \quad \frac{1}{1+D} \right].$$

*O valor de  $\mathbf{r}$  recebido é exatamente o mesmo existente na Tabela 3.1. Observa-se na Tabela 3.2 que os valores de  $\gamma_i(\mathbf{r}_k, m', m)$  e  $\Lambda(u_k)$  são os mesmos valores encontrados na Tabela 3.1. Os valores de  $\alpha_k(m)$ , calculados de acordo com (3.44), não decrescem monotonicamente à medida que  $k$  cresce. Os valores de  $\beta_k(m)$ , calculados de acordo com (3.49), não decrescem monotonicamente à medida que os valores de  $k$  vão se tornando menores.*

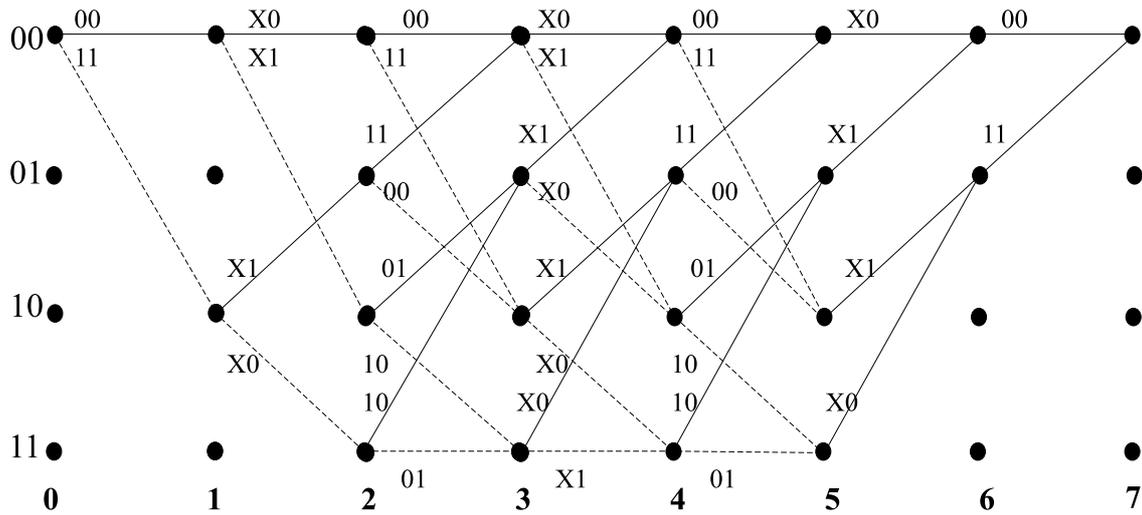


Figura 3.11: Diagrama de treliça para o código perfurado gerado a partir do código com matriz geradora  $G(D) = [1 + D^2 \ 1 + D + D^2]$ . A cada quatro símbolos de saída um é perfurado (extraído), o codificador produz três símbolos na saída para cada dois símbolos de informação. O novo código portanto terá taxa assintótica  $2/3$ .

### 3.7 Código Convolutional Perfurado

Em certas situações, há a necessidade de variar a taxa de um código convolutional, sem entretanto alterar a estrutura do codificador. A taxa do código é alterada pela não transmissão de certos símbolos de paridade da palavra código ou perfurando o código original [6] [54]. A principal razão para a construção de códigos perfurados é o fato das suas treliças terem estrutura mais simples do que as correspondentes para códigos não perfurados, permitindo simplificar a implementação do algoritmo de Viterbi, com um pequeno aumento na probabilidade de erro de bit. Códigos perfurados são códigos convolucionais com parâmetros  $(n, k, m)$ , em geral obtidos a partir de um código convolutional com parâmetros  $(n, 1, m)$ . É possível demonstrar o método de construção usando um exemplo.

**Exemplo 3.7.1** Considere o codificador mostrado na Figura 3.6, com matriz geradora

$$G(D) = \begin{bmatrix} 1 + D^2 & 1 + D + D^2 \end{bmatrix},$$

e diagrama de treliça ilustrado na Figura 3.7. Se a cada quatro símbolos de saída um é perfurado (extraído), o codificador produz três símbolos na saída para cada dois símbolos de informação. O novo código portanto terá taxa assintótica  $2/3$ . O diagrama de treliça para este código é mostrado na Figura 3.11. O “X” indica os símbolos perfurados. O conjunto de seqüências código produzido por esse código perfurado é idêntico ao conjunto de seqüências código gerado

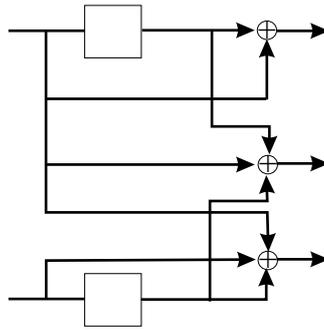


Figura 3.12: codificador de um código com taxa 2/3.

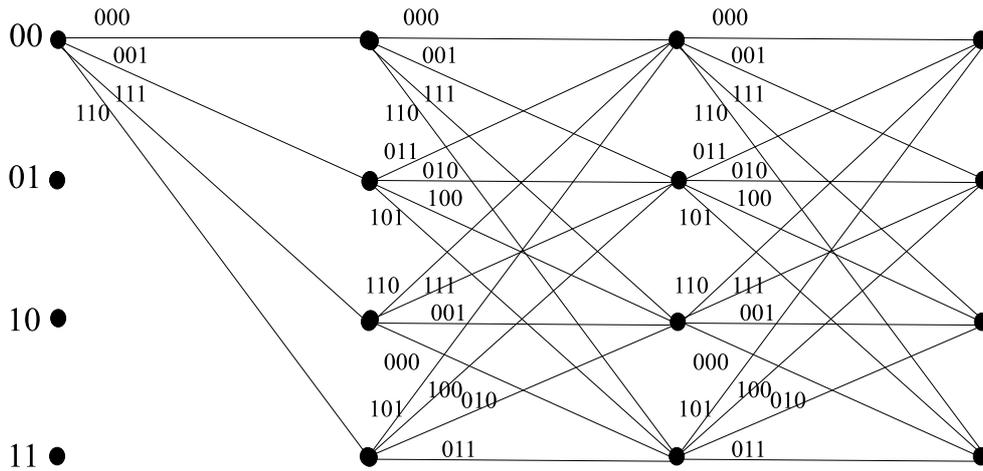


Figura 3.13: Treliça resultante do codificador ilustrado na Figura 3.12.

por um código com taxa assintótica 2/3 com matriz geradora

$$\mathbf{G}(D) = \begin{bmatrix} 1 + D & 1 + D & 1 \\ 0 & D & 1 + D \end{bmatrix}.$$

O codificador e o diagrama de treliça para este código estão ilustrados nas Figuras 3.12 e 3.13, respectivamente. A treliça ilustrada na Figura 3.13 é mais complexa do que a treliça para o código perfurado com taxa assintótica 2/3 da Figura 3.11, uma vez que existem quatro ramos entrando em cada estado em vez de apenas dois ramos. Assim, as operações de codificação e decodificação são mais complexas.

A extração de símbolos do código na treliça da Figura 3.11 pode ser descrita por uma tabela de perfuração

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

em que um “0” na tabela significa que um símbolo do código não é transmitido. No exemplo acima, o primeiro símbolo no segundo ramo não é transmitido.

Em geral, um código convolucional de taxa  $p/q$  perfurado pode ser construído a partir de um código com parâmetros  $(n, 1, m)$  após extração de  $(np - q)$  símbolos de cada  $(np)$  símbolos do código correspondendo a  $p$  símbolos de informação. O código  $(n, 1, m)$  é chamado de código mãe. Ele é especificado pela matriz geradora

$$\left[ \mathbf{g}_1^{(1)} \mathbf{g}_1^{(2)} \cdots \mathbf{g}_1^{(n)} \right].$$

Os símbolos extraídos são representados pela tabela de perfuração

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1p} \\ p_{21} & p_{22} & \cdots & p_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{np} \end{bmatrix},$$

com  $p_{ij} \in \{0, 1\}$ ,  $1 \leq i \leq n$  e  $1 \leq j \leq p$ . Como a perfuração é feita periodicamente a cada  $np$  símbolos do código,  $p$  é chamado de período de perfuração.

# Capítulo 4

## Códigos Turbo

Códigos turbo foram introduzidos na literatura por Berrou, Glavieux, e Thitimajshima [39]. Os códigos são construídos usando dois ou mais códigos componentes, tendo como entradas versões entrelaçadas da mesma seqüência de informação. Para o código turbo apresentar os resultados desejados, o algoritmo de decodificação utilizado deve passar decisões suaves de um decodificador para outro e repetir este processo inúmeras vezes [51] [53].

### 4.1 Entrelaçador

Um entrelaçador é um dispositivo com uma única entrada e uma única saída, cuja entrada é uma seqüência de símbolos de um determinado alfabeto e cuja saída é uma seqüência de símbolos do mesmo alfabeto, idêntica a entrada, mas ordenada de uma maneira diferente [4] [6].

Considere o entrelaçador  $I$  de comprimento  $N$  mostrado na Figura 4.1. Considere que a seqüência de dados na entrada do entrelaçador  $I$  é dada por

$$\mathbf{u} = (u_1, u_2, u_3, \dots, u_N),$$

o entrelaçador então permuta a seqüência  $u$  para uma seqüência

$$\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \dots, \tilde{u}_N),$$

A seqüência  $\tilde{\mathbf{u}}$  tem todos os elementos de  $\mathbf{u}$  mas em uma ordem diferente. Ao considerar a seqüência de entrada  $\mathbf{u}$  e a seqüência de saída  $\tilde{\mathbf{u}}$  como um par de conjuntos com  $N$  elementos, existe uma correspondência unívoca

$$u_i \longrightarrow \tilde{u}_j \quad 1 \leq i \leq N, \quad 1 \leq j \leq N,$$



Figura 4.1: Entrelaçador  $I$  com seqüência de dados na entrada  $\mathbf{u} = (u_1, u_2, u_3, \dots, u_N)$  e seqüência de dados na saída  $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \dots, \tilde{u}_N)$ .

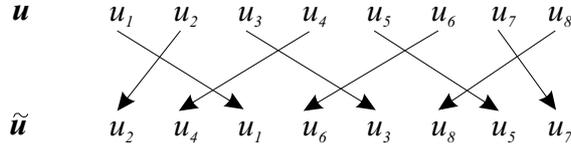


Figura 4.2: Função de mapeamento correspondente para entrelaçador com seqüência de entrada  $\mathbf{u} = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8)$  e seqüência entrelaçada  $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4, \tilde{u}_5, \tilde{u}_6, \tilde{u}_7, \tilde{u}_8) = (u_2, u_4, u_1, u_6, u_3, u_8, u_5, u_7)$ .

entre cada elemento de  $\mathbf{u}$  e cada elemento de  $\tilde{\mathbf{u}}$ .

Considere um conjunto  $A = 1, 2, \dots, N$ . O entrelaçador pode ser definido pela função de mapeamento

$$\pi(A \longrightarrow A) : j = \pi(i), \quad i, j \in A$$

em que  $i$  e  $j$  são os índices do elemento na seqüência original  $\mathbf{u}$  e na seqüência entrelaçada  $\tilde{\mathbf{u}}$ , respectivamente. A função de mapeamento pode ser representada por um vetor de entrelaçamento

$$\pi_N = (\pi(1), \pi(2), \pi(3), \dots, \pi(N)).$$

Considere, por exemplo, um entrelaçador com comprimento  $N = 8$ . A seqüência de entrada é representada por:

$$\mathbf{u} = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8).$$

A seqüência entrelaçada é:

$$\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4, \tilde{u}_5, \tilde{u}_6, \tilde{u}_7, \tilde{u}_8) \quad (4.1)$$

$$= (u_2, u_4, u_1, u_6, u_3, u_8, u_5, u_7). \quad (4.2)$$

A função de mapeamento correspondente está ilustrada na Figura 4.2. O vetor de entrelaçamento é expresso como

$$\pi_8 = (\pi(1), \pi(2), \pi(3), \pi(4), \pi(5), \pi(6), \pi(7), \pi(8)) \quad (4.3)$$

$$= (3, 1, 5, 2, 7, 4, 8, 6). \quad (4.4)$$

Tabela 4.1: Bloco  $3 \times 3$ , usado para exemplificar os tipos de entrelaçadores de bloco.

0	1	2
3	4	5
6	7	8

Um desentrelaçador atua na saída do entrelaçador e põe os símbolos de volta na ordem original.

#### 4.1.1 Entrelaçador de Bloco

Um entrelaçador de bloco clássico pode ser descrito em termos de uma matriz  $N \times M$ . Esses entrelaçadores são caracterizados por um processo no qual os dados são escritos como linhas de uma matriz e lidos ao longo de colunas. Existem quatro tipos de entrelaçadores de bloco clássicos, que variam de acordo com a ordem na qual as colunas e as linhas são lidas :

- LR/TB - As linhas são lidas da esquerda para a direita e as colunas de cima para baixo;
- LR/BT - As linhas são lidas da esquerda para a direita e as colunas de baixo para cima;
- RL/TB - As linhas são lidas da direita para a esquerda e as colunas de cima para baixo;
- RL/BT - As linhas são lidas da direita para a esquerda e as colunas de baixo para cima.

**Exemplo 4.1.1** *Considere um caso simples em que  $N = 3$  e  $M = 3$ , como mostrado na Tabela 4.1.*

*Vejamos alguns tipos de entrelaçamento:*

*O entrelaçamento LR/TB*

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 3 & 6 & 1 & 4 & 7 & 2 & 5 & 8 \end{pmatrix};$$

*O entrelaçamento LR/BT:*

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 3 & 0 & 7 & 4 & 1 & 8 & 5 & 2 \end{pmatrix};$$

*O entrelaçamento RL/TB:*

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 5 & 8 & 1 & 4 & 7 & 0 & 3 & 6 \end{pmatrix};$$

O entrelaçamento RL/BT:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 5 & 2 & 7 & 4 & 1 & 6 & 3 & 0 \end{pmatrix}.$$

### 4.1.2 Entrelaçador de Berrou-Glavieux

Este entrelaçador foi usado por Berrou e Glavieux [4] [39] [52] para uso em códigos turbo.

Sejam  $N$  e  $M$  potências de 2. Defina oito números primos  $p(1) = 17$ ,  $p(2) = 37$ ,  $p(3) = 19$ ,  $p(4) = 29$ ,  $p(5) = 41$ ,  $p(6) = 23$ ,  $p(7) = 13$ ,  $p(8) = 7$ . Para cada  $0 \leq i < N.M = T$ , faça

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

onde

$$r(i) = \text{mod}(p(l+1).(c_0+1) - 1, N),$$

$$c(i) = \text{mod}((M/2+1).(r_0+c_0), M),$$

$$r_0 = \text{mod}(i, M),$$

$$c_0 = (i - r_0)/M,$$

$$l = \text{mod}((r_0+c_0), 8).$$

## 4.2 O Codificador

Um codificador turbo é formado pela concatenação paralela de dois códigos convolucionais recursivos separados por um entrelaçador. A estrutura do codificador é paralela uma vez que os dois codificadores operam com o mesmo conjunto de símbolos de entrada. Códigos turbo, portanto são referidos como códigos convolucionais de concatenação paralela [4] [6] [39] [52] [53].

Vemos na Figura 4.3 um diagrama de blocos de um codificador turbo com taxa 1/3. A matriz geradora polinomial de cada código componente com taxa 1/2 pode ser representada por:

$$\mathbf{G}(D) = \begin{bmatrix} 1 & \frac{g_1(D)}{g_0(D)} \end{bmatrix}.$$

No codificador a mesma seqüência de informação é codificada duas vezes, mas em uma ordem diferente. O codificador 1 opera diretamente com a seqüência de entrada  $\mathbf{u}$ , de comprimento  $L$  e tem duas saídas. A primeira saída  $\mathbf{v}^{(0)}$ , é igual a seqüência de entrada, pois o codificador

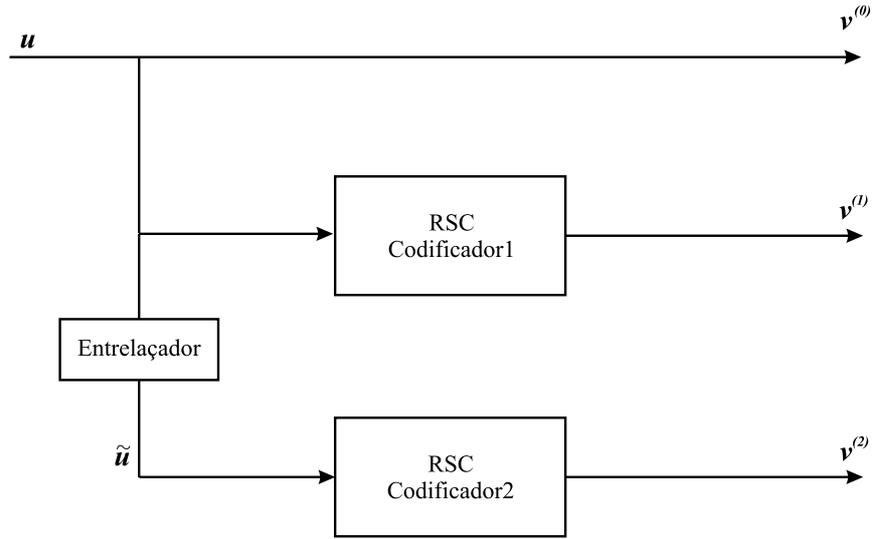


Figura 4.3: Diagrama de blocos de um codificador turbo com taxa 1/3.

é sistemático. A segunda saída é a sequência de paridade  $\mathbf{v}^{(1)}$ . A sequência de informação  $\mathbf{u}$  é entrelaçada, gerando  $\tilde{\mathbf{u}}$ , que é usada como entrada para o codificador 2. Em relação ao codificador 2 somente a sequência de paridade  $\mathbf{v}^{(2)}$  é transmitida. A sequência de informação  $\mathbf{u} = \mathbf{v}^{(0)}$  e as sequências de paridade dos dois codificadores  $\mathbf{v}^{(1)}$  e  $\mathbf{v}^{(2)}$ , são multiplexadas para gerar a sequência de saída do codificador turbo. A taxa assintótica total é 1/3.

**Exemplo 4.2.1** *Um codificador turbo com taxa 1/3, utilizando dois códigos RSC idênticos com parâmetros (2, 1, 4), é mostrado na Figura 4.4. Os dois códigos RSC componentes têm taxa assintótica 1/2 e cada codificador tem 16 estados. A matriz geradora do código é dada por*

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

Considere que

$$\mathbf{u} = (1011001),$$

e que o estado inicial é o estado (0000), então tem-se

$$\mathbf{v}^{(0)} = (1011001),$$

$$\mathbf{v}^{(1)} = (1110001).$$

Após passagem pelo entrelaçador, cujo vetor de entrelaçamento é  $\pi_7 = (4, 3, 1, 2, 5, 7, 6)$ , temos que

$$\tilde{\mathbf{u}} = (1101010).$$

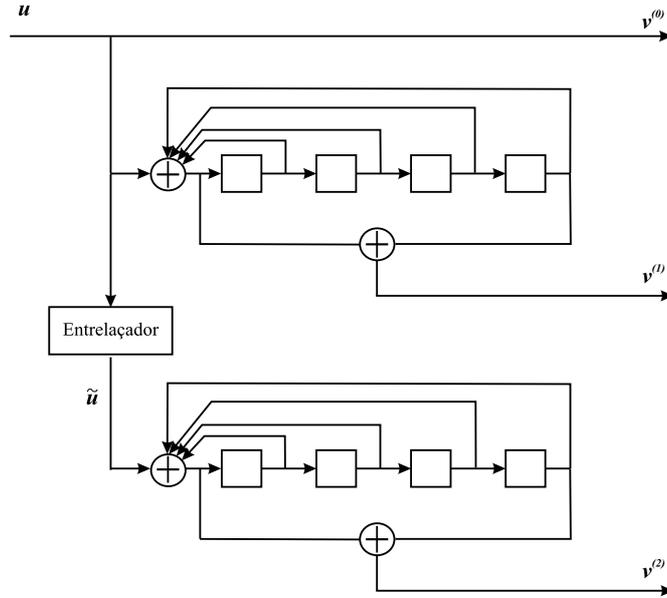


Figura 4.4: Codificador turbo com taxa 1/3, utilizando dois códigos RSC idênticos com parâmetros (2, 1, 4).

Considerando que o codificador 2 também tem o seu estado inicial igual a (0000), a seqüência de paridade gerada é

$$\mathbf{v}^{(2)} = (1011000).$$

A seqüência turbo é então

$$\mathbf{v} = (111, 011, 111, 101, 000, 000, 110).$$

### 4.3 O Decodificador

O decodificador turbo utiliza o princípio da decodificação iterativa [51] e consiste de dois decodificadores componentes concatenados em série como pode ser visto na Figura 4.5. Pode-se supor que os dois decodificadores usam o algoritmo BCJR.

Na entrada do decodificador DEC1 tem-se as seqüências recebidas  $\mathbf{r}^{(0)}$  e  $\mathbf{r}^{(1)}$ , correspondentes respectivamente às seqüências de informação e paridade geradas pelo codificador 1. DEC1 então produz uma saída suave, que é entrelaçada e usada para produzir uma estimativa das probabilidades *a priori* das seqüências de informação para o decodificador DEC2.

Na entrada de DEC2 estão as seqüências recebidas,  $\tilde{\mathbf{r}}^{(0)}$  e  $\mathbf{r}^{(2)}$  correspondentes respectivamente à seqüência de informação entrelaçada e a seqüência de paridade produzida pelo codificador 2. O decodificador DEC2 também produz uma saída suave que é usada para melhorar a estimativa das probabilidades *a priori* das seqüências de informação na entrada de DEC1.

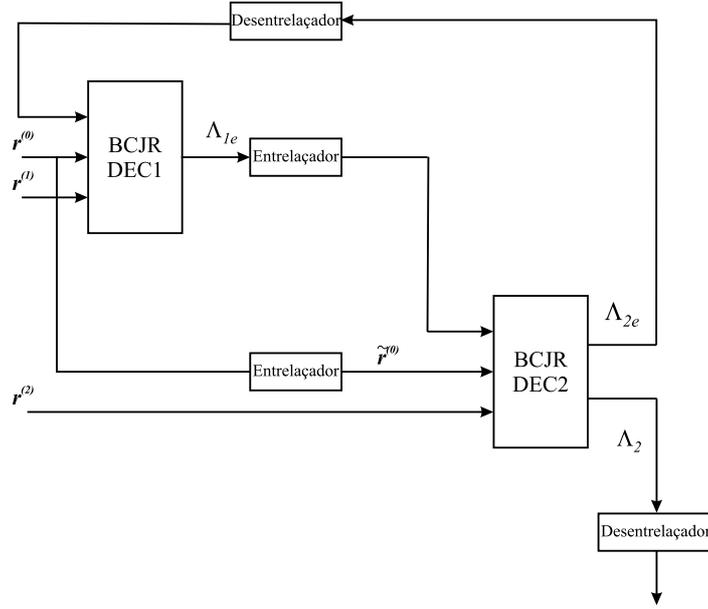


Figura 4.5: O decodificador turbo consiste de dois decodificadores componentes concatenados em série.

Pode-se examinar os decodificadores BCJR (DEC1 e DEC2) para códigos componentes com taxa  $1/n$ , assim,  $(n - 1)$  é o número de bits de paridade do bloco codificado. A razão de log-verossimilhança dada em (3.35) pode ser reescrita como abaixo para DEC1, ficando subentendido, para simplicidade de notação, que  $m, m' \in B_k^i$ , em toda esta seção.

$$\Lambda_1(u_k) = \log \frac{\sum_m \sum_{m'} p_k^1(1) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - v_{k,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} p_k^1(0) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - v_{k,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}, \quad (4.5)$$

$$\Lambda_1(u_k) = \log \frac{p_k^1(1)}{p_k^1(0)} + \log \frac{\sum_m \sum_{m'} \exp\left(-\frac{(r_k^{(0)} - v_{k,1}^{(0)}) + \sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \exp\left(-\frac{(r_k^{(0)} - v_{k,0}^{(0)}) + \sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}. \quad (4.6)$$

Na qual introduziu-se a notação  $p_k^1(1)$  e  $p_k^1(0)$  para representar as probabilidades *a priori*  $P\{u_k = 1\}$  e  $P\{u_k = 0\}$  respectivamente, na entrada do primeiro decodificador. Na entrada do segundo decodificador as probabilidades *a priori*  $P\{\tilde{u}_k = 1\}$  e  $P\{\tilde{u}_k = 0\}$  serão representadas como  $p_k^2(1)$  e  $p_k^2(0)$  respectivamente.

Como o código é sistemático  $v_{k,i}^{(0)}$ ,  $i = 0, 1$  é independente da treliça e dos estados  $m$  e  $m'$ . Assim, considerando  $v_{k,1}^{(0)} = 1$  e  $v_{k,0}^{(0)} = -1$  tem-se:

$$\Lambda_1(u_k) = \log \frac{p_k^1(1)}{p_k^1(0)} + \frac{2}{\sigma^2} r_k^{(0)} + \Lambda_{1e}(u_k), \quad (4.7)$$

em que,

$$\Lambda_{1e}(u_k) = \log \frac{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}. \quad (4.8)$$

O termo  $\Lambda_{1e}$  é chamado de informação extrínseca e é função da informação redundante introduzida pelo codificador. Ele não contém a entrada do decodificador  $r_k^{(0)}$ . O termo  $\Lambda_{1e}$  é usado para melhorar a estimativa da probabilidade *a priori* para o próximo estágio de decodificação.

Pode-se agora observar a entrada de DEC2. Como a entrada para DEC2 inclui  $\tilde{\mathbf{r}}^{(0)}$ , que é uma versão entrelaçada de  $\mathbf{r}^{(0)}$ , a seqüência recebida por DEC2 está correlacionada com a saída suave entrelaçada do primeiro decodificador  $\tilde{\Lambda}_1(u_k)$ . Portanto, a contribuição devida a  $r_k^{(0)}$  deve ser retirada de  $\Lambda_1(u_k)$  para eliminar esta correlação. Como  $\Lambda_{1e}(u_k)$  não contém  $r_k^{(0)}$ , este termo pode ser usado como probabilidade *a priori* para decodificação do segundo estágio, isto é, a informação extrínseca de DEC1 ( $\tilde{\Lambda}_{1e}(u_k)$ ) é a estimativa da probabilidade *a priori* para DEC2.

$$\Lambda_{1e}(u_k) = \log \frac{p_k^2(1)}{p_k^2(0)}. \quad (4.9)$$

Utilizando (4.9) e a igualdade  $p_k^2(1) = 1 - p_k^2(0)$ , pode-se escrever as probabilidades *a priori* de DEC2 da seguinte forma

$$p_k^2(1) = \frac{\exp \tilde{\Lambda}_{1e}(u_k)}{1 + \exp \tilde{\Lambda}_{1e}(u_k)}, \quad (4.10)$$

$$p_k^2(0) = \frac{1}{1 + \exp \tilde{\Lambda}_{1e}(u_k)}. \quad (4.11)$$

No segundo estágio da decodificação, DEC2 estima a razão de log-verossimilhança  $\Lambda_2(u_k)$ . Similarmente como em (4.8), a razão de log-verossimilhança pode ser escrita como

$$\Lambda_2(u_k) = \log \frac{p_k^2(1)}{p_k^2(0)} + \frac{2}{\sigma^2} \tilde{r}_k^{(0)} + \Lambda_{2e}(u_k). \quad (4.12)$$

Substituindo (4.10) e (4.11) em (4.12) tem-se

$$\Lambda_2(u_k) = \tilde{\Lambda}_{1e}(u_k) + \frac{2}{\sigma^2} \tilde{r}_k^{(0)} + \Lambda_{2e}(u_k), \quad (4.13)$$

em que  $\Lambda_{2e}(u_k)$  é a informação extrínseca para o segundo decodificador, que depende da informação redundante suprida pelo codificador 2, como em (4.9). A informação extrínseca de DEC2 pode ser usada como uma estimativa das probabilidades *a priori* para DEC1 como em (4.8). A razão de log-verossimilhança para DEC1 pode ser escrita como

$$\Lambda_1(u_k) = \hat{\Lambda}_{2e}(u_k) + \frac{2}{\sigma^2} r_k^{(0)} + \Lambda_{1e}(u_k), \quad (4.14)$$

em que  $\hat{\Lambda}_{2e}$  corresponde ao  $\Lambda_{2e}$  desentrelaçado.

# Capítulo 5

## Códigos de Treliça Unicamente Decodificáveis para o Canal Aditivo com Dois Usuários Binários

No que segue supõe-se que cada um dos dois usuários do 2-BAC dispõe de um código com estrutura em treliça, cujos detalhes de construção serão apresentados na Seção 5.1.

### 5.1 Treliça para o 2-BAC

A seguir está descrita a construção de Peterson e Costello [56] de uma treliça para o 2-BAC, denominada *treliça para dois usuários*, a partir das treliças individuais de cada usuário. Deve-se supor que as treliças de cada usuário são iniciadas num mesmo instante de tempo e que são considerados pares de ramos (um ramo de cada treliça) que ocorrem em um mesmo intervalo de tempo. Ao estado  $S_k = s_i$ , na treliça do usuário 1, e ao estado  $S'_k = s_r$ , na treliça do usuário 2, associa-se o estado denotado por  $s_i s_r$ , na treliça para dois usuários. Cada par de ramos, ocorrendo em um mesmo intervalo de tempo, nas respectivas treliças de dois usuários, é associado a um único ramo na treliça para dois usuários. Dito de outra forma, se o ramo correspondente ao usuário 1 segue do estado  $s_i$  para o estado  $s_j$ , e o ramo correspondente ao usuário 2 segue do estado  $s_r$  para o estado  $s_l$ , então na treliça para dois usuários corresponde um único ramo seguindo do estado  $s_i s_r$  para o estado  $s_j s_l$ . Se a treliça de cada usuário tem respectivamente  $L_1$  e  $L_2$  estados, a treliça para dois usuários terá  $L_1 L_2$  estados. O conceito de treliça para dois usuários é ilustrado por meio de um exemplo.

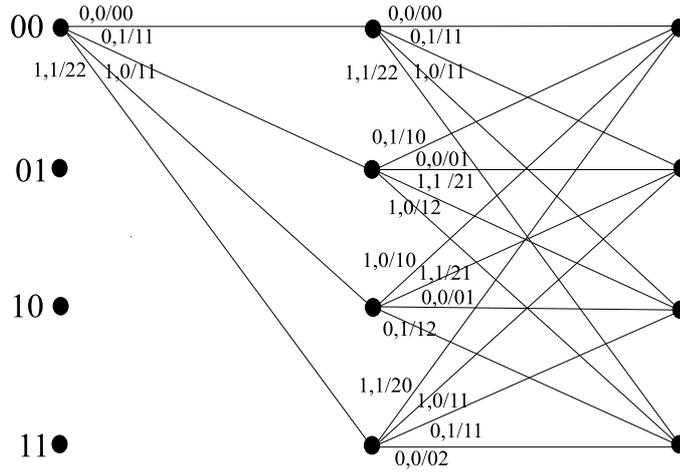


Figura 5.1: Treliça para dois usuários em que, para cada usuário, é usado um mesmo código convolucional com matriz geradora  $G(D) = \begin{bmatrix} 1 & \frac{1}{1+D} \end{bmatrix}$ . Os rótulos nos ramos  $(u_k, d_k/\text{saída})$  correspondem respectivamente ao par de símbolos de informação do usuário 1 e usuário 2 e à saída do 2-BAC sem ruído.

**Exemplo 5.1.1** Vamos supor uma situação hipotética, apenas para ilustrar a construção da treliça para dois usuários, na qual um mesmo código convolucional é usado por cada usuário do 2-BAC. Seja  $\mathcal{C}$  o código convolucional recursivo sistemático com taxa assintótica  $1/2$  e matriz geradora polinomial:

$$G(D) = \begin{bmatrix} 1 & \frac{1}{1+D} \end{bmatrix} \quad (5.1)$$

Como o código tem apenas um elemento de memória, a treliça de cada usuário possui 2 estados, i.e.  $L_1 = L_2 = 2$ , e a treliça para dois usuários terá  $L_1 L_2 = 4$  estados como ilustrado na Figura 5.1.

## 5.2 Arranjos de Treliça e Subarranjos

Suponha um par de códigos convolucionais  $(\mathcal{C}_1, \mathcal{C}_2)$  com comprimento de bloco  $n$ . Os codificadores para  $\mathcal{C}_1$  e  $\mathcal{C}_2$  têm  $L_1$  e  $L_2$  estados respectivamente. Geralmente um codificador convolucional tem seu estado inicial nulo e após a geração de  $N$  sub-blocos, a seqüência de sub-blocos é truncada ou finalizada. Em ambos os casos, o código resultante pode ser tratado como um código de bloco.

Seja  $a(s_i, s_p) \in \mathcal{C}_1$  a seqüência binária de  $N$  sub-blocos começando no estado  $s_i$ ,  $i = 1, 2, \dots, L_1$ , e terminando no estado  $s_p$ ,  $p = 1, \dots, L_1$ , consistindo da concatenação de  $N$  sub-blocos de  $\mathcal{C}_1$ . Similarmente, seja  $b(s_r, s_v) \in \mathcal{C}_2$  a seqüência binária de  $N$  sub-blocos começando

no estado  $s_r$ ,  $r = 1, 2, \dots, L_2$  e terminando no estado  $s_v$ ,  $v = 1, \dots, L_2$ , consistindo da concatenação de  $N$  sub-blocos de  $\mathcal{C}_2$ . Casos para os quais  $a(s_i, s_p)$  ou  $b(s_r, s_v)$  não são únicos resultariam em ramos paralelos nas respectivas treliças e não serão considerados na tese.

Denomina-se *Arranjo de treliça*  $\mathcal{A}$  [57] [58] [59] de um dado par de códigos convolucionais  $(\mathcal{C}_1, \mathcal{C}_2)$ , um arranjo de linhas e colunas, em que o elemento de cada célula em  $\mathcal{A}$  é uma seqüência ternária  $\mathbf{z}$ , representando a soma aritmética das duas seqüências binárias  $a(s_i, s_p)$  e  $b(s_r, s_v)$ , isto é,  $\mathbf{z} = a(s_i, s_p) + b(s_r, s_v)$ , em que  $a(s_i, s_p)$  e  $b(s_r, s_v)$  indicam, respectivamente, o índice da linha e o índice da coluna, e  $i = 1, 2, \dots, L_1$ ,  $p = 1, 2, \dots, L_1$ ,  $r = 1, 2, \dots, L_2$ , e  $v = 1, 2, \dots, L_2$ . Se  $i$  for fixo em  $a(s_i, s_p)$  (o índice da linha de  $\mathcal{A}$ ) e  $r$  for fixo em  $b(s_r, s_v)$  (o índice da coluna de  $\mathcal{A}$ ), e  $p = 1, 2, \dots, L_1$  e  $v = 1, 2, \dots, L_2$ , obtém-se o sub-arranjo  $S(s_i, s_r)$ . Ilustra-se o conceito de arranjo de treliça e sub-arranjo de treliça com os exemplos a seguir.

**Exemplo 5.2.1** *Seja  $\mathcal{C}$  um código convolucional recursivo sistemático com taxa assintótica 2/3 e matriz geradora polinomial*

$$G(D) = \begin{bmatrix} 1 & 0 & \frac{D}{1+D} \\ 0 & 1 & \frac{1}{1+D} \end{bmatrix}. \quad (5.2)$$

*O objetivo é usar  $\mathcal{C}$  nas linhas e colunas para construir o arranjo de treliça  $\mathcal{A}$  ilustrado na Figura 5.2. Neste caso, as seqüências binárias  $a(s_i, s_p)$  e  $b(s_r, s_v)$  em que  $i = 1, 2, 3, 4$ ,  $p = 1, 2, 3, 4$ ,  $r = 1, 2, 3, 4$ , e  $v = 1, 2, 3, 4$ , são constituídas por apenas  $N = 1$  sub-bloco. Os sub-arranjos estão delimitados pelas linhas em destaque. Observe que em cada sub-arranjo de  $\mathcal{A}$  existem seqüências ternárias  $\mathbf{z}$  repetidas. Por exemplo, em  $S(s_1, s_1)$  tem-se  $a(s_1, s_4) + b(s_1, s_1) = a(s_1, s_3) + b(s_1, s_2) = a(s_1, s_2) + b(s_1, s_3) = a(s_1, s_1) + b(s_1, s_4) = 111$ . A treliça correspondente para dois usuários tem 16 estados e de cada estado saem 16 ramos.*

**Exemplo 5.2.2** *Seja  $\mathcal{C}$  um código convolucional recursivo sistemático com taxa assintótica 1/2 e matriz geradora polinomial:*

$$G(D) = \begin{bmatrix} 1 & \frac{1 + D^2}{1 + D + D^2} \end{bmatrix}. \quad (5.3)$$

*Pode-se usar  $\mathcal{C}$  nas linhas e colunas para construir o arranjo de treliça  $\mathcal{A}$  ilustrado na Figura 5.3. Neste caso, as seqüências binárias  $a(s_i, s_p)$  e  $b(s_r, s_v)$  em que  $i = 1, 2, 3, 4$ ,  $p = 1, 2, 3, 4$ ,  $r = 1, 2, 3, 4$ , e  $v = 1, 2, 3, 4$ , são constituídas por  $N = 2$  sub-blocos. Os sub-arranjos estão delimitados pelas linhas em destaque.*

Considerando os sub-arranjos de treliça provou-se a seguinte proposição [57] para códigos convolucionais.

		$b(s_1, s_1)$	$b(s_1, s_2)$	$b(s_1, s_3)$	$b(s_1, s_4)$	$b(s_2, s_1)$	$b(s_2, s_2)$	$b(s_2, s_3)$	$b(s_2, s_4)$	$b(s_3, s_1)$	$b(s_3, s_2)$	$b(s_3, s_3)$	$b(s_3, s_4)$	$b(s_4, s_1)$	$b(s_4, s_2)$	$b(s_4, s_3)$	$b(s_4, s_4)$
		000	101	010	111	100	001	110	011	011	110	001	100	111	010	101	000
$a(s_1, s_1)$	000	000	101	010	111	100	001	110	011	011	110	001	100	111	010	101	000
$a(s_1, s_2)$	101	101	202	111	212	201	102	211	112	112	211	102	201	212	111	202	101
$a(s_1, s_3)$	010	010	111	020	121	110	011	120	021	021	120	011	110	121	020	111	010
$a(s_1, s_4)$	111	111	212	121	222	211	112	221	122	122	221	112	211	222	121	212	111
$a(s_2, s_1)$	100	100	201	110	211	200	101	210	111	111	210	101	200	211	110	201	100
$a(s_2, s_2)$	001	001	102	011	112	101	002	111	012	012	111	002	101	112	011	102	001
$a(s_2, s_3)$	110	110	211	120	221	210	111	220	121	121	220	111	210	221	120	211	110
$a(s_2, s_4)$	011	011	112	021	122	111	012	121	022	022	121	012	111	122	021	112	011
$a(s_3, s_1)$	011	011	112	021	122	111	012	121	022	022	121	012	111	122	021	112	011
$a(s_3, s_2)$	110	110	211	120	221	210	111	220	121	121	220	111	210	221	120	211	110
$a(s_3, s_3)$	001	001	102	011	112	101	002	111	012	012	111	002	101	112	011	102	001
$a(s_3, s_4)$	100	100	201	110	211	200	101	210	111	111	210	101	200	211	110	201	100
$a(s_4, s_1)$	111	111	212	121	222	211	112	221	122	122	221	112	211	222	121	212	111
$a(s_4, s_2)$	010	010	111	020	121	110	011	120	021	021	120	011	110	121	020	111	010
$a(s_4, s_3)$	101	101	202	111	212	201	102	211	112	112	211	102	201	212	111	202	101
$a(s_4, s_4)$	000	000	101	010	111	100	001	110	011	011	110	001	100	111	010	101	000

Figura 5.2: Arranjo de treliça para o par de códigos com matriz geradora polinomial dada em (5.2) e seqüências binárias  $a(s_i, s_p)$  e  $b(s_r, s_v)$  constituídas por apenas  $N = 1$  sub-bloco.

		$b(s_1, s_1)$	$b(s_1, s_2)$	$b(s_1, s_3)$	$b(s_1, s_4)$	$b(s_2, s_1)$	$b(s_2, s_2)$	$b(s_2, s_3)$	$b(s_2, s_4)$	$b(s_3, s_1)$	$b(s_3, s_2)$	$b(s_3, s_3)$	$b(s_3, s_4)$	$b(s_4, s_1)$	$b(s_4, s_2)$	$b(s_4, s_3)$	$b(s_4, s_4)$
		0000	1110	0011	1101	1100	0010	1111	0001	1011	0101	1000	0110	0111	1001	0100	1010
$a(s_1, s_1)$	0000	0000	1110	0011	1101	1100	0010	1111	0001	1011	0101	1000	0110	0111	1001	0100	1010
$a(s_1, s_2)$	1110	1110	2220	1121	2211	2210	1120	2221	1111	2121	1211	2110	1220	1221	2111	1210	2120
$a(s_1, s_3)$	0011	0011	1121	0022	1112	1111	0021	1122	0012	1022	0112	1011	0121	0122	1012	0111	1021
$a(s_1, s_4)$	1101	1101	2211	1112	2202	2201	1111	2212	1102	2112	1202	2101	1211	1212	2102	1201	2111
$a(s_2, s_1)$	1100	1100	2210	1111	2201	2200	1110	2211	1101	2111	1201	2100	1210	1211	2101	1200	2110
$a(s_2, s_2)$	0010	0010	1120	0021	1111	1110	0020	1121	0011	1021	0111	1010	0120	0121	1011	0110	1020
$a(s_2, s_3)$	1111	1111	2221	1122	2212	2211	1121	2222	1112	2122	1212	2111	1221	1222	2112	1211	2121
$a(s_2, s_4)$	0001	0001	1111	0012	1102	1101	0011	1112	0002	1012	0102	1001	0111	0112	1002	0101	1011
$a(s_3, s_1)$	1011	1011	2121	1022	2112	2111	1021	2122	1012	2022	1112	2011	1121	1122	2012	1111	2021
$a(s_3, s_2)$	0101	0101	1211	0112	1202	1201	0111	1212	0102	1112	0202	1101	0211	0212	1102	0201	1111
$a(s_3, s_3)$	1000	1000	2110	1011	2101	2100	1010	2111	1001	2011	1101	2000	1110	1111	2001	1100	2010
$a(s_3, s_4)$	0110	0110	1220	0121	1211	1210	0120	1221	0111	1121	0211	1110	0220	0221	1111	0210	1120
$a(s_4, s_1)$	0111	0111	1221	0122	1212	1211	0121	1222	0112	1122	0212	1111	0221	0222	1112	0211	1121
$a(s_4, s_2)$	1001	1001	2111	1012	2102	2101	1011	2112	1002	2012	1102	2001	1111	1112	2002	1101	2011
$a(s_4, s_3)$	0100	0100	1210	0111	1201	1200	0110	1211	0101	1111	0201	1100	0210	0211	1101	0200	1110
$a(s_4, s_4)$	1010	1010	2120	1021	2111	2110	1020	2121	1011	2021	1111	2010	1120	1121	2011	1110	2020

Figura 5.3: Arranjo de treliça para o par de códigos com matriz geradora polinomial dada em (5.3) e seqüências binárias  $a(s_i, s_p)$  e  $b(s_r, s_v)$  constituídas pela concatenação de  $N = 2$  sub-blocos.

**Proposição 5.2.1** *Um par de códigos de treliça é unicamente decodificável no 2-BAC se, e somente se, em cada sub-arranjo  $S(s_i, s_r), i = 1, 2, \dots, L_1$  e  $r = 1, 2, \dots, L_2$ , não existir seqüências ternárias repetidas.*

A prova segue primeiramente mostrando que, contrariando a hipótese, se a seqüência ternária  $\mathbf{z}$  ocorre mais que uma vez em  $S(s_i, s_r)$  tem-se que,  $\mathbf{z} = a(s_i, s_p) + b(s_r, s_v) = a(s_i, s_{p'}) + b(s_r, s_{v'}), p \neq p', v \neq v'$ . Como o par de estados iniciais é comum a ambas as células no sub-arranjo, não é possível para o decodificador resolver a ambigüidade de entregar  $a(s_i, s_p)$  associado ao usuário 1 e  $b(s_r, s_v)$  associado ao usuário 2, ou entregar  $a(s_i, s_{p'})$  associado ao usuário 1 e  $b(s_r, s_{v'})$  associado ao usuário 2. Supondo agora que a seqüência ternária  $\mathbf{z}$  ocorre apenas uma vez por sub-arranjo mas aparece mais de uma vez no arranjo de treliça, segue que  $\mathbf{z} \in S(s_i, s_t)$  e  $\mathbf{z} \in S(s_{i'}, s_{t'}), i \neq i'$  ou/e  $t \neq t'$  ocorrem. Nesta situação, contudo, o decodificador sabe *a priori* o par de estados iniciais dos códigos binários componentes, por exemplo o par  $(s_i, s_t)$  com  $s_i$  de  $\mathcal{C}_1$  e  $s_t$  de  $\mathcal{C}_2$ , e sem ambigüidade separa as palavras-código dos dois usuários.

### 5.3 Código de Treliça para o 2-BAC

Nesta secção é mostrado um método de construção de códigos de treliça a partir de uma concatenação serial de um par de códigos unicamente decodificáveis para o 2-BAC e de um par de códigos convolucionais.

Considere um par de códigos convolucionais  $(\mathcal{C}_1, \mathcal{C}_2)$  recursivos e sistemáticos com taxas assintóticas iguais a  $k/n$ , com memórias iguais a  $m_1$  e  $m_2$  respectivamente [2, p.303-308]. Seja  $(\mathbf{C}_1, \mathbf{C}_2)$  um par de códigos de bloco unicamente decodificáveis para o 2-BAC.

O código para o usuário 1 é construído a partir de  $\mathcal{C}_1$  e de  $\mathbf{C}_1$  do seguinte modo. O usuário 1 envia suas mensagens para o codificador de  $\mathcal{C}_1$ , e as palavras código resultantes de  $\mathcal{C}_1$  são enviadas como mensagens para o codificador de  $\mathbf{C}_1$ . Desta forma o usuário 1 estará se servindo de um dicionário contendo um subconjunto das palavras código de  $\mathcal{C}_1$ , escolhidas de acordo com as “mensagens” alimentadas por  $\mathbf{C}_1$  ao codificador de  $\mathcal{C}_1$ . A codificação para o usuário 2 é semelhante, empregando o código  $\mathcal{C}_2$  e o código  $\mathbf{C}_2$ . Conseqüentemente, o usuário 2 estará se servindo de um dicionário contendo um subconjunto das palavras código de  $\mathcal{C}_2$ , escolhidas de acordo com as “mensagens” alimentadas por  $\mathbf{C}_2$  ao codificador de  $\mathcal{C}_2$ .

Essencialmente a operação de codificação desempenhada por cada usuário é uma concatenação em série dos seus respectivos códigos de bloco com o código convolucional, conforme

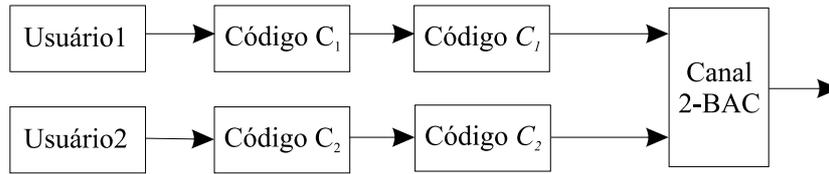


Figura 5.4: Modelo de construção de código unicamente decodificável para o 2-BAC.

ilustrado na Figura 5.4.

Como  $\mathcal{C}_1$  e  $\mathcal{C}_2$  são sistemáticos e possuem taxas assintóticas iguais, a soma aritmética bit a bit das palavras código de  $\mathcal{C}_1$  produzidas pelo usuário 1 e das palavras código de  $\mathcal{C}_2$  produzidas pelo usuário 2, é unicamente decodificável. Esta afirmação procede porque o par  $(\mathcal{C}_1, \mathcal{C}_2)$  é unicamente decodificável para o 2-BAC e a soma aritmética das palavras-código de  $\mathcal{C}_1$  e  $\mathcal{C}_2$  aparecem na seção de informação da soma aritmética das palavras código de  $\mathcal{C}_1$  e  $\mathcal{C}_2$ .

O uso dos códigos  $\mathcal{C}_1$  e  $\mathcal{C}_2$  leva à eliminação de alguns ramos e, algumas vezes, leva à eliminação de alguns estados na treliça para dois usuários. Em outras palavras, serão eliminados caminhos nas treliças dos códigos convolucionais empregados, evitando assim problemas de ambigüidade na decodificação.

Se a taxa do par  $(\mathcal{C}_1, \mathcal{C}_2)$  é  $R$  segue desta construção que  $R_C = (\frac{k}{n})R$  é a taxa do código construído para o 2-BAC. Portanto, se  $\mathcal{C}_1$  e  $\mathcal{C}_2$  forem escolhidos para serem códigos com taxas assintóticas aproximadamente igual a 1,  $R_C$  terá um valor muito próximo a  $R$ . Isto significa que se  $R$  alcançar o valor máximo da taxa de transmissão (capacidade) para o 2-BAC, então  $R_C$  também alcançará este valor.

**Exemplo 5.3.1** Considere que  $\mathcal{C} = \mathcal{C}_1 = \mathcal{C}_2$  é o código convolucional recursivo com taxa assintótica  $1/2$  e matriz geradora polinomial dada em (5.1) e que  $\mathcal{C}_1 = \{00, 11\}$  e  $\mathcal{C}_2 = \{00, 01, 10\}$ . A treliça resultante da concatenação em série de  $\mathcal{C}$  e de  $\mathcal{C}_1$  está ilustrada na Figura 5.5. A treliça resultante da concatenação de  $\mathcal{C}$  e de  $\mathcal{C}_2$  está ilustrada na Figura 5.6. É importante observar que o ramo de saída rotulado (1/11) que parte do estado (00) no instante de tempo  $t = 3$  só existe, caso o ramo de saída em  $t = 2$  seja o rotulado com (0/00). Da mesma forma, o ramo de saída rotulado (1/10) que parte do estado (01) no instante de tempo  $t = 3$  só existe, caso o ramo de saída em  $t = 2$  seja o rotulado com (0/01). A treliça resultante para dois usuários está ilustrada na Figura 5.7. Verifica-se, nas Figuras 5.5 e 5.6, que vários ramos na treliça (completa) de  $\mathcal{C}$  foram eliminados, assim como também foram eliminados alguns estados. O código para o 2-BAC assim construído é unicamente decodificável com taxa 0,645.

**Exemplo 5.3.2** Considere  $\mathcal{C}$  o código convolucional recursivo e sistemático com taxa assintótica



		$b(s_1, s_1)$	$b(s_1, s_4)$	$b(s_2, s_2)$	$b(s_2, s_3)$	$b(s_3, s_2)$	$b(s_3, s_3)$	$b(s_4, s_1)$	$b(s_4, s_4)$
		000	111	001	110	110	001	111	000
$a(s_1, s_1)$	000	000	111	001	110	110	001	111	000
$a(s_1, s_2)$	101	101	212	102	211	211	102	212	101
$a(s_1, s_3)$	010	010	121	011	120	120	011	121	010
$a(s_2, s_1)$	100	100	211	101	210	210	101	211	100
$a(s_2, s_2)$	001	001	112	002	111	111	002	112	001
$a(s_2, s_4)$	011	011	122	012	121	121	012	122	011
$a(s_3, s_1)$	011	011	122	012	121	121	012	122	011
$a(s_3, s_3)$	001	001	112	002	111	111	002	112	001
$a(s_3, s_4)$	100	100	211	101	210	210	101	211	100
$a(s_4, s_2)$	010	010	121	011	120	120	011	121	010
$a(s_4, s_3)$	101	101	212	102	211	211	102	212	101
$a(s_4, s_4)$	000	000	111	001	110	110	001	111	000

Figura 5.8: Arranjo de treliça perfurado, encontrado após eliminação de algumas linhas e colunas do arranjo de treliça ilustrado na Figura 5.2, devido ao uso da concatenação serial ilustrada em 5.4, tendo  $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$  dado em (5.2) e  $\mathbf{C}_1 = \{00, 11\}$  e  $\mathbf{C}_2 = \{00, 01, 10\}$ .

$2/3$  e matriz geradora polinomial dada em (5.2). Ao utilizar  $\mathcal{C}$  no esquema de concatenação serial descrito nesta seção e assumindo  $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$ , o uso dos códigos  $\mathbf{C}_1 = \{00, 11\}$  e  $\mathbf{C}_2 = \{00, 01, 10\}$ , leva a eliminação de algumas linhas e algumas colunas no arranjo de treliça  $\mathcal{A}$  mostrado na Figura 5.2, e a eliminação de alguns estados e ramos na treliça para dois usuários. Obtém-se então o arranjo de treliça perfurado mostrado na Figura 5.8 e a treliça para dois usuários mostrada na Figura 5.9, correspondendo a um par de códigos de treliça unicamente decodificáveis. Observe agora que em cada sub-arranjo (Figura 5.8) todas as células possuem conteúdos distintos.

**Exemplo 5.3.3** Considere  $\mathcal{C}$  o código convolucional recursivo e sistemático com taxa assintótica  $1/2$  e matriz geradora polinomial dada em (5.3), se for utilizado  $\mathcal{C}$  no esquema de concatenação serial descrito nesta seção e assumindo  $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$ , o uso dos códigos  $\mathbf{C}_1 = \{00, 11\}$  e  $\mathbf{C}_2 = \{00, 01, 10\}$ , fornece o arranjo de treliça perfurado mostrado na Figura 5.10. Observe que em cada sub-arranjo todas as células possuem conteúdos distintos, resultando na decodibilidade única.

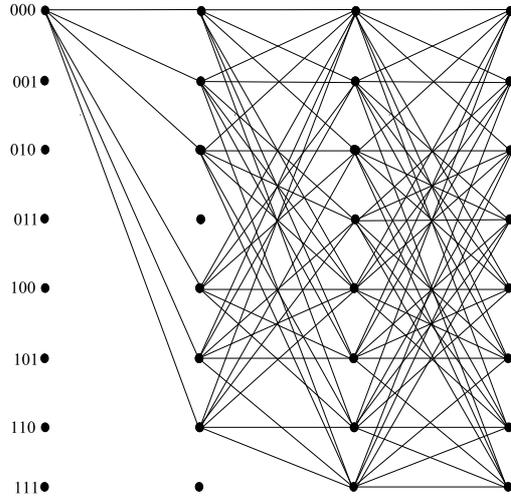


Figura 5.9: Treliça para dois usuários após o uso do esquema de concatenação serial, tendo  $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$  dado em (5.2) e  $\mathbf{C}_1 = \{00, 11\}$  e  $\mathbf{C}_2 = \{00, 01, 10\}$ .

		$b(s_1, s_1)$	$b(s_1, s_3)$	$b(s_1, s_4)$	$b(s_2, s_1)$	$b(s_2, s_2)$	$b(s_2, s_4)$	$b(s_3, s_2)$	$b(s_3, s_3)$	$b(s_3, s_4)$	$b(s_4, s_1)$	$b(s_4, s_2)$	$b(s_4, s_3)$
		0000	0011	1101	1100	0010	0001	0101	1000	0110	0111	1001	0100
$a(s_1, s_1)$	0000	0000	0011	1101	1100	0010	0001	0101	1000	0110	0111	1001	0100
$a(s_1, s_2)$	1110	1110	1121	2211	2210	1120	1111	1211	2110	1220	1221	2111	1210
$a(s_2, s_3)$	1111	1111	1122	2212	2211	1121	1112	1212	2111	1221	1222	2112	1211
$a(s_2, s_4)$	0001	0001	0012	1102	1101	0011	0002	0102	1001	0111	0112	1002	0101
$a(s_3, s_1)$	1011	1011	1022	2112	2111	1021	1012	1112	2011	1121	1122	2012	1111
$a(s_3, s_2)$	0101	0101	0112	1202	1201	0111	0102	0202	1101	0211	0212	1102	0201
$a(s_4, s_3)$	0100	0100	0111	1201	1200	0110	0101	0201	1100	0210	0211	1101	0200
$a(s_4, s_4)$	1010	1010	1021	2111	2110	1020	1011	1111	2010	1120	1121	2011	1110

Figura 5.10: Arranjo de treliça perfurado, encontrado após eliminação de algumas linhas e colunas do arranjo de treliça ilustrado na Figura 5.3, devido ao uso do esquema de concatenação serial (Figura 5.4), tendo  $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$  dado em (5.3) e  $\mathbf{C}_1 = \{00, 11\}$  e  $\mathbf{C}_2 = \{00, 01, 10\}$ .

# Capítulo 6

## Decodificação Iterativa para o Canal Aditivo com Dois Usuários Binários Usando Códigos de Treliça

Nesta capítulo, é descrito um esquema de decodificação iterativa, que pode ser utilizado quando os codificadores convolucionais ilustrados na Figura 5.4, Seção 5.3 são substituídos por codificadores turbo.

### 6.1 O Codificador

Considere o modelo de construção de código unicamente decodificável para o 2-BAC introduzido na Seção 5.3 e ilustrado na Figura 5.4. Considere que, a partir de agora, o codificador para  $\mathcal{C}_1$ , utiliza o esquema de concatenação paralela introduzido em [39] [52] e revisado na Seção 4.2. Desta forma, o codificador para  $\mathcal{C}_1$  é formado pela concatenação paralela de dois códigos convolucionais recursivos componentes,  $\mathcal{C}_1^-$  e  $\mathcal{C}_1^|$ , não necessariamente iguais. As entradas de ambos os codificadores componentes utilizam os mesmos bits de informação  $u_k$ , mas em uma ordem diferente, devido à presença do entrelaçador. Similarmente, o codificador para  $\mathcal{C}_2$  é formado pela concatenação paralela de dois códigos convolucionais recursivos componentes,  $\mathcal{C}_2^-$  e  $\mathcal{C}_2^|$ , não necessariamente iguais. As entradas de ambos os codificadores componentes utilizam os mesmos bits de informação  $d_k$ , mas em uma ordem diferente, devido à presença do entrelaçador, que deve ser idêntico ao entrelaçador utilizado para  $\mathcal{C}_1$ . A taxa de transmissão de  $\mathcal{C}_1$  deve ser igual à taxa de transmissão de  $\mathcal{C}_2$ . Nas Figuras 6.1 e 6.2 estão ilustrados os

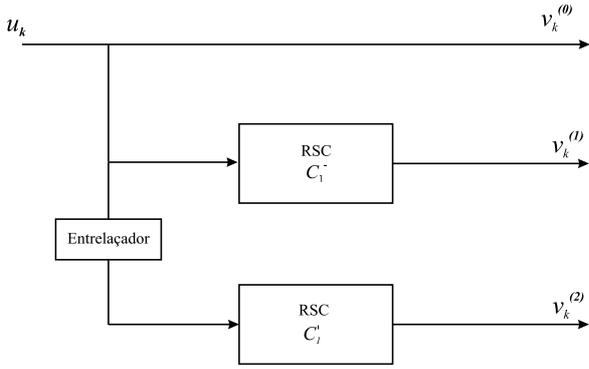


Figura 6.1: Esquema de codificação paralela para o codificador de  $\mathcal{C}_1$ .

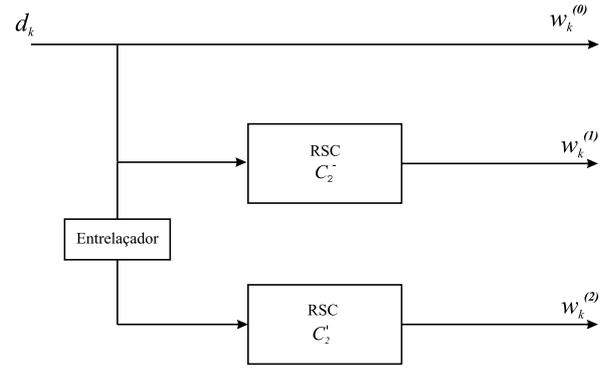


Figura 6.2: Esquema de codificação paralela para o codificador de  $\mathcal{C}_2$ .

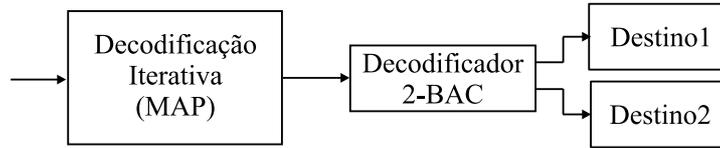


Figura 6.3: O decodificador empregado utiliza a decodificação iterativa para estimar a seqüência ternária mais provável e em seguida usa o decodificador 2-BAC para separar a informação relativa ao usuário 1 e ao usuário 2.

codificadores de  $\mathcal{C}_1$  e  $\mathcal{C}_2$  respectivamente, para códigos convolucionais constituintes com taxa  $1/2$ .

## 6.2 O Decodificador

O decodificador utilizado (Figura 6.3) para o esquema de codificação introduzido na seção anterior, utiliza a decodificação iterativa para estimar a seqüência ternária mais provável e em seguida usa o decodificador 2-BAC para separar a informação relativa ao usuário 1 e ao usuário 2, utilizando para isto o par de blocos  $(\mathcal{C}_1, \mathcal{C}_2)$  unicamente decodificáveis para o 2-BAC.

### 6.2.1 Algoritmo BCJR para Dois Usuários

Considere, sem perda de generalidade, que cada codificador recursivo sistemático tem taxa de transmissão assintótica  $1/n$  e  $M$  estados, para ambos usuários. Tem-se que as seqüências de símbolos de informação para o usuário 1 e usuário 2 são representadas respectivamente por:

$$\mathbf{u} = \mathbf{u}_1^N = \{u_1, u_2, \dots, u_k, \dots, u_N\},$$

$$\mathbf{d} = \mathbf{d}_1^N = \{d_1, d_2, \dots, d_k, \dots, d_N\}.$$

As seqüências código associadas para o usuário 1 e usuário 2 são representadas respectivamente por:

$$\mathbf{v} = \mathbf{v}_1^N = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k, \dots, \mathbf{v}_N\},$$

$$\mathbf{w} = \mathbf{w}_1^N = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k, \dots, \mathbf{w}_N\}.$$

em que  $\mathbf{v}_k = (v_k^{(0)}, v_k^{(1)}, \dots, v_k^{(n-1)}) = (u_k, v_k^{(2)}, \dots, v_k^{(n-1)})$ , é a saída associada a cada símbolo de informação do usuário 1 e similarmente,  $\mathbf{w}_k = (w_k^{(0)}, w_k^{(1)}, \dots, w_k^{(n-1)}) = (d_k, w_k^{(1)}, \dots, w_k^{(n-1)})$ , é a saída associada a cada símbolo de informação do usuário 2.  $v_k^{(0)}$  e  $w_k^{(0)}$  são as saídas sistemáticas dos codificadores para o usuário 1 e usuário 2, respectivamente. Os termos  $\mathbf{v}_1^N$  e  $\mathbf{w}_1^N$  são as entradas para um canal aditivo ruidoso com dois usuários binários, sem memória. O ruído aqui considerado é o ruído branco gaussiano. A seqüência de sub-blocos na treliça para dois usuários é dada por

$$\mathbf{x} = \mathbf{x}_1^N = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k, \dots, \mathbf{x}_N\},$$

em que  $\mathbf{x}_k = (x_k^{(0)}, x_k^{(1)}, \dots, x_k^{(n-1)})$ . A variável aleatória  $x_k^{(j)}$   $j = 0, \dots, n-1$ , no instante de tempo  $k$ , é definida por meio da seguinte igualdade

$$x_k^{(j)} = (2v_k^{(j)} - 1) + (2w_k^{(j)} - 1) \quad j = 0, \dots, n-1. \quad (6.1)$$

A saída do canal é a seqüência recebida

$$\mathbf{r} = \mathbf{r}_1^N = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k, \dots, \mathbf{r}_N\},$$

em que  $\mathbf{r}_k = (r_k^{(0)}, r_k^{(1)}, \dots, r_k^{(n-1)})$ . A variável aleatória  $r_k^{(j)}$   $j = 0, \dots, n-1$ , no instante de tempo  $k$ , é definida pela seguinte igualdade

$$r_k^{(j)} = x_k^{(j)} + q_k^{(j)} \quad j = 0, \dots, n-1, \quad (6.2)$$

em que  $q_k^{(j)}$  são ruídos independentes com a mesma variância  $\sigma^2$  e média zero.

O algoritmo BCJR utiliza a treliça para dois usuários definida na Seção 5.1 e calcula as razões de log-verossimilhança  $\Lambda_1(u_k, d_k)$ ,  $\Lambda_2(u_k, d_k)$  e  $\Lambda_3(u_k, d_k)$  associadas ao par dos símbolos de informação  $(u_k, d_k)$ , relativos ao usuário 1 e ao usuário 2, respectivamente.

$$\Lambda_1(u_k, d_k) = \log \frac{\mathrm{P}\{u_k = 1, d_k = 0 | \mathbf{r}\}}{\mathrm{P}\{u_k = 0, d_k = 0 | \mathbf{r}\}}, \quad (6.3)$$

$$\Lambda_2(u_k, d_k) = \log \frac{\mathrm{P}\{u_k = 1, d_k = 1 | \mathbf{r}\}}{\mathrm{P}\{u_k = 0, d_k = 0 | \mathbf{r}\}}, \quad (6.4)$$

$$\Lambda_3(u_k, d_k) = \log \frac{\mathrm{P}\{u_k = 0, d_k = 1 | \mathbf{r}\}}{\mathrm{P}\{u_k = 0, d_k = 0 | \mathbf{r}\}}, \quad (6.5)$$

em que  $P\{u_k = i, d_k = s|\mathbf{r}\}$ ,  $i = 0, 1$ ,  $s = 0, 1$  é a probabilidade *a posteriori* do par dos símbolos de informação  $u_k, d_k$ .

Considerando  $1 \leq k \leq N$ , em que  $N$  é o comprimento da seqüência recebida, o decodificador poderá tomar as seguintes decisões:

- $u_k = 0, d_k = 0$ ,  
se  $P\{u_k = 0, d_k = 0|\mathbf{r}\} \geq P\{u_k = i, d_k = s|\mathbf{r}\}$ ,  $i \neq 0, s \neq 0$ ;
- $u_k = 0, d_k = 1$ ,  
se  $P\{u_k = 0, d_k = 1|\mathbf{r}\} \geq P\{u_k = i, d_k = s|\mathbf{r}\}$ ,  $i \neq 0, s \neq 1$ ;
- $u_k = 1, d_k = 0$ ,  
se  $P\{u_k = 1, d_k = 0|\mathbf{r}\} \geq P\{u_k = i, d_k = s|\mathbf{r}\}$ ,  $i \neq 1, s \neq 0$ ;
- $u_k = 1, d_k = 1$ ,  
Se  $P\{u_k = 1, d_k = 1|\mathbf{r}\} \geq P\{u_k = i, d_k = s|\mathbf{r}\}$ ,  $i \neq 1, s \neq 1$ .

O estado da treliça para dois usuários no  $k$ -ésimo intervalo de tempo é dado por  $S_k$ . Considere que a treliça para dois usuários possui um total de  $M$  estados. A probabilidade *a posteriori* de cada par dos símbolos de informação dos dois usuários pode ser extraída da probabilidade conjunta  $\lambda_k^{i,s}(m)$  definida por:

$$\begin{aligned} \lambda_k^{i,s}(m) &= P\{u_k = i, d_k = s, S_k = m|\mathbf{r}_1^N\} \\ &= \frac{p\{u_k = i, d_k = s, S_k = m, \mathbf{r}_1^N\}}{p\{\mathbf{r}_1^N\}}. \end{aligned} \quad (6.6)$$

A probabilidade *a posteriori* dos pares dos símbolos de informação  $u_k$  e  $d_k$  é:

$$P\{u_k = i, d_k = s|\mathbf{r}_1^N\} = \sum_{m=0}^{M-1} \lambda_k^{i,s}(m), \quad i = 0, 1 \quad s = 0, 1.$$

As relações (6.3), (6.4) e (6.5) portanto, podem ser escritas da seguinte forma:

$$\Lambda_1(u_k, d_k) = \log \frac{\sum_{m=0}^{M-1} \lambda_k^{1,0}(m)}{\sum_{m=0}^{M-1} \lambda_k^{0,0}(m)}, \quad (6.7)$$

$$\Lambda_2(u_k, d_k) = \log \frac{\sum_{m=0}^{M-1} \lambda_k^{1,1}(m)}{\sum_{m=0}^{M-1} \lambda_k^{0,0}(m)}, \quad (6.8)$$

$$\Lambda_3(u_k, d_k) = \log \frac{\sum_{m=0}^{M-1} \lambda_k^{0,1}(m)}{\sum_{m=0}^{M-1} \lambda_k^{0,0}(m)}. \quad (6.9)$$

De (6.6), (6.7), (6.8) e (6.9) segue que:

$$\Lambda_1(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \mathbb{P}\{u_k = 1, d_k = 0, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}{\sum_m \sum_{m'} \mathbb{P}\{u_k = 0, d_k = 0, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}, \quad (6.10)$$

$$\Lambda_2(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \mathbb{P}\{u_k = 1, d_k = 1, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}{\sum_m \sum_{m'} \mathbb{P}\{u_k = 0, d_k = 0, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}, \quad (6.11)$$

$$\Lambda_3(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \mathbb{P}\{u_k = 0, d_k = 1, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}{\sum_m \sum_{m'} \mathbb{P}\{u_k = 0, d_k = 0, S_k = m, S_{k-1} = m', \mathbf{r}_1^{k-1}, \mathbf{r}_k, \mathbf{r}_{k+1}^N\}}. \quad (6.12)$$

Levando em consideração que eventos depois do instante de tempo  $k$  não são influenciados pela observação  $\mathbf{r}_1^k$  e pelos bits  $u_k$  e  $d_k$  se o estado  $S_k$  é conhecido tem-se que:

$$\Lambda_1(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\} \mathbb{P}\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} \mathbb{P}\{u_k = 1, d_k = 0, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}}{\sum_m \sum_{m'} \mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\} \mathbb{P}\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} \mathbb{P}\{u_k = 0, d_k = 0, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}},$$

$$\Lambda_2(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\} \mathbb{P}\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} \mathbb{P}\{u_k = 1, d_k = 1, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}}{\sum_m \sum_{m'} \mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\} \mathbb{P}\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} \mathbb{P}\{u_k = 0, d_k = 0, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}},$$

$$\Lambda_3(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\} \mathbb{P}\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} \mathbb{P}\{u_k = 0, d_k = 1, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}}{\sum_m \sum_{m'} \mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\} \mathbb{P}\{S_{k-1} = m', \mathbf{r}_1^{k-1}\} \mathbb{P}\{u_k = 0, d_k = 0, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}}.$$

Introduzindo as funções de probabilidade:

$$\alpha_k(m) = \mathbb{P}\{S_k = m | \mathbf{r}_1^k\}, \quad (6.13)$$

$$\beta_k(m) = \frac{\mathbb{P}\{\mathbf{r}_{k+1}^N | S_k = m\}}{\mathbb{P}\{\mathbf{r}_{k+1}^N | \mathbf{r}_1^k\}}, \quad (6.14)$$

$$\gamma_{i,s}(\mathbf{r}_k, m', m) = \mathbb{P}\{u_k = i, d_k = s, S_k = m, \mathbf{r}_k | S_{k-1} = m'\}, \quad (6.15)$$

tem-se que:

$$\Lambda_1(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \gamma_{1,0}(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \gamma_{0,0}(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}, \quad (6.16)$$

$$\Lambda_2(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \gamma_{1,1}(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \gamma_{0,0}(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}, \quad (6.17)$$

$$\Lambda_3(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \gamma_{0,1}(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \gamma_{0,0}(\mathbf{r}_k, m', m) \alpha_{k-1}(m') \beta_k(m)}. \quad (6.18)$$

em que  $\alpha_k(m)$ , para  $k = 1, 2, \dots, N$ , pode ser calculado, similarmente como para (3.44), por meio de

$$\alpha_k(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_{i,s}(\mathbf{r}_k, m', m) \alpha_{k-1}(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \gamma_{i,s}(\mathbf{r}_k, m', m) \alpha_{k-1}(m')}. \quad (6.19)$$

Considerando que a treliça é inicializada no estado  $S_0 = \mathbf{0}$ , então as condições de contorno são as seguintes:

$$\alpha_0(\mathbf{0}) = 1, \text{ e } \alpha_0(m) = 0, \text{ para } m \neq \mathbf{0}. \quad (6.20)$$

O termo  $\beta_k(m)$ , para  $k = 1, 2, \dots, N-1$ , pode ser calculado, similarmente como para (3.49), por

$$\beta_k(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_{i,s}(\mathbf{r}_{k+1}, m, m') \beta_{k+1}(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \gamma_{i,s}(\mathbf{r}_{k+1}, m', m) \alpha_k(m')}. \quad (6.21)$$

As condições de contorno apropriadas quando a treliça para dois usuários é levada a terminar no estado  $S_N = \mathbf{0}$  são:

$$\beta_N(\mathbf{0}) = 1 \text{ e } \beta_N(m) = 0, \text{ para } m \neq 0. \quad (6.22)$$

Quando não se sabe o estado final da treliça para dois usuários, as condições de contorno apropriadas são:

$$\beta_N(\mathbf{0}) = 1/M \text{ e } \beta_N(m) = 0, \text{ para } m \neq 0. \quad (6.23)$$

(6.19) e (6.21) mostram que  $\alpha_k(m)$  e  $\beta_k(m)$  são calculados de maneira recursiva.

As probabilidades  $\gamma_{i,s}(\mathbf{r}_k, m', m)$  podem ser determinadas a partir das probabilidades de transição do canal aditivo com dois usuários binários contaminado com ruído branco gaussiano e das probabilidades de transição da treliça para dois usuários:

$$\begin{aligned} \gamma_{i,s}(\mathbf{r}_k, m', m) &= \text{P}\{S_k = m | S_{k-1} = m'\} \text{p}\{\mathbf{r}_k | u_k = i, d_k = s, S_k = m, S_{k-1} = m'\} \\ &\quad \text{P}\{u_k = i, d_k = s | S_k = m, S_{k-1} = m'\} \end{aligned} \quad (6.24)$$

As probabilidades de transição  $\text{P}\{S_k = m | S_{k-1} = m'\}$  são definidas pelas probabilidades *a priori* dos bits de entrada. Quando os bits de entrada são equiprováveis  $\text{P}\{u_k = 0, d_k = 0\} = 1/4$ ,  $\text{P}\{u_k = 1, d_k = 0\} = 1/4$ ,  $\text{P}\{u_k = 1, d_k = 1\} = 1/4$ ,  $\text{P}\{u_k = 0, d_k = 1\} = 1/4$  então  $\text{P}\{S_k = m | S_{k-1} = m'\} = 1/4$ . A probabilidade  $\text{P}\{\mathbf{r}_k | u_k = i, d_k = s, S_k = m, S_{k-1} = m'\}$  é a probabilidade de transição do canal aditivo com dois usuários binários contaminado com ruído branco gaussiano, podendo ser escrita da seguinte forma:

$$\text{p}\{\mathbf{r}_k | u_k = i, d_k = s, S_k = m, S_{k-1} = m'\} = \text{p}\{\mathbf{r}_k | \mathbf{x}_k\}, \quad (6.25)$$

em que,

$$\text{p}\{\mathbf{r}_k | \mathbf{x}_k\} = \prod_{j=0}^{n-1} \text{p}\{r_k^{(j)} | x_k^{(j)}\},$$

e

$$\text{p}\{r_k^{(j)} | x_k^{(j)}\} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_k^{(j)} - x_k^{(j)})^2}{2\sigma^2}\right), \quad (6.26)$$

assim,



Para simplicidade de notação façamos  $P\{u_k = i, d_k = s\} = p_k(i, s)$

$$\Lambda_1(u_k) = \log \frac{\sum_m \sum_{m'} p_k(1, 0) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - x_{k,1,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} p_k(0, 0) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - x_{k,0,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}, \quad (6.31)$$

$$\Lambda_2(u_k) = \log \frac{\sum_m \sum_{m'} p_k(1, 1) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - x_{k,1,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} p_k(0, 0) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - x_{k,0,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}, \quad (6.32)$$

$$\Lambda_3(u_k) = \log \frac{\sum_m \sum_{m'} p_k(0, 1) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - x_{k,0,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} p_k(0, 0) \exp\left(-\frac{\sum_{j=0}^{n-1} (r_k^{(j)} - x_{k,0,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}. \quad (6.33)$$

O algoritmo BCJR portanto segue da seguinte forma:

1. As condições iniciais de  $\alpha_0(m)$  e  $\beta_N(m)$  estão em (6.20) e (6.22) ou (6.23);
2. Quando  $\mathbf{r}_k$  é recebido, o decodificador calcula  $\gamma_{i,s}(\mathbf{r}_k, m', m)$  usando (6.27) e  $\alpha_k(m)$  usando (6.19). Os valores de  $\alpha_k(m)$  são armazenados para todo  $k$  e  $m$ ;
3. Depois que a seqüência completa  $\mathbf{r}_1^N$  é recebida, o decodificador calcula de forma recursiva  $\beta_k(m)$  usando (6.21). O valor de  $\beta_k(m)$  calculado pode ser multiplicado pelo  $\alpha_k(m)$  e  $\gamma_{i,s}(\mathbf{r}_k, m', m)$  apropriado para obtenção de (6.31), (6.32) e (6.33).

## 6.2.2 Decodificação Iterativa

O decodificador utiliza o princípio da decodificação iterativa [51] e consiste de dois decodificadores componentes concatenados em série como pode ser visto na Figura 6.4.

Na entrada do primeiro decodificador BCJR DEC1 tem-se as seqüências recebidas  $\mathbf{r}^{(0)} = \{r_1^{(0)}, r_2^{(0)}, \dots, r_N^{(0)}\}$  e  $\mathbf{r}^{(1)} = \{r_1^{(1)}, r_2^{(1)}, \dots, r_N^{(1)}\}$ , em que  $r_k^{(j)}$  foi definido em (6.2). DEC1 então produz as saídas suaves  $(\Lambda_{1,1}(u_k, d_k), \Lambda_{2,1}(u_k, d_k), \Lambda_{3,1}(u_k, d_k))$ , que são entrelaçadas e usadas para produzir estimativas das probabilidades *a priori* dos pares de seqüências de informação para o segundo decodificador BCJR DEC2. Introduziu-se a notação  $\Lambda_{1,1}(u_k, d_k)$ ,  $\Lambda_{2,1}(u_k, d_k)$ ,  $\Lambda_{3,1}(u_k, d_k)$  para as saídas suaves  $\Lambda_1(u_k, d_k)$ ,  $\Lambda_2(u_k, d_k)$  e  $\Lambda_3(u_k, d_k)$  respectivamente, associadas com DEC1.

Na entrada de DEC2 estão as seqüências recebidas  $\tilde{\mathbf{r}}^{(0)}$  e  $\mathbf{r}^{(2)} = \{r_1^{(2)}, r_2^{(2)}, \dots, r_N^{(2)}\}$ . A seqüência  $\tilde{\mathbf{r}}^{(0)}$  corresponde à seqüência  $\mathbf{r}^{(0)}$  entrelaçada. DEC2 também produz saídas suaves  $(\Lambda_{1,2}(u_k, d_k), \Lambda_{2,2}(u_k, d_k), \Lambda_{3,2}(u_k, d_k))$ . A notação indica que as saídas suaves  $\Lambda_1(u_k, d_k)$ ,  $\Lambda_2(u_k, d_k)$

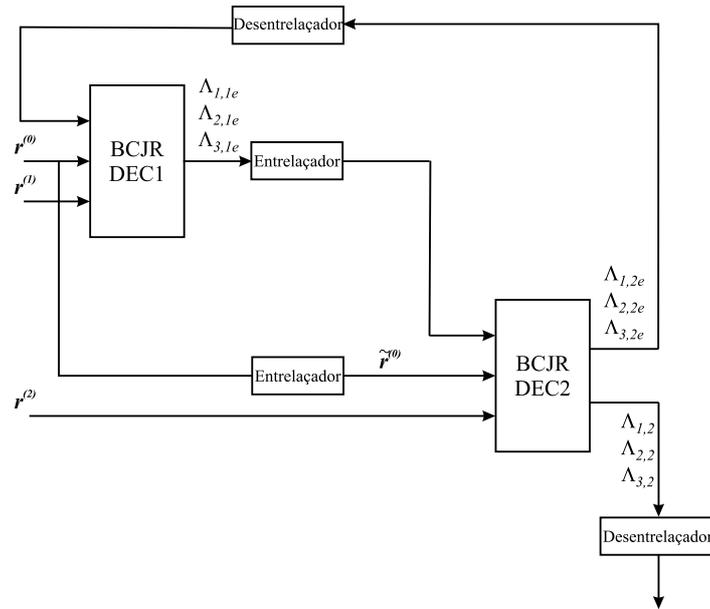


Figura 6.4: Decodificador turbo para dois usuários. O decodificador utiliza o princípio da decodificação iterativa e consiste de dois decodificadores componentes concatenados em série. É usado para estimar a seqüência ternária mais provável.

e  $\Lambda_3(u_k, d_k)$  estão associadas com DEC2. Estas saídas suaves são usadas para melhorar a estimativa das probabilidades a priori dos pares de seqüências  $(u_k, d_k)$  de informação na entrada de DEC1.

Como o código é sistemático  $x_{k,i,s}^{(0)}$ ,  $i = 0, 1$ ,  $s = 0, 1$  é independente dos códigos para dois usuários e dos estados  $m$  e  $m'$ . Assim, considerando  $x_{k,0,0}^{(0)} = -2$ ,  $x_{k,1,0}^{(0)} = 0$ ,  $x_{k,0,1}^{(0)} = 0$  e  $x_{k,1,1}^{(0)} = 2$ , tem-se:

$$\Lambda_{1,1}(u_k, d_k) = \log \frac{p_k^1(1, 0)}{p_k^1(0, 0)} + \frac{2r_k^{(0)} + 2}{\sigma^2} + \Lambda_{1,1e}(u_k, d_k), \quad (6.34)$$

$$\Lambda_{2,1}(u_k, d_k) = \log \frac{p_k^1(1, 1)}{p_k^1(0, 0)} + \frac{4}{\sigma^2} r_k^{(0)} + \Lambda_{2,1e}(u_k, d_k), \quad (6.35)$$

$$\Lambda_{3,1}(u_k, d_k) = \log \frac{p_k^1(0, 1)}{p_k^1(0, 0)} + \frac{2r_k^{(0)} + 2}{\sigma^2} + \Lambda_{3,1e}(u_k, d_k), \quad (6.36)$$

em que

$$\Lambda_{1,1e}(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - x_{k,1,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - x_{k,0,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}, \quad (6.37)$$

$$\Lambda_{2,1e}(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,1,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,0,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}, \quad (6.38)$$

$$\Lambda_{3,1e}(u_k, d_k) = \log \frac{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,0,1}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \exp\left(-\frac{\sum_{j=1}^{n-1} (r_k^{(j)} - v_{k,0,0}^{(j)})^2}{2\sigma^2}\right) \alpha_{k-1}(m') \beta_k(m)}. \quad (6.39)$$

Os valores  $\Lambda_{1,1e}(u_k, d_k)$ ,  $\Lambda_{2,1e}(u_k, d_k)$  e  $\Lambda_{3,1e}(u_k, d_k)$  são as informações extrínsecas e estão relacionados com as informações redundantes introduzidas pelo codificadores  $\mathcal{C}_1^-$  e  $\mathcal{C}_2^-$ . Eles não contêm a entrada de DEC1  $r_k^{(0)}$ , assim são usados para melhorar as estimativas das probabilidades *a priori* para o próximo estágio de decodificação, isto é, as informações extrínsecas de DEC1 entrelaçadas  $\tilde{\Lambda}_{1,1e}(u_k, d_k)$ ,  $\tilde{\Lambda}_{2,1e}(u_k, d_k)$  e  $\tilde{\Lambda}_{3,1e}(u_k, d_k)$  são as estimativas das probabilidades *a priori* para DEC2.

$$\tilde{\Lambda}_{1,1e}(u_k, d_k) = \log \frac{p_k^2(1, 0)}{p_k^2(0, 0)}, \quad (6.40)$$

$$\tilde{\Lambda}_{2,1e}(u_k, d_k) = \log \frac{p_k^2(1, 1)}{p_k^2(0, 0)}, \quad (6.41)$$

$$\tilde{\Lambda}_{3,1e}(u_k, d_k) = \log \frac{p_k^2(0, 1)}{p_k^2(0, 0)}. \quad (6.42)$$

Utilizando (6.40), (6.41), (6.42) e a igualdade

$$1 = p_k^2(0, 0) + p_k^2(1, 0) + p_k^2(1, 1) + p_k^2(0, 1),$$

pode-se escrever as probabilidades *a priori* para DEC2 da seguinte forma:

$$p_k^2(0, 0) = \frac{1}{1 + \exp \tilde{\Lambda}_{1,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{2,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{3,1e}(u_k, d_k)}, \quad (6.43)$$

$$p_k^2(1, 0) = \frac{\exp \tilde{\Lambda}_{1,1e}(u_k, d_k)}{1 + \exp \tilde{\Lambda}_{1,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{2,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{3,1e}(u_k, d_k)}, \quad (6.44)$$

$$p_k^2(1, 1) = \frac{\exp \tilde{\Lambda}_{2,1e}(u_k, d_k)}{1 + \exp \tilde{\Lambda}_{1,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{2,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{3,1e}(u_k, d_k)}, \quad (6.45)$$

$$p_k^2(0, 1) = \frac{\exp \tilde{\Lambda}_{3,1e}(u_k, d_k)}{1 + \exp \tilde{\Lambda}_{1,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{2,1e}(u_k, d_k) + \exp \tilde{\Lambda}_{3,1e}(u_k, d_k)}. \quad (6.46)$$

DEC2 estima as razões de log-verossimilhança  $\Lambda_{1,2}(u_k, d_k)$ ,  $\Lambda_{2,2}(u_k, d_k)$  e  $\Lambda_{3,2}(u_k, d_k)$ . Similarmen-  
te como para (6.34), (6.35) e (6.36), as razões de log-verossimilhança podem ser escritas  
como:

$$\Lambda_{1,2}(u_k, d_k) = \log \frac{p_k^2(1, 0)}{p_k^2(0, 0)} + \frac{2\tilde{r}_k^{(0)} + 2}{\sigma^2} + \Lambda_{1,2e}(u_k, d_k), \quad (6.47)$$

$$\Lambda_{2,2}(u_k, d_k) = \log \frac{p_k^2(1, 1)}{p_k^2(0, 0)} + \frac{4}{\sigma^2} \tilde{r}_k^{(0)} + \Lambda_{2,2e}(u_k, d_k), \quad (6.48)$$

$$\Lambda_{3,2}(u_k, d_k) = \log \frac{p_k^2(0, 1)}{p_k^2(0, 0)} + \frac{2\tilde{r}_k^{(0)} + 2}{\sigma^2} + \Lambda_{3,2e}(u_k, d_k). \quad (6.49)$$

Substituindo (6.43), (6.44), (6.45) e (6.46) em (6.47), (6.48) e (6.49) tem-se:

$$\Lambda_{1,2}(u_k, d_k) = \tilde{\Lambda}_{1,2e}(u_k, d_k) + \frac{2\tilde{r}_k^{(0)} + 2}{\sigma^2} + \Lambda_{1,2e}(u_k, d_k), \quad (6.50)$$

$$\Lambda_{2,2}(u_k, d_k) = \tilde{\Lambda}_{2,2e}(u_k, d_k) + \frac{4}{\sigma^2} \tilde{r}_k^{(0)} + \Lambda_{2,2e}(u_k, d_k), \quad (6.51)$$

$$\Lambda_{3,2}(u_k, d_k) = \tilde{\Lambda}_{3,2e}(u_k, d_k) + \frac{2\tilde{r}_k^{(0)} + 2}{\sigma^2} + \Lambda_{3,2e}(u_k, d_k), \quad (6.52)$$

em que  $\Lambda_{1,2e}(u_k, d_k)$ ,  $\Lambda_{2,2e}(u_k, d_k)$  e  $\Lambda_{3,2e}(u_k, d_k)$  são as informações extrínsecas para DEC2,  
que dependem das informações redundantes supridas pelos codificadores  $\mathcal{C}_1^|$  e  $\mathcal{C}_2^|$ , como em  
(6.40), (6.41) e (6.42). As informações extrínsecas de DEC2 podem ser usadas como estimativas  
das probabilidades *a priori* para DEC1 como em (6.37), (6.38) e (6.39). A razão de log-  
verossimilhança para o primeiro decodificador pode ser escrita como:

$$\Lambda_{1,1}(u_k, d_k) = \hat{\Lambda}_{1,2e}(u_k, d_k) + \frac{2r_k^{(0)} + 2}{\sigma^2} + \Lambda_{1,1e}(u_k, d_k), \quad (6.53)$$

$$\Lambda_{2,1}(u_k, d_k) = \hat{\Lambda}_{2,2e}(u_k, d_k) + \frac{4}{\sigma^2} r_k^{(0)} + \Lambda_{2,1e}(u_k, d_k), \quad (6.54)$$

$$\Lambda_{3,1}(u_k, d_k) = \hat{\Lambda}_{3,2e}(u_k, d_k) + \frac{2r_k^{(0)} + 2}{\sigma^2} + \Lambda_{3,1e}(u_k, d_k), \quad (6.55)$$

em que  $\hat{\Lambda}_{1,2e}(u_k, d_k)$ ,  $\hat{\Lambda}_{2,2e}(u_k, d_k)$  e  $\hat{\Lambda}_{3,2e}(u_k, d_k)$  correspondem respectivamente aos  $\Lambda_{1,2e}(u_k, d_k)$ ,  
 $\Lambda_{2,2e}(u_k, d_k)$  e  $\Lambda_{3,2e}(u_k, d_k)$  desentrelaçados.

**Exemplo 6.2.1** Considere que se está usando o esquema de concatenação ilustrado na Figura  
5.4, com  $C_1 = \{01, 10\}$  e  $C_2 = \{00, 01, 11\}$  e que os codificadores para  $\mathcal{C}_1$  e  $\mathcal{C}_2$  estão ilustrados  
nas Figuras 6.1 e 6.2, respectivamente, nas quais os codificadores para  $\mathcal{C}_1^- = \mathcal{C}_1^| = \mathcal{C}_2^- = \mathcal{C}_2^|$   
têm matrizes geradoras polinomiais  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix}$ . Na Figura 6.5 estão as curvas  
relacionando a probabilidade de erro por bit versus relação sinal ruído para dois usuários, para  
comparação do caso em que é usado apenas o algoritmo BCJR, e para o caso em que é utilizada  
a decodificação iterativa em que o entrelaçador utilizado é o de Berrou-Glavieux (Seção 4.1.2)  
com comprimento 512.

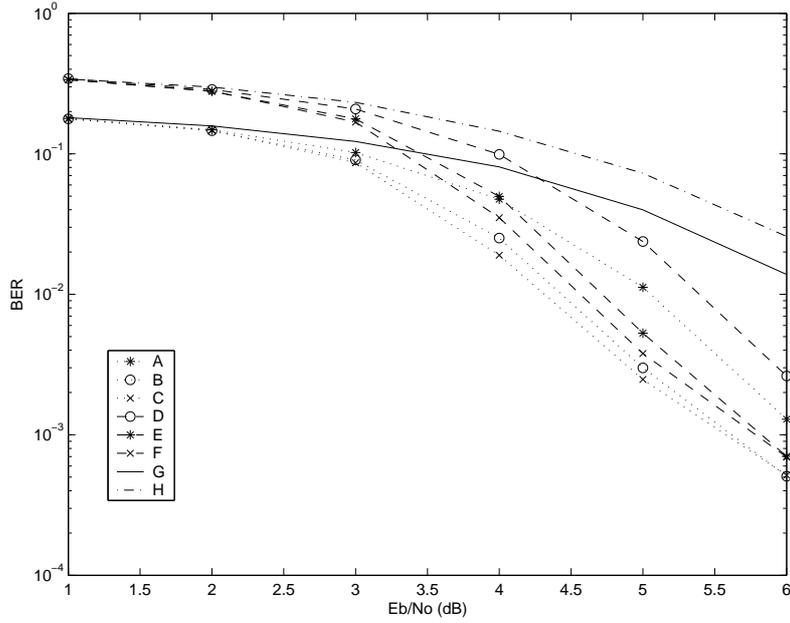


Figura 6.5: A- Usuário 1(1 iteração), B- Usuário 1(2 iterações), C- Usuário 1(3 iterações), D- Usuário 2 (1 iteração), E- Usuário 2 (2 iterações), F- Usuário 2 (3 iterações), G - Usuário 1 (Algoritmo BCJR), H - Usuário 2 (Algoritmo BCJR).

**Exemplo 6.2.2** Considere o exemplo 6.2.1, tendo os codificadores para  $\mathcal{C}_1^- = \mathcal{C}_1^+ = \mathcal{C}_2^- = \mathcal{C}_2^+$  matrizes geradoras polinomiais  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^3}{1+D^2+D^3} \end{bmatrix}$ . Na Figura 6.6 estão ilustradas as curvas relacionando a probabilidade de erro por bit versus relação sinal ruído para dois usuários, para comparação do caso em que é usado apenas o algoritmo BCJR, e para o caso em que é utilizada a decodificação iterativa em que o entrelaçador utilizado é o de Berrou-Glavieux (Seção 4.1.2) com comprimento 512. As Figuras 6.7 e 6.8 comparam para o usuário 1 e 2 respectivamente, os casos em que  $\mathcal{C}_1^- = \mathcal{C}_1^+ = \mathcal{C}_2^- = \mathcal{C}_2^+$  têm matrizes geradoras polinomiais  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix}$  (Exemplo 6.2.1) e o caso em que  $\mathcal{C}_1^- = \mathcal{C}_1^+ = \mathcal{C}_2^- = \mathcal{C}_2^+$  têm matrizes geradoras polinomiais  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^3}{1+D^2+D^3} \end{bmatrix}$ .

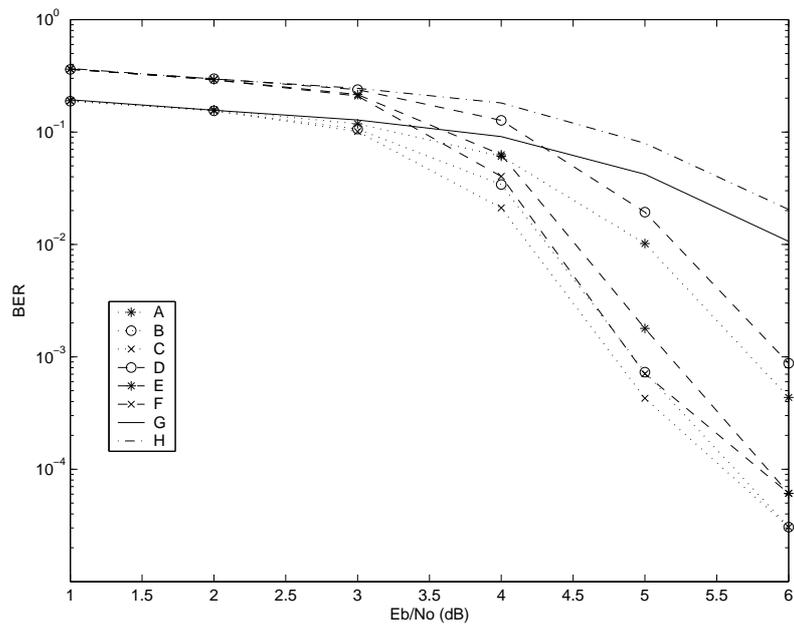


Figura 6.6: A- Usuário 1 (1 iteração), B- Usuário 1 (2 iterações), C- Usuário 1 (3 iterações), D- Usuário 2 (1 iteração), E- Usuário 2 (2 iterações), F- Usuário 2 (3 iterações), G - Usuário 1 (Algoritmo BCJR), H - Usuário 2 (Algoritmo BCJR).

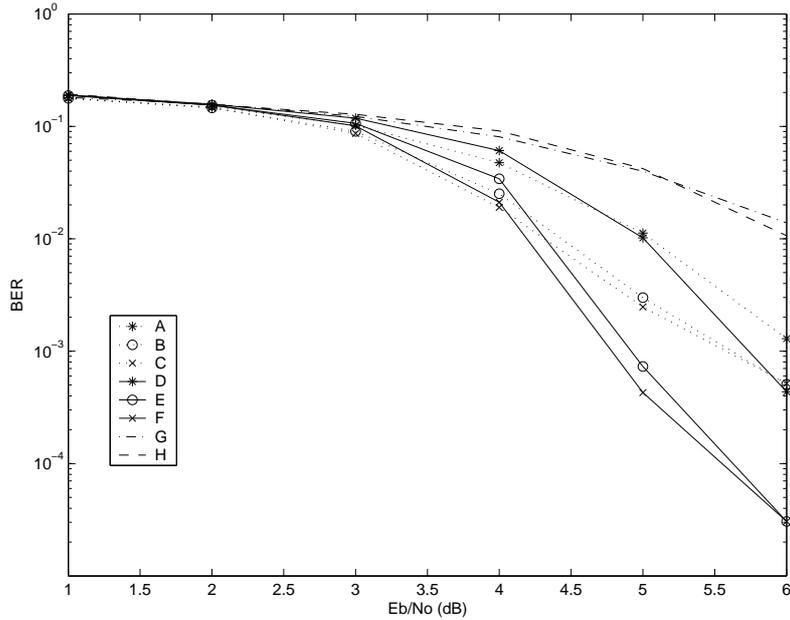


Figura 6.7: Curvas relacionadas ao usuário 1. Os casos para os quais  $\mathcal{C}_1^- = \mathcal{C}_1^| = \mathcal{C}_2^- = \mathcal{C}_2^|$  têm matrizes geradoras polinomiais  $G(D) = \left[1 \quad \frac{1+D+D^2}{1+D^2}\right]$  estão ilustrados em: A- 1 iteração, B- 2 iterações e C- 3 iterações; Os casos para os quais  $\mathcal{C}_1^- = \mathcal{C}_1^| = \mathcal{C}_2^- = \mathcal{C}_2^|$  têm matrizes geradoras polinomiais  $G(D) = \left[1 \quad \frac{1+D+D^3}{1+D^2+D^3}\right]$  estão ilustrados em: D- 1 iteração, E- 2 iterações e F- 3 iterações; O uso do algoritmo BCJR, sem a decodificação iterativa está ilustrado em: G - matriz geradora do codificador convolucional é  $G(D) = \left[1 \quad \frac{1+D+D^2}{1+D^2}\right]$ , H- matriz geradora do codificador convolucional é  $G(D) = \left[1 \quad \frac{1+D+D^3}{1+D^2+D^3}\right]$ .

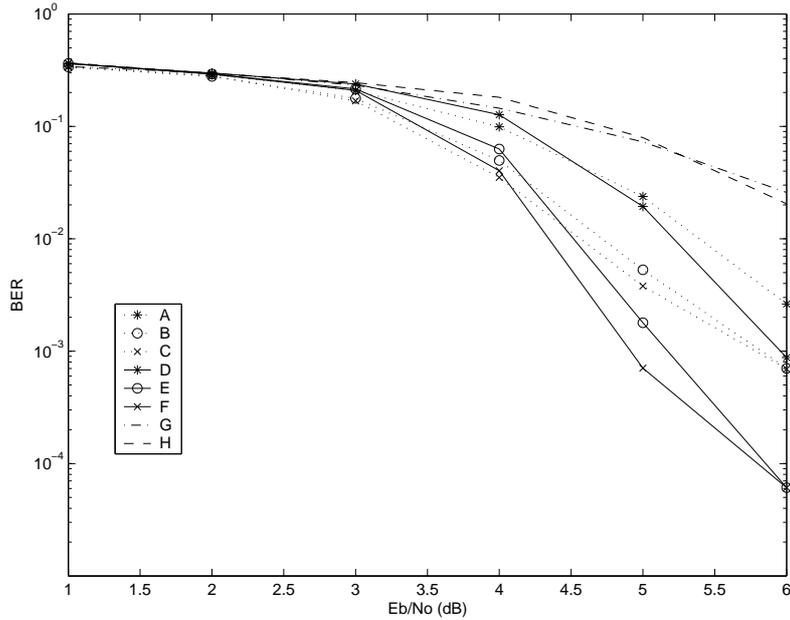


Figura 6.8: Curvas relacionadas ao usuário 2. Os casos para os quais  $\mathcal{C}_1^- = \mathcal{C}_1^| = \mathcal{C}_2^- = \mathcal{C}_2^|$  têm matrizes geradoras polinomiais  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix}$  estão ilustrados em: A- 1 iteração, B- 2 iterações e C- 3 iterações; Os casos para os quais  $\mathcal{C}_1^- = \mathcal{C}_1^| = \mathcal{C}_2^- = \mathcal{C}_2^|$  têm matrizes geradoras polinomiais  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^3}{1+D^2+D^3} \end{bmatrix}$  estão ilustrados em: D- 1 iteração, E- 2 iterações e F- 3 iterações; O uso do algoritmo BCJR, sem a decodificação iterativa está ilustrado em: G - matriz geradora do codificador convolucional é  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix}$ , H- matriz geradora do codificador convolucional é  $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^3}{1+D^2+D^3} \end{bmatrix}$ .

# Capítulo 7

## Decodificação Iterativa para o Canal Aditivo com Dois Usuários Binários Usando *CCMA*

O esquema de codificação (CCMA) (*Collaborative Coding Multiple Access*) [29] [30] [31] [32] permite que múltiplos usuários transmitam independentemente e simultaneamente através de um mesmo canal de comunicações, sem necessitar de subdivisão no tempo ou frequência e sem utilizar códigos ortogonais. Em um CCMA básico cada usuário transmite suas palavras código que são combinadas no canal e a decodificação é feita por um único decodificador no receptor. Este capítulo introduz o uso de decodificação iterativa para o 2-BAC, usando uma concatenação de códigos de bloco.

### 7.1 O Codificador

Nesta seção duas construções de códigos são apresentadas. A primeira, denominada construção 1, usa uma concatenação de dois códigos de bloco binários e a transmissão é feita através do 2-BAC. A segunda construção, denominada construção 2, usa também uma concatenação de dois códigos de bloco, sendo um deles binário e o outro ternário. Neste segundo caso, os dois usuários devem estar perto fisicamente e a transmissão se dará através de um canal ternário, ao invés do 2-BAC.

### 7.1.1 Construção 1

Seja  $\mathbf{C}_1$  um código de bloco com parâmetros  $(n, k_1)$  e  $\mathbf{C}_2$  um código de bloco com parâmetros  $(n, k_2)$ . O par  $(\mathbf{C}_1, \mathbf{C}_2)$  é unicamente decodificável no 2-BAC. Seja  $(\mathcal{C}_1, \mathcal{C}_2)$  um par de códigos produto sistemáticos com parâmetros  $(N_1N_2 - (N_1 - K_1)(N_2 - K_2), K_1K_2)$  [2] ilustrados nas figuras 7.1 e 7.2, em que  $K_1.K_2 = a.n$ ,  $a = 1, 2, \dots$

O código para o usuário 1 é construído a partir de  $\mathcal{C}_1$  e de  $\mathbf{C}_1$  do seguinte modo. O usuário 1 envia suas mensagens para o codificador de  $\mathbf{C}_1$ , e as palavras código resultantes de  $\mathbf{C}_1$  são enviadas como mensagens para o codificador de  $\mathcal{C}_1$ . Desta forma o usuário 1 estará se servindo de um dicionário contendo um subconjunto das palavras código de  $\mathcal{C}_1$ , escolhidas de acordo com as “mensagens” alimentadas por  $\mathbf{C}_1$  ao codificador de  $\mathcal{C}_1$ . A codificação para o usuário 2 é semelhante, empregando o código  $\mathcal{C}_2$  e o código  $\mathbf{C}_2$ . Conseqüentemente, o usuário 2 estará se servindo de um dicionário contendo um subconjunto das palavras código de  $\mathcal{C}_2$ , escolhidas de acordo com as “mensagens” alimentadas por  $\mathbf{C}_2$  ao codificador de  $\mathcal{C}_2$ . As mensagens dos dois codificadores de  $\mathcal{C}_1$  e  $\mathcal{C}_2$  devem ser alimentadas da mesma forma, linha a linha ou coluna a coluna. Essencialmente, a operação de codificação executada por cada usuário é uma concatenação serial do seu respectivo código de bloco com o código produto como ilustrado na Figura 7.4.

Pelas mesmas razões apresentadas na seção 5.3 a soma aritmética das palavras código produzidas pelo usuário 1 e pelo usuário 2, é unicamente decodificável se  $K_1.K_2 = a.n$   $a = 1, 2, \dots$  (A seção de informação dos códigos produto  $\mathcal{C}_1$  e  $\mathcal{C}_2$  deve conter um número inteiro de palavras código de  $\mathbf{C}_1$  e de  $\mathbf{C}_2$ , respectivamente).

Seja  $\mathcal{C}_T$  (Figura 7.3) o código resultante da soma aritmética de todas as palavras código de  $\mathcal{C}_1$  e todas as palavras código de  $\mathcal{C}_2$ . A seção de informação de cada palavra código de  $\mathcal{C}_T$  é a soma aritmética das palavras código de  $\mathbf{C}_1$  e  $\mathbf{C}_2$  e a seção de paridade é composta pela soma aritmética dos bits constantes nas seções de paridade das palavras código de  $\mathcal{C}_1$  e  $\mathcal{C}_2$ .

Se a taxas de  $\mathbf{C}_1$  e  $\mathbf{C}_2$  são respectivamente iguais a  $R_1$  e  $R_2$  segue desta construção que

$$R_C = (R_1 + R_2) \frac{K_1K_2}{N_1N_2 - (N_1 - K_1)(N_2 - K_2)},$$

é a taxa do código consistindo da soma aritmética das palavras código de  $\mathcal{C}_1$  e  $\mathcal{C}_2$ . Portanto, se a taxa  $\frac{K_1K_2}{N_1N_2 - (N_1 - K_1)(N_2 - K_2)}$  se aproximar de 1,  $R_C$  estará muito próximo de  $R_1 + R_2$ . Assim, se  $R_1 + R_2$  para o par  $(\mathbf{C}_1, \mathbf{C}_2)$  alcança a capacidade no 2-BAC então  $R_C$  também alcançará.

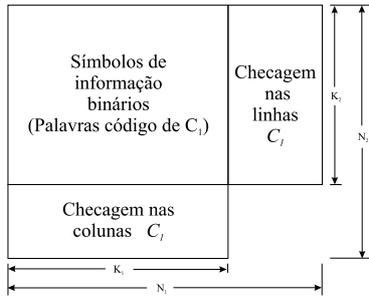


Figura 7.1: Arranjo representando palavra código de  $\mathcal{C}_1$ .

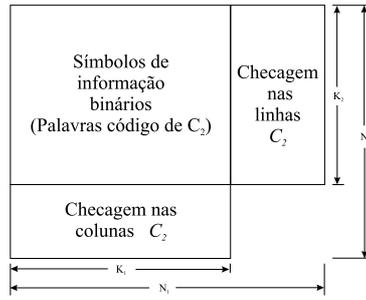


Figura 7.2: Arranjo representando palavra código de  $\mathcal{C}_2$ .

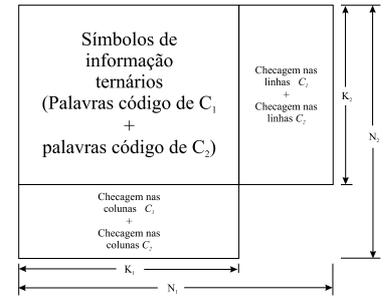


Figura 7.3: Arranjo representando palavra código de  $\mathcal{C}_T$ .

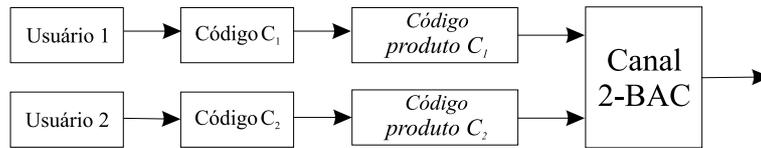


Figura 7.4: Esquema de concatenação serial empregando um par de códigos de bloco unicamente decodificáveis e um par de códigos produto.

### 7.1.2 Construção 2

Seja  $\mathbf{C}_1$  um código de bloco com parâmetros  $(n, k_1)$  e  $\mathbf{C}_2$  um código de bloco com parâmetros  $(n, k_2)$ . O par  $(\mathbf{C}_1, \mathbf{C}_2)$  é unicamente decodificável no 2-BAC. Seja  $\mathcal{C}_T^*$  um código de bloco sistemático ternário com parâmetros  $(N_1 N_2 - (N_1 - K_1)(N_2 - K_2), K_1 K_2)$ , em que  $K_1 \cdot K_2 = a \cdot n$ ,  $a = 1, 2, \dots$ . O código de bloco  $\mathcal{C}_T^*$  é formado de tal forma que cada palavra código é um arranjo com  $N_1$  colunas e  $N_2$  linhas no qual cada linha é uma palavra código em  $\mathcal{C}^-$ , em que  $\mathcal{C}^-$  é um código de bloco sistemático com parâmetros  $(N_1, K_1)$ , e cada coluna é uma palavra código em  $\mathcal{C}^|$ , em que  $\mathcal{C}^|$  é um código de bloco sistemático com parâmetros  $(N_2, K_2)$ . As palavras código de  $\mathcal{C}_T^*$  têm a estrutura ilustrada na Figura 7.5. Os  $K_1 K_2$  símbolos ternários localizados na quina superior esquerda são símbolos de informação. Os  $(N_1 - K_1) \cdot K_2$  símbolos ternários localizados na quina superior direita são calculados a partir das regras de paridade de  $\mathcal{C}^-$  e os  $(N_2 - K_2) \cdot K_1$  símbolos ternários localizados na quina inferior esquerda são calculados a partir das regras de paridade de  $\mathcal{C}^|$ .

O usuário 1 alimenta suas mensagens no codificador de  $\mathbf{C}_1$ . Similarmente, o usuário 2 alimenta suas mensagens no codificador de  $\mathbf{C}_2$ . A soma aritmética das palavras código resultantes de  $\mathbf{C}_1$  e de  $\mathbf{C}_2$  é alimentada como mensagem para o codificador de  $\mathcal{C}_T^*$ . Este esquema de codificação está ilustrado na Figura 7.6. Como  $\mathcal{C}_T^*$  é sistemático então esse esquema de codificação

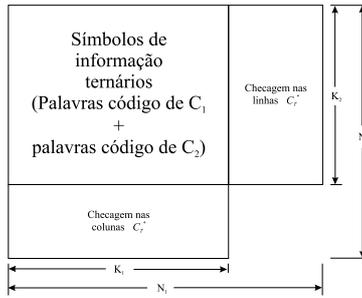


Figura 7.5: Arranjo representando palavra código do código ternário  $\mathcal{C}_T^*$ .

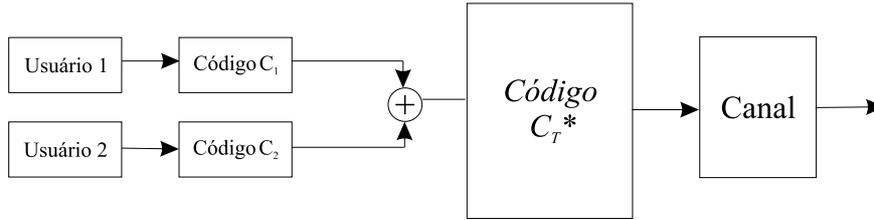


Figura 7.6: Esquema de concatenação serial empregando um par de códigos de bloco unicamente decodificáveis para o 2-BAC e um código de bloco ternário.

também é unicamente decodificável para 2-BAC (ver Seção 5.3).

Se a taxa de  $\mathbf{C}_1$  é  $R_1$  e a taxa de  $\mathbf{C}_2$  é  $R_2$  segue da construção que  $R_C = (R_1 + R_2) \frac{K_1 K_2}{N_1 N_2 - (N_1 - K_1)(N_2 - K_2)}$  é a taxa do código consistindo da soma aritmética de palavras código de  $\mathcal{C}_1$  e  $\mathcal{C}_2$ . Portanto, se a taxa  $\frac{K_1 K_2}{N_1 N_2 - (N_1 - K_1)(N_2 - K_2)}$  se aproximar de 1,  $R_C$  estará muito próximo de  $R_1 + R_2$ . Assim, se  $R_1 + R_2$  para o par  $(\mathbf{C}_1, \mathbf{C}_2)$  alcança a capacidade no 2-BAC então  $R_C$  também alcançará.

## 7.2 Decodificação Iterativa

O decodificador a ser utilizado para os esquemas de codificação (construção 1 e construção 2) introduzidos neste capítulo, emprega o decodificador turbo seguido pelo decodificador 2-BAC ilustrado na Figura 6.3. O decodificador turbo pode usar a regra MAP [40] [51] [53] para estimar a sequência ternária mais provável. Em seguida o decodificador 2-BAC faz uma estimativa das mensagens dos usuários 1 e 2, baseando-se para isto na decodibilidade única do par  $(\mathbf{C}_1, \mathbf{C}_2)$ .

### 7.2.1 Álgebra de Log-verossimilhança

Seja  $V$  em  $GF(2)$  com os elementos  $\{+1, -1\}$ , e similarmente seja  $W$  em  $GF(2)$  com os elementos  $\{+1, -1\}$  em que  $+1$  é o elemento nulo sob a adição  $\oplus$ . Seja  $X$  em  $GF(3)$  com os elementos  $\{-2, 0, +2\}$ , em que cada elemento  $x$  em  $X$  é a soma aritmética dos elementos  $v$  em

$V$  e do elemento  $w$  em  $W$ . Então  $x = v + w$ .

Sejam  $P\{v = i\}$ ,  $P\{w = s\}$ ,  $P\{x = t\}$  as probabilidades que as variáveis aleatórias  $V$ ,  $W$  e  $X$  tenham os valores  $v = i$ ,  $w = s$  e  $x = t$ , respectivamente. Segue que:

$$P\{x = 0\} = P\{v = -1\}P\{w = 1\} + P\{v = 1\}P\{w = -1\}, \quad (7.1)$$

$$P\{x = -2\} = P\{v = -1\}P\{w = -1\}, \quad (7.2)$$

$$P\{x = +2\} = P\{v = 1\}P\{w = 1\}. \quad (7.3)$$

As razões de log-verossimilhança das variáveis aleatórias ternárias  $X$  são definidas como  $\Lambda_1(x)$  e  $\Lambda_2(x)$  em que,

$$\Lambda_1(x) = \log \frac{P\{x = 2\}}{P\{x = -2\}} = \log \frac{P\{v = 1\}P\{w = 1\}}{P\{v = -1\}P\{w = -1\}}, \quad (7.4)$$

$$\Lambda_2(x) = \log \frac{P\{x = 0\}}{P\{x = -2\}} = \log \frac{P\{v = -1\}P\{w = 1\} + P\{v = 1\}P\{w = -1\}}{P\{v = -1\}P\{w = -1\}} \quad (7.5)$$

Utilizando a igualdade

$$1 = P\{x = -2\} + P\{x = 0\} + P\{x = 2\}, \quad (7.6)$$

prova-se que:

$$P\{x = -2\} = \frac{1}{1 + \exp(\Lambda_1(x)) + \exp(\Lambda_2(x))}, \quad (7.7)$$

$$P\{x = 2\} = \frac{\exp(\Lambda_1(x))}{1 + \exp(\Lambda_1(x)) + \exp(\Lambda_2(x))}, \quad (7.8)$$

$$P\{x = 0\} = \frac{\exp(\Lambda_2(x))}{1 + \exp(\Lambda_1(x)) + \exp(\Lambda_2(x))}. \quad (7.9)$$

A codificação dos valores binários  $v$  e  $w$ , implica na codificação do valor ternário  $x = v + w$  cujos valores suaves são  $\Lambda_1(x)$  e  $\Lambda_2(x)$ . Após a transmissão através do canal 2-BAC contaminado com ruído branco gaussiano tem-se a razão de log-verossimilhança de  $x$  condicionada à saída  $r$ .

$$\begin{aligned} \Lambda_1(x|r) &= \log \frac{P\{x = +2|r\}}{P\{x = -2|r\}} \\ &= \log \left( \frac{P\{r|x = +2\} P\{x = +2\}}{P\{r|x = -2\} P\{x = -2\}} \right) \\ &= \log \left( \frac{\exp\left(\frac{-(r-2)^2}{2\sigma^2}\right)}{\exp\left(\frac{-(r+2)^2}{2\sigma^2}\right)} \right) + \log \frac{P\{x = +2\}}{P\{x = -2\}} \\ &= \frac{8r}{2\sigma^2} + \Lambda_1(x), \end{aligned} \quad (7.10)$$

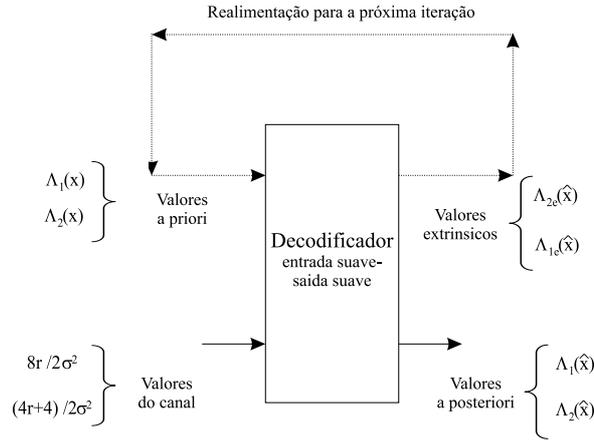


Figura 7.7: O decodificador com entrada suave/saída suave usa os valores a priori  $\Lambda_1(x_k)$  e  $\Lambda_2(x_k)$  para todos os símbolos de informação, se disponível, e os valores do canal  $\frac{4r+4}{2\sigma^2}$  e  $\frac{8r}{2\sigma^2}$  para todos os símbolos codificados. Ele também entrega as saídas suaves  $\Lambda_1(\hat{x}_k)$  e  $\Lambda_2(\hat{x}_k)$  para todos os símbolos de informação e as informações extrínsecas  $\Lambda_{1e}(\hat{x}_k)$  e  $\Lambda_{2e}(\hat{x}_k)$ .

$$\begin{aligned}
 \Lambda_2(x|r) &= \log \frac{P\{x = +0|r\}}{P\{x = -2|r\}} \\
 &= \log \left( \frac{P\{r|x = 0\} P\{x = 0\}}{P\{r|x = -2\} P\{x = -2\}} \right) \\
 &= \log \left( \frac{\exp\left(\frac{-r^2}{2\sigma^2}\right)}{\exp\left(\frac{-(r+2)^2}{2\sigma^2}\right)} \right) + \log \frac{P\{x = 0\}}{P\{x = -2\}} \\
 &= \frac{4r + 4}{2\sigma^2} + \Lambda_2(x), \tag{7.11}
 \end{aligned}$$

em que  $\sigma^2$  é a variância do ruído.

### 7.2.2 Decodificador Turbo

No que segue, considere  $\mathcal{C}_T$  e  $\mathcal{C}_T^*$ . Seja  $\mathcal{C}^-$  o código de bloco horizontal e seja  $\mathcal{C}^l$  o código de bloco vertical. Assuma que há um decodificador *soft-in/soft-out* (entrada suave/saída suave) disponível como mostrado na Figura 7.7 para decodificar os códigos componentes. Considere  $\Lambda_1(\hat{x}_k)$  a razão de log-verossimilhança para um “+2” transmitido e um “-2” transmitido na sequência de informação e  $\Lambda_2(\hat{x}_k)$  a razão de log-verossimilhança para um “0” transmitido e um “-2” transmitido na sequência de informação.

$$\begin{aligned}
 \Lambda_1(\hat{x}_k) &= \Lambda_1(x_k|\mathbf{r}) = \log \frac{P\{x_k = +2|\mathbf{r}\}}{P\{x_k = -2|\mathbf{r}\}}, \\
 \Lambda_2(\hat{x}_k) &= \Lambda_2(x_k|\mathbf{r}) = \log \frac{P\{x_k = 0|\mathbf{r}\}}{P\{x_k = -2|\mathbf{r}\}}.
 \end{aligned}$$

O decodificador usa os valores a priori  $\Lambda_1(x_k)$  e  $\Lambda_2(x_k)$  para todos os símbolos de informação, se disponível, e os valores do canal  $\frac{4r+4}{2\sigma^2}$  e  $\frac{8r}{2\sigma^2}$  para todos os símbolos codificados. Ele também entrega as saídas suaves  $\Lambda_1(\hat{x}_k)$  e  $\Lambda_2(\hat{x}_k)$  para todos os símbolos de informação e as informações extrínsecas  $\Lambda_{1e}(\hat{x}_k)$  e  $\Lambda_{2e}(\hat{x}_k)$ , que não são influenciadas por  $\Lambda_1(x_k)$ ,  $\Lambda_2(x_k)$ ,  $\frac{4r+4}{2\sigma^2}$  e  $\frac{8r}{2\sigma^2}$  relacionados com o símbolo atual. Para códigos sistemáticos, as saídas suaves para o símbolo de informação  $x_k$  são dadas por:

$$\Lambda_1(\hat{x}_k) = \frac{8r}{2\sigma^2} + \Lambda_1(x_k) + \Lambda_{1e}(\hat{x}_k), \quad (7.12)$$

$$\Lambda_2(\hat{x}_k) = \frac{4r+4}{2\sigma^2} + \Lambda_2(x_k) + \Lambda_{2e}(\hat{x}_k). \quad (7.13)$$

Tem-se então três estimativas independentes para cada razão de log-verossimilhança dos bits de informação: os valores do canal  $\frac{4r+4}{2\sigma^2}$  e  $\frac{8r}{2\sigma^2}$ , os valores a priori  $\Lambda_1(x_k)$  e  $\Lambda_2(x_k)$  e os valores  $\Lambda_{1e}(\hat{x}_k)$  e  $\Lambda_{2e}(\hat{x}_k)$ . Assumindo que inicialmente não há nenhuma informação a respeito dos símbolos de informação tem-se que os valores *a priori* disponíveis para a primeira iteração, são dados por  $\Lambda_1(x_k) = 0$  e  $\Lambda_2(x_k) = 0$ .

A decodificação horizontal do código  $C^-$  começa usando os valores correspondentes  $\frac{8r}{2\sigma^2}$  e  $\frac{4r+4}{2\sigma^2}$ . As informações extrínsecas  $\Lambda_{1e}^-(\hat{x}_k)$  e  $\Lambda_{2e}^-(\hat{x}_k)$  do código horizontal  $C^-$  para os símbolos de informação  $x_k$  são:

$$\Lambda_{1e}^-(\hat{x}_k) = \Lambda_1^-(\hat{x}_k) - \frac{8r}{2\sigma^2}, \quad (7.14)$$

$$\Lambda_{2e}^-(\hat{x}_k) = \Lambda_2^-(\hat{x}_k) - \frac{4+4r}{2\sigma^2}. \quad (7.15)$$

Estas estimativas independentes em  $x_k$  são agora usadas como valores *a priori* para a decodificação vertical do código  $C^l$  para obtenção de:

$$\Lambda_{1e}^l(\hat{x}_k) = \Lambda_1^l(\hat{x}_k) - \left(\frac{8r}{2\sigma^2} + \Lambda_{1e}^-(\hat{x}_k)\right), \quad (7.16)$$

$$\Lambda_{2e}^l(\hat{x}_k) = \Lambda_2^l(\hat{x}_k) - \left(\frac{4r+4}{2\sigma^2} + \Lambda_{2e}^-(\hat{x}_k)\right). \quad (7.17)$$

Esta informação extrínseca vertical será usada como um novo valor *a priori* na decodificação subsequente do código  $C^-$  no próximo passo da iteração. Para a decisão final (ou saída suave) depois da última iteração vertical, combina-se as duas últimas parcelas extrínsecas de informação com os valores recebidos para obter:

$$\Lambda_1(\hat{x}_k) = \frac{8r}{2\sigma^2} + \Lambda_{1e}^-(x_k) + \Lambda_{1e}^l(\hat{x}_k), \quad (7.18)$$

$$\Lambda_2(\hat{x}_k) = \frac{4r+4}{2\sigma^2} + \Lambda_{2e}^-(x_k) + \Lambda_{2e}^l(\hat{x}_k). \quad (7.19)$$

A estrutura do código para os dois usuários e do código para dois usuários\* permite dois passos de codificação separados, horizontal e vertical. O resumo da decodificação iterativa procede como segue:

1. Inicialize a informação *a priori*:

$$\Lambda_1(x_k) = 0,$$

$$\Lambda_2(x_k) = 0.$$

2. Decodifique horizontalmente e obtenha a informação extrínseca horizontal usando (7.12) e (7.13) como mostrado abaixo:

$$\Lambda_{1e}^-(\hat{x}_k) = \Lambda_1(\hat{x}_k) - \left(\frac{8r}{2\sigma^2} + \Lambda_1(x_k)\right), \quad (7.20)$$

$$\Lambda_{2e}^-(\hat{x}_k) = \Lambda_2(\hat{x}_k) - \left(\frac{4r+4}{2\sigma^2} + \Lambda_2(x_k)\right). \quad (7.21)$$

3. Considere  $\Lambda_1(x_k) = \Lambda_{1e}^-(\hat{x}_k)$  e  $\Lambda_2(x_k) = \Lambda_{2e}^-(\hat{x}_k)$ .
4. Decodifique verticalmente, e usando (7.12) e (7.13) obtenha a informação extrínseca vertical como mostrado abaixo:

$$\Lambda_{1e}^l(\hat{x}_k) = \Lambda_1(\hat{x}_k) - \left(\frac{8r}{2\sigma^2} + \Lambda_1(x_k)\right), \quad (7.22)$$

$$\Lambda_{2e}^l(\hat{x}_k) = \Lambda_2(\hat{x}_k) - \left(\frac{4r+4}{2\sigma^2} + \Lambda_2(x_k)\right). \quad (7.23)$$

5. Considere  $\Lambda_1(x_k) = \Lambda_{1e}^l(\hat{x}_k)$  e  $\Lambda_2(x_k) = \Lambda_{2e}^l(\hat{x}_k)$ .
6. Se o número de iterações já é suficiente para tomar a decisão, vá para o passo 7, senão vá para o passo 2.
7. As saídas suaves são:

$$\Lambda_1(\hat{x}_k) = \frac{8r}{2\sigma^2} + \Lambda_{1e}^-(x_k) + \Lambda_{1e}^l(\hat{x}_k), \quad (7.24)$$

$$\Lambda_2(\hat{x}_k) = \frac{4r+4}{2\sigma^2} + \Lambda_{2e}^-(x_k) + \Lambda_{2e}^l(\hat{x}_k). \quad (7.25)$$

## 7.3 Exemplos Tutoriais

**Exemplo 7.3.1** Considere que se está usando a construção serial ilustrada na Figura 7.4. Seja  $(\mathbf{C}_1, \mathbf{C}_2)$  o par de códigos de bloco unicamente decodificáveis sobre o 2-BAC, em que

$v_{11}$	$v_{12}$	$p_1^-$
$v_{21}$	$v_{22}$	$p_2^-$
$p_1^+$	$p_2^+$	

Figura 7.8: Palavra código do código produto binário  $\mathcal{C}_1$ .

$w_{11}$	$w_{12}$	$q_1^-$
$w_{21}$	$w_{22}$	$q_2^-$
$q_1^+$	$q_2^+$	

Figura 7.9: Palavra código do código produto binário  $\mathcal{C}_2$ .

$x_{11}$	$x_{12}$	$s_1^-$
$x_{21}$	$x_{22}$	$s_2^-$
$s_1^+$	$s_2^+$	

Figura 7.10: Palavra código do código ternário  $\mathcal{C}_T$ .

$\mathbf{C}_1 = \{-1 + 1, +1 - 1\}$  e  $\mathbf{C}_2 = \{-1 - 1, -1 + 1, +1 + 1\}$ . Sejam  $\mathcal{C}_1$  e  $\mathcal{C}_2$  dois códigos produto idênticos cujos códigos componentes têm parâmetros  $(3, 2)$ . Os  $K_1 K_2$  bits de informação  $\mathbf{v}$  de  $\mathcal{C}_1$  são ordenados em uma matriz retangular como mostrado na Figura 7.8. Junto a eles estão os bits de paridade  $p_1^- = v_{11} \oplus v_{12}$ ,  $p_2^- = v_{21} \oplus v_{22}$ ,  $p_1^+ = v_{11} \oplus v_{21}$  e  $p_2^+ = v_{12} \oplus v_{22}$  dos dois códigos sistemáticos  $\mathcal{C}_1^-$  e  $\mathcal{C}_1^+$ . Similarmente, os  $K_1 K_2$  bits de informação  $\mathbf{w}$  de  $\mathcal{C}_2$  são ordenados em uma matriz retangular como mostrado na Figura 7.9. Junto a eles estão os bits de paridade  $q_1^- = w_{11} \oplus w_{12}$ ,  $q_2^- = w_{21} \oplus w_{22}$ ,  $q_1^+ = w_{11} \oplus w_{21}$  e  $q_2^+ = w_{12} \oplus w_{22}$  dos dois códigos sistemáticos  $\mathcal{C}_2^-$  e  $\mathcal{C}_2^+$ .

A soma aritmética dos bits de informação  $\mathbf{v}$  e  $\mathbf{w}$  são ordenadas em uma matriz retangular, chamada matriz para dois usuários, como mostrado na Figura 7.10, em que  $x_{11} = v_{11} + w_{11}$ ,  $x_{12} = v_{12} + w_{12}$ ,  $x_{21} = v_{21} + w_{21}$  e  $x_{22} = v_{22} + w_{22}$ . Junto a eles estão os bits de paridade  $s_1^- = p_1^- + q_1^- = (v_{11} \oplus v_{12}) + (w_{11} \oplus w_{12})$ ,  $s_2^- = p_2^- + q_2^- = (v_{21} \oplus v_{22}) + (w_{21} \oplus w_{22})$ ,  $s_1^+ = p_1^+ + q_1^+ = (v_{11} \oplus v_{21}) + (w_{11} \oplus w_{21})$ ,  $s_2^+ = p_2^+ + q_2^+ = (v_{12} \oplus v_{22}) + (w_{12} \oplus w_{22})$ .

Usa-se os símbolos  $\diamond$  como a notação para as operações definidas por  $x_{11} \diamond s_1^- = x_{12}$ ,  $x_{12} \diamond s_1^- = x_{11}$ ,  $x_{21} \diamond s_2^- = x_{22}$ ,  $x_{22} \diamond s_2^- = x_{21}$ ,  $x_{11} \diamond s_1^+ = x_{21}$ ,  $x_{21} \diamond s_1^+ = x_{11}$ ,  $x_{12} \diamond s_2^+ = x_{22}$ ,  $x_{22} \diamond s_2^+ = x_{12}$ . Para simplicidade de notação pode-se chamar  $x_i$  e  $x_j$  o primeiro e o segundo termos usados nas operações  $\diamond$  definidas acima, segue que:

$$\begin{aligned} P\{x_i \diamond x_j = 0\} &= P\{x_i = 0\}P\{x_j = +2\} + P\{x_i = +2\}P\{x_j = 0\} + \\ &P\{x_i = 0\}P\{x_j = -2\} + P\{x_i = -2\}P\{x_j = 0\} \end{aligned} \quad (7.26)$$

$$\begin{aligned} P\{x_i \diamond x_j = -2\} &= P\{x_i = 0\}P\{x_j = 0\} + P\{x_i = -2\}P\{x_j = +2\} + \\ &P\{x_i = +2\}P\{x_j = -2\} \end{aligned} \quad (7.27)$$

$$\begin{aligned} P\{x_i \diamond x_j = +2\} &= P\{x_i = 0\}P\{x_j = 0\} + P\{x_i = -2\}P\{x_j = -2\} + \\ &P\{x_i = +2\}P\{x_j = +2\} \end{aligned} \quad (7.28)$$

-1	1	-1
-1	1	-1
1	1	

Figura 7.11: Palavra código para o código binário  $\mathcal{C}_1$ .

-1	-1	1
-1	-1	1
1	1	

Figura 7.12: Palavra código para o código binário  $\mathcal{C}_2$ .

-2	0	0
-2	0	0
2	2	

Figura 7.13: Palavra código para o código ternário  $\mathcal{C}_T$ .

-7.68	2.64	0.19
-3.62	-3.04	-0.13
9.98	8.12	

Figura 7.14: Valores recebidos  $\frac{8r}{2\sigma^2}$ .

-1.91	3.25	2.03
0.12	0.41	1.86
6.92	5.99	

Figura 7.15: Valores recebidos  $\frac{4r+4}{2\sigma^2}$ .

Usando (7.26), (7.27), (7.28) e (7.7), (7.8), (7.9) pode-se provar que para as variáveis aleatórias estatisticamente independentes  $X_i$  e  $X_j$  tem-se que:

$$\begin{aligned} \Lambda_1(x_i \diamond x_j) &= \log \frac{P\{x_i \diamond x_j = +2\}}{P\{x_i \diamond x_j = -2\}} \\ &= \log \frac{1 + \exp(\Lambda_1(x_i) + \Lambda_1(x_j)) + \exp(\Lambda_2(x_i) + \Lambda_2(x_j))}{\exp(\Lambda_1(x_i)) + \exp(\Lambda_1(x_j)) + \exp(\Lambda_2(x_i) + \Lambda_2(x_j))}, \\ \Lambda_2(x_i \diamond x_j) &= \log \frac{P\{x_i \diamond x_j = 0\}}{P\{x_i \diamond x_j = -2\}} \\ &= \log \frac{\exp(\Lambda_2(x_i)) + \exp(\Lambda_2(x_j)) + \exp(\Lambda_2(x_i) + \Lambda_1(x_j)) + \exp(\Lambda_2(x_j) + \Lambda_1(x_i))}{\exp(\Lambda_1(x_i)) + \exp(\Lambda_1(x_j)) + \exp(\Lambda_2(x_i) + \Lambda_2(x_j))}. \end{aligned}$$

Codifica-se dois bits de informação utilizando o codificador para  $\mathbf{C}_1$ . As duas palavras-código resultantes são entradas para o codificador de  $\mathcal{C}_1$ , isto é, estas duas palavras-código representam quatro bits de informação para o codificador de  $\mathcal{C}_1$  cujos elementos pertencem a  $\{+1, -1\}$  como mostrado na Figura 7.11. Similarmente para o usuário 2 dois bits de informação são codificados usando o codificador para  $\mathbf{C}_2$ . As palavras-código resultantes são alimentadas como mensagens para o codificador de  $\mathcal{C}_2$  e representam quatro bits de informação para o codificador de  $\mathcal{C}_2$  cujos elementos pertencem a  $\{+1, -1\}$  como mostrado na Figura 7.12. A palavra-código para dois usuários respectiva está ilustrada na Figura 7.13. Pode-se assumir que as palavras código dos dois usuários são transmitidas por um 2-BAC contaminado com ruído branco gaussiano. Suponha que o vetor recebido é  $r$ . Os valores correspondentes a  $\frac{8r}{2\sigma^2}$  e  $\frac{4r+4}{2\sigma^2}$  são mostrados nas Figuras 7.14 e 7.15, respectivamente.

Nenhuma informação a priori está ainda disponível. Pode-se começar com a decodificação

0.01	-0.09
0.01	0.01

Figura 7.16: Informação extrínscica  $\Lambda_{1e}^-$  após a primeira decodificação horizontal.

-0.21	1.22
-0.10	0.06

Figura 7.17: Informação extrínscica  $\Lambda_{2e}^-$  após a primeira decodificação horizontal.

-2.63	-1.60
-5.09	0.71

Figura 7.18: Informação extrínscica  $\Lambda_{1e}^\perp$  após a primeira decodificação vertical.

0.02	0.37
-1.79	2.06

Figura 7.19: Informação extrínscica  $\Lambda_{2e}^\perp$  após a primeira decodificação vertical.

*horizontal: A informação para o bit  $x_{11}$  é recebido duas vezes: Diretamente via  $x_{11}$  e indiretamente via  $x_{12} \diamond s_1^-$ . Como  $x_{12}$  e  $s_1^-$  são transmitidos estatisticamente independentes pode-se calcular*

$$\Lambda_{1e}^-(x_{12} \diamond s_1^-),$$

$$\Lambda_{2e}^-(x_{12} \diamond s_1^-).$$

*Esta informação indireta sobre  $x_{11}$  são os valores extrínscicos horizontais e estão armazenados nas Figuras 7.16 e 7.17. Para  $x_{12}$  obtém-se pelos mesmos argumentos os valores extrínscicos horizontais para a segunda linha. Quando se conclui o preenchimento das tabelas com os valores extrínscicos horizontais, se começa a decodificação vertical usando os valores  $\Lambda_{1e}^-$  e  $\Lambda_{2e}^-$  como valores a priori para a decodificação vertical. Os valores extrínscicos verticais estão armazenados nas Figuras 7.18 e 7.19. Ao parar as iterações por aqui obtém-se como saídas suaves depois da iteração vertical os valores  $\Lambda_1(\hat{x})$  (7.24) e  $\Lambda_2(\hat{x})$  (7.25) mostrados nas Figuras 7.20 e 7.21. As probabilidades relacionadas com os símbolos  $-2$ ,  $2$  e  $0$  estão mostradas nas Figuras 7.22, 7.23 and 7.24. Assim os valores estimados são  $\hat{x}_{11} = -2$ ,  $\hat{x}_{12} = 0$ ,  $\hat{x}_{21} = -2$  e  $\hat{x}_{22} = 0$ . Alimentando o decodificador 2-BAC com estes valores ternários, tem-se que as seqüências de informação binárias estimadas para o usuário 1 e o usuário 2 são dadas respectivamente por  $\hat{v}_{11} = -1$ ,  $\hat{v}_{12} = 1$ ,  $\hat{v}_{21} = -1$  e  $\hat{v}_{22} = 1$ ,  $\hat{w}_{11} = -1$ ,  $\hat{w}_{12} = -1$ ,  $\hat{w}_{21} = -1$  e  $\hat{w}_{22} = -1$ . A Figura 7.25 ilustra um gráfico da probabilidade de erro versus a relação sinal ruído para comparação do desempenho para os dois usuários. As curvas ilustram o caso para uma interação e para duas iterações.*

**Exemplo 7.3.2** Considere que se está usando a construção serial ilustrada na Figura 7.6.

-10.3	0.94
-8.70	-2.32

Figura 7.20: Saída suave  $\Lambda_1(\hat{x})$  após a primeira decodificação horizontal e vertical.

-2.10	4.84
-1.78	2.53

Figura 7.21: Saída suave  $\Lambda_2(\hat{x})$  após a primeira decodificação horizontal e vertical.

0.89	0.01
0.85	0.07

Figura 7.22: Probabilidades relativas ao símbolo -2.

0.00	0.02
0.00	0.01

Figura 7.23: Probabilidades relativas ao símbolo 2.

0.11	0.97
0.14	0.92

Figura 7.24: Probabilidades relativas ao símbolo 0.

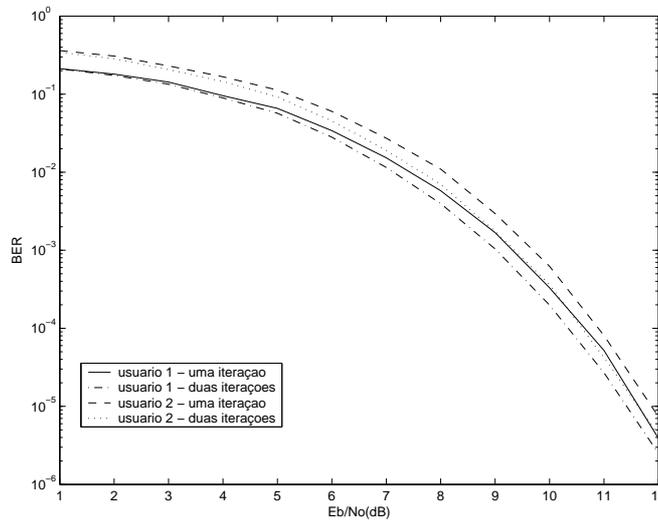


Figura 7.25: Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação do desempenho alcançado após uma e duas iterações para ambos usuários usando a construção 1.

$x_{11}$	$x_{12}$	$s_1^-$
$x_{21}$	$x_{22}$	$s_2^-$
$s_1^+$	$s_2^+$	

Figura 7.26: Palavra código para o código ternário  $\mathcal{C}_T^*$ .

Seja  $(\mathbf{C}_1, \mathbf{C}_2)$  um par de códigos de bloco unicamente decodificáveis sobre o 2-BAC, em que  $\mathbf{C}_1 = \{-1+1, +1-1\}$  e  $\mathbf{C}_2 = \{-1-1, -1+1, +1+1\}$ . O usuário 1 alimenta suas mensagens para o codificador de  $\mathbf{C}_1$ . Similarmente, o usuário 2 alimenta suas mensagens para o codificador de  $\mathbf{C}_2$ . A soma aritmética das palavras código resultantes de  $\mathbf{C}_1$  e de  $\mathbf{C}_2$  são alimentadas como mensagens para o codificador de  $\mathcal{C}_T^*$ . Seja  $\mathcal{C}^- = \mathcal{C}^{\dagger} = \{000, 0-2-2, 0+2+2, +2+2-2, -2-2+2, -2+20, +2-20, +20+2, -20-2\}$  os códigos de bloco componentes com parâmetros  $(3, 2)$  para  $\mathcal{C}_T^*$ . A soma aritmética dos bits de informação  $\mathbf{v}$  (de  $\mathbf{C}_1$ ) e  $\mathbf{w}$  (de  $\mathbf{C}_2$ ) são ordenados em uma matriz retangular, chamada matriz para dois usuários\*, como mostrado na Figura 7.26, onde  $x_{11} = v_{11} + w_{11}$ ,  $x_{12} = v_{12} + w_{12}$ ,  $x_{21} = v_{21} + w_{21}$  e  $x_{22} = v_{22} + w_{22}$ . Junto a eles estão os bits de paridade  $s_1^-, s_2^-, s_1^+, s_2^+$  encontrados com o uso dos códigos componentes  $\mathcal{C}^- = \mathcal{C}^{\dagger}$ .

Usa-se o símbolo  $\diamond$  como a notação para as operações definidas como  $x_{11} \diamond s_1^- = x_{12}$ ,  $x_{12} \diamond s_1^- = x_{11}$ ,  $x_{21} \diamond s_2^- = x_{22}$ ,  $x_{22} \diamond s_2^- = x_{21}$ ,  $x_{11} \diamond s_1^+ = x_{21}$ ,  $x_{21} \diamond s_1^+ = x_{11}$ ,  $x_{12} \diamond s_2^+ = x_{22}$ ,  $x_{22} \diamond s_2^+ = x_{12}$ . Para simplicidade de notação vamos chamam-se  $x_i$  e  $x_j$  o primeiro e o segundo termos respectivamente, nas operações  $\diamond$  definidas acima, segue que:

$$P\{x_i \diamond x_j = 0\} = P\{x_i = 0\}P\{x_j = 0\} + P\{x_i = -2\}P\{x_j = -2\} + P\{x_i = 2\}P\{x_j = 2\} \quad (7.29)$$

$$P\{x_i \diamond x_j = -2\} = P\{x_i = 0\}P\{x_j = -2\} + P\{x_i = -2\}P\{x_j = +2\} + P\{x_i = 2\}P\{x_j = 0\} \quad (7.30)$$

$$P\{x_i \diamond x_j = +2\} = P\{x_i = 0\}P\{x_j = 2\} + P\{x_i = 2\}P\{x_j = -2\} + P\{x_i = -2\}P\{x_j = 0\} \quad (7.31)$$

Usando (7.29), (7.30), (7.31) e (7.7), (7.8), (7.9) não é difícil provar que para variáveis

aleatórias estatisticamente independentes  $X_i$  e  $X_j$  tem-se que:

$$\begin{aligned}\Lambda_1(x_i \diamond x_j) &= \log \frac{P\{x_i \diamond x_j = +2\}}{P\{x_i \diamond x_j = -2\}} \\ &= \log \frac{\exp(\Lambda_2(x_i) + \Lambda_1(x_j)) + \exp(\Lambda_1(x_i) + \Lambda_2(x_j))}{\exp(\Lambda_2(x_i)) + \exp(\Lambda_1(x_j)) + \exp(\Lambda_1(x_i) + \Lambda_2(x_j))}, \\ \Lambda_2(x_i \diamond x_j) &= \log \frac{P\{x_i \diamond x_j = 0\}}{P\{x_i \diamond x_j = -2\}} \\ &= \log \frac{1 + \exp(\Lambda_2(x_i) + \Lambda_2(x_j)) + \exp(\Lambda_1(x_i) + \Lambda_1(x_j))}{\exp(\Lambda_2(x_i)) + \exp(\Lambda_1(x_j)) + \exp(\Lambda_1(x_i) + \Lambda_2(x_j))}.\end{aligned}$$

Seguindo o mesmo procedimento usado no Exemplo 7.3.1, encontram-se as curvas ilustradas na Figura 7.27, na qual se mostra um gráfico da probabilidade de erro versus relação sinal ruído, comparando o desempenho alcançado com uma e duas interações para os dois usuários.

A Figura 7.28 mostra o gráfico da probabilidade de erro por bit versus relação sinal ruído a fim de comparar o desempenho alcançado com ambas as construções para o usuário 1. Similarmente, a Figura 7.29 mostra o gráfico da probabilidade de erro por bit versus relação sinal ruído a fim de comparar o desempenho alcançado com ambas as construções para o usuário 2.

Os resultados obtidos com o uso da construção 2 são melhores do que os resultados obtidos com o uso da construção 1. Podendo ser explicado porque os códigos componentes  $\mathcal{C}^- = \mathcal{C}^1$  usados no Exemplo 2 tem uma distância mínima igual a 2, então o código resultante  $\mathcal{C}_{\mathcal{T}}^*$  tem uma distância mínima igual a 4. No Exemplo 1 o código resultante  $\mathcal{C}_{\mathcal{T}}$  tem distância mínima igual a 1.

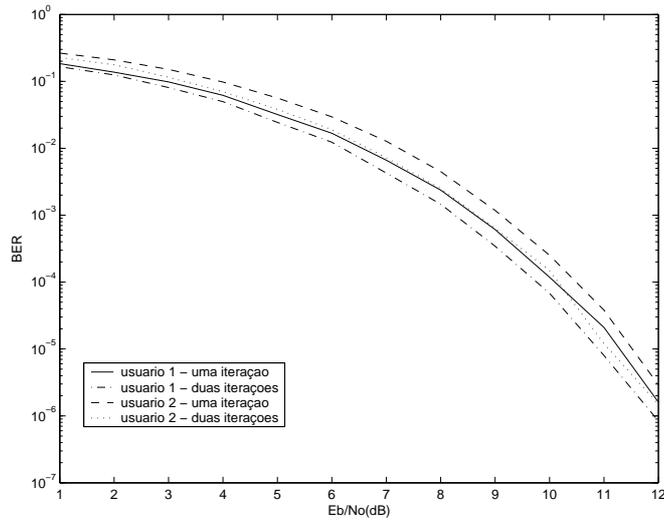


Figura 7.27: Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação do desempenho alcançado após uma e duas iterações para ambos usuários usando a construção 2.

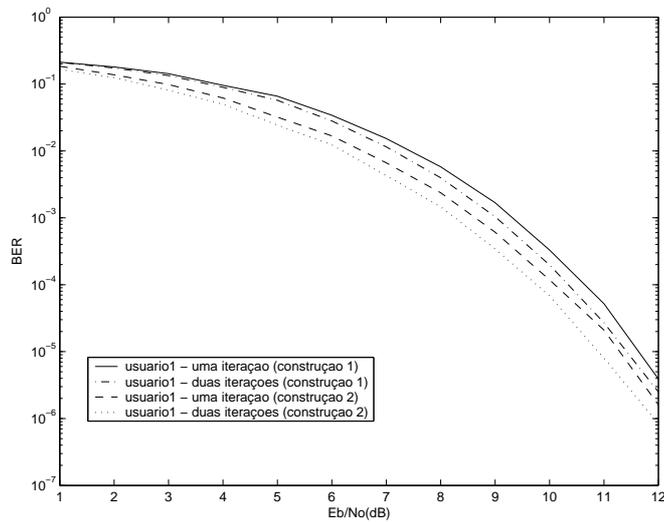


Figura 7.28: Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação de desempenho alcançado usando a construção 1 e a construção 2 para o usuário 1.

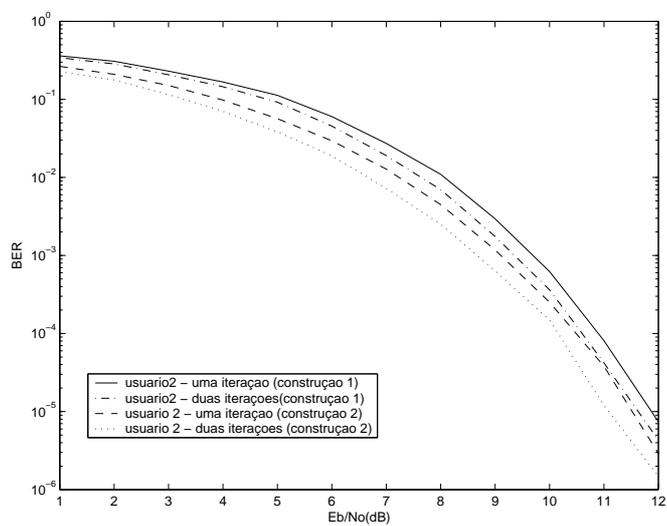


Figura 7.29: Curvas de probabilidade de erro por bit versus relação sinal ruído para comparação de desempenho alcançado usando a construção 1 e a construção 2 para o usuário 2.

# Capítulo 8

## Codificação Hierárquica para a Ethernet

No início dos anos 70, a *XEROX Corporation* introduziu *Ethernet* [60] para codificação de redes de dados. A especificação original utiliza o código *Manchester* [61, pp.145-147] e pode suportar um tráfego agregado de 10 Mbits/s. Um grande número destes sistemas foram instalados e permanecem em uso até hoje. Neste capítulo propõe-se um esquema de codificação que permite um acréscimo do tráfego total em redes *Ethernet*, mantendo compatibilidade com os usuários já existentes. Em outras palavras, novos usuários serão adicionados com o código proposto, enquanto os usuários existentes não precisarão mudar de *hardware*.

### 8.1 Código Manchester

O código *Manchester* (MC) é um código binário com dois elementos, em que o símbolo “0” é representado por  $p(t)$  e o símbolo “1” é representado por  $-p(t)$ , em que  $p(t)$  representa dois pulsos de polaridade opostas. Pode-se denotar este mapeamento de “0’s” e “1’s” como  $0 \rightarrow (-1, +1)$  e  $1 \rightarrow (+1, -1)$ , em que  $(-1, +1)$  denota  $p(t)$ .

### 8.2 Codificação Hierárquica

Pode-se introduzir novos usuários com o uso de códigos para o 2-BAC [16] como ilustrado na Figura 8.1.

O efeito positivo dessa nova técnica pode ser resumido abaixo:

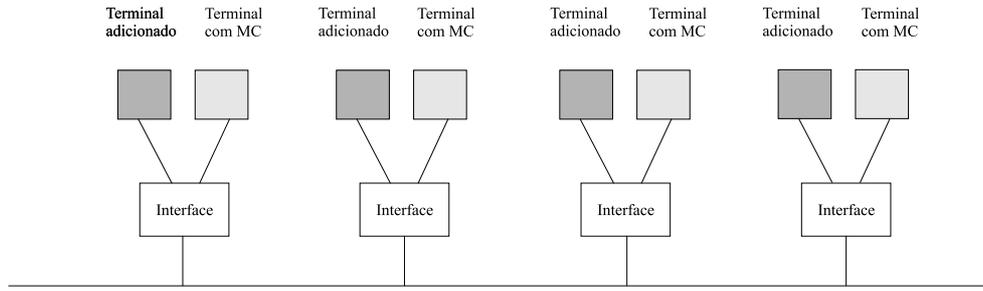


Figura 8.1: Inclusão de novos usuários em redes *Ethernet* já existentes. A especificação original utiliza o código Manchester.

Tabela 8.1: Par de códigos unicamente decodificáveis para o 2-BAC.

	-1, -1	-1, +1	+1, +1
-1, +1	-2, 0	-2, +2	0, +2
+1, -1	0, -2	0, 0	+2, 0

1. O código *Manchester* tem capacidade de  $R_1 = 1/2$  bits por uso do canal.
2. Um segundo usuário pode ser adicionado com capacidade de  $R_2 \leq 1$  bits por uso do canal.
3. O código para dois usuários resultante tem uma taxa total de  $R$ ,  $R = R_1 + R_2 \leq 1,5$  bits por uso do canal [16].

Sem perda de generalidade, pode-se especificar o par de códigos para o 2-BAC  $(C_1, C_2)$ , em que  $C_1 = \{01, 10\}$  e  $C_2 = \{00, 01, 11\}$  com taxa total de 1.29 bits por uso do canal. O par  $(C_1, C_2)$  é equivalente a um par de taxa idêntica introduzido em [16]. Outros pares de códigos unicamente decodificáveis sobre o 2-BAC a princípio podem ser usados com um mapeamento apropriado. Aplicando o mapeamento  $0 \rightarrow -1$ ;  $1 \rightarrow +1$  nas palavras código de  $C_1$  e  $C_2$ , respectivamente, obtém-se como resultado o código *Manchester* para o usuário 1 e um código para o usuário 2 que será chamado de código *pseudo Manchester*. Alimentando um 2-BAC sem ruído com +1's e -1's as saídas correspondentes estão mostradas na Tabela 8.1, na qual as linhas são indexadas pelas palavras códigos de  $C_1$  e as colunas são indexadas pelas palavras código de  $C_2$ .

Como os códigos  $C_1 = \{01, 10\}$  e  $C_2 = \{00, 01, 11\}$  têm diferentes taxas, pode-se utilizar um código de linha conhecido como 3B2T [61] para codificar blocos de três símbolos binários do usuário 2 em palavras código de  $C_2$ . A seqüência binária do usuário 2 é segmentada em blocos

Tabela 8.2: Mapeamento do código 3B2T no código *pseudo Manchester*.

BINÁRIO	TERNÁRIO	PSEUDO MANCHESTER
000	00	-1, -1, -1, -1
001	01	-1, -1, -1, +1
010	02	-1, -1, +1, +1
011	10	-1, +1, -1, -1
100	11	-1, +1, -1, +1
101	12	-1, +1, +1, +1
110	20	+1, +1, -1, -1
111	21	+1, +1, -1, +1
*	22	+1, +1, +1, +1

Tabela 8.3: Par de códigos unicamente decodificáveis para o 2-BAC com taxa 1,25.

	(-1,-1,-1,-1)	(-1,-1,-1,+1)	(-1,-1,+1,+1)	(-1,+1,-1,-1)	(-1,+1,-1,+1)	(-1,+1,+1,+1)	(+1,+1,-1,-1)	(+1,+1,-1,+1)
(-1,+1,-1,+1)	(-2, 0,-2, 0)	(-2, 0,-2,+2)	(-2, 0, 0,+2)	(-2,+2,-2, 0)	(-2,+2,-2,+2)	(-2,+2, 0,+2)	( 0,+2,-2, 0)	( 0,+2,-2,+2)
(-1,+1,+1,-1)	(-2, 0, 0,-2)	(-2, 0, 0, 0)	(-2, 0,+2, 0)	(-2,+2, 0,-2)	(-2,+2, 0, 0)	(-2,+2,+2, 0)	( 0,+2, 0,-2)	( 0,+2, 0, 0)
(+1,-1,-1,+1)	( 0,-2,-2, 0)	( 0,-2,-2,+2)	( 0,-2, 0,+2)	( 0, 0,-2, 0)	( 0, 0,-2,+2)	( 0, 0, 0,+2)	(+2, 0,-2, 0)	(+2, 0,-2,+2)
(+1,-1,+1,-1)	( 0,-2, 0,-2)	( 0,-2, 0, 0)	( 0,-2,+2, 0)	( 0, 0, 0,-2)	( 0, 0, 0, 0)	( 0, 0,+2, 0)	(+2, 0, 0,-2)	(+2, 0, 0, 0)

de comprimento 3, e cada bloco é mapeado em 2 símbolos ternários. Cada símbolo ternário nas 2-uplas do código de linha 3B2T é mapeado em uma palavra código de  $C_2$  do seguinte modo:  $0 \rightarrow (-1, -1)$ ,  $1 \rightarrow (-1, +1)$ , e  $2 \rightarrow (+1, +1)$ . O usuário 2 portanto tem taxa de informação de  $3/2$  bits por uso do canal. Em outras palavras, blocos de símbolos binários de comprimento 3 do usuário 2 são mapeados em 4-uplas como mostrado na Tabela 8.2, formando uma concatenação de palavras código do código *pseudo Manchester*. Palavras código síncronas de comprimento 4 serão produzidas por cada codificador, pela codificação simultânea de 2 símbolos de informação do usuário 1 e de 3 símbolos de informação do usuário 2, respectivamente. Além do mais, a adição de 4-uplas dos usuários 1 e 2 produzem o código unicamente decodificável mostrado na Tabela 8.3. O esquema de codificação introduzido está ilustrado na Figura 8.2.

A Figura 8.3 mostra um gráfico da probabilidade de erro por bit versus relação sinal ruído para comparação do esquema de codificação proposto com dois usuários (Figura 8.2) com o

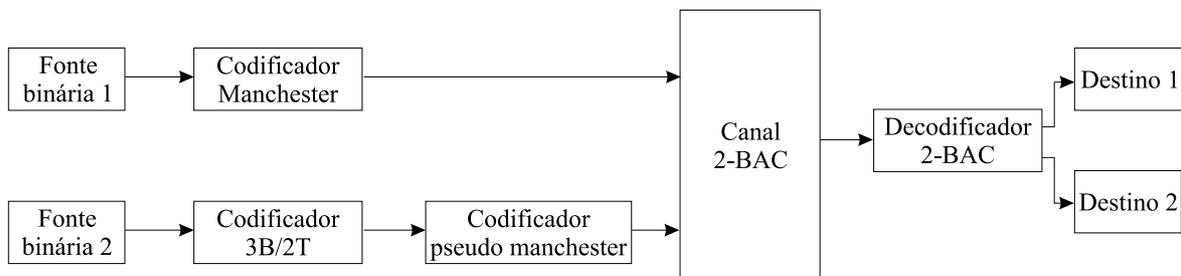


Figura 8.2: Esquema de codificação hierárquica com dois usuários.

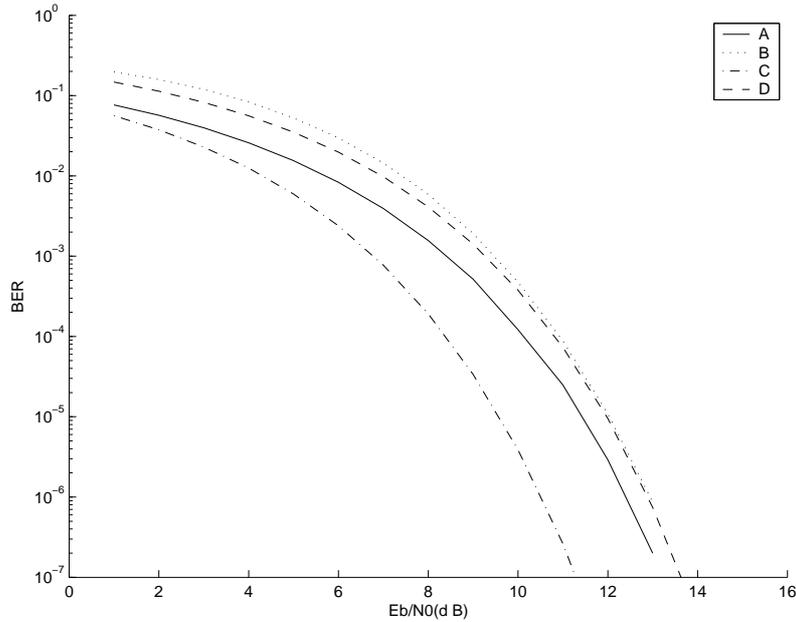


Figura 8.3: A - Usuário 1 (codificação hierárquica), B- Usuário 2 (codificação hierárquica), C- Código Manchester, D- PAM ternário.

código *Manchester* padrão e a modulação ternária (3-PAM) [41], na presença de ruído branco gaussiano. Verifica-se uma perda de desempenho quando compara-se o caso para um usuário e para dois usuários quando é usado o esquema de codificação hierárquica proposto, entretanto a taxa total passa a ser de 1,25 para dois usuários, enquanto para um usuário a taxa resultante é a do código Manchester de 1/2.

Pode-se usar o esquema de codificação serial introduzido na seção 5.3, em que há a concatenação de um código de bloco e de um código turbo para cada usuário, aplicado ao caso de codificação hierárquica conforme ilustrado na Figura 8.4. Considere por simplicidade que o codificador turbo 1 e o codificador turbo 2 são idênticos (Figura 8.5), e cada RSC componente  $\mathcal{C}$  tem matriz geradora polinomial  $G(D) = \left[ 1 \quad \frac{1+D^2}{1+D+D^2} \right]$ . O decodificador utilizado foi introduzido na Seção 6.2 e está ilustrado na Figura 8.4. Após o decodificador turbo estimar a sequência ternária mais provável, o decodificador 2-BAC estima a sequência de informação dos dois usuários usando a decodibilidade única do par de códigos *Manchester* e *pseudo Manchester*.

A Figura 8.6 mostra um gráfico da probabilidade de erro por bit versus relação sinal ruído para comparação do esquema hierárquico proposto com dois usuários concatenado com código turbo para 3 iterações.

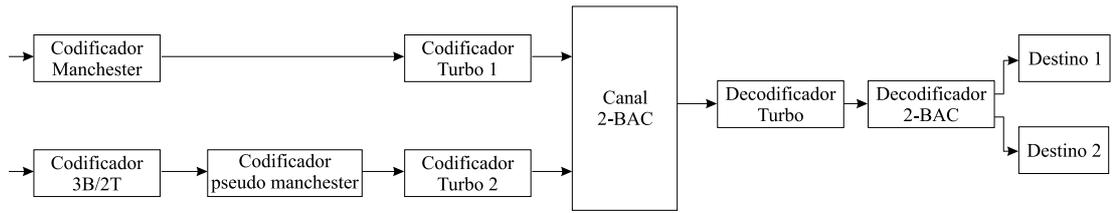


Figura 8.4: Concatenação do esquema de codificação hierárquica com código turbo.

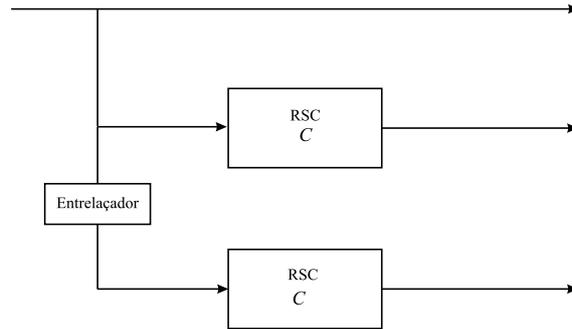


Figura 8.5: Diagrama de blocos representando o codificador turbo com taxa 1/3.

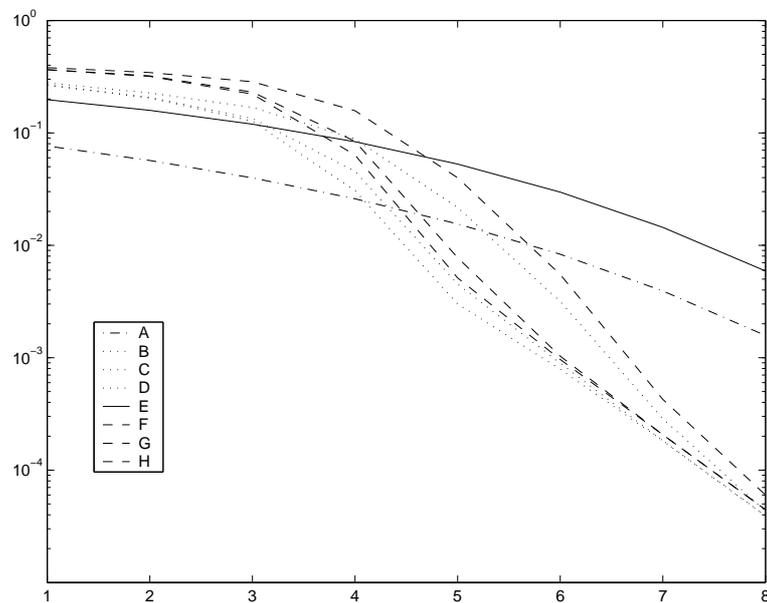


Figura 8.6: A - Usuário 1 (codificação hierárquica), B - Usuário 1 com turbo (1 iteração), C - Usuário 1 com turbo (2 iterações), D- Usuário 1 com turbo (3 iterações) E - Usuário 2 (codificação hierárquica), F - Usuário 2 com turbo (1 iteração), G - Usuário 2 com turbo (2 iterações), H- Usuário 2 com turbo (3 iterações).

## Capítulo 9

# Decodibilidade Única para uma Classe de Códigos Transmitindo por meio do 2-BAC Quase-Síncrono

O 2-BAC é dito síncrono, se os codificadores e o decodificador estão em sincronismo de bloco. Ele é dito quase-síncrono, se os dois codificadores não estão em sincronismo entre si, mas o decodificador sabe a posição do bloco de cada codificador, na seqüência recebida dos símbolos. Ele é dito assíncrono se não existe nenhum sincronismo de bloco entre o codificador e o decodificador [62]. Em todos os três casos, há o sincronismo de símbolo. Quando um par unicamente decodificável é usado no 2-BAC sem ruído, uma correta decodificação é garantida.

Em um 2-BAC quase-síncrono [63], a diferença de fase  $s$  entre duas palavras-código de dois códigos de comprimento de bloco  $n$  é chamada de *slippage*, sendo que  $0 \leq s < n$ . A Figura 9.1 ilustra esta situação. O código para dois usuários  $(C_1, C_2)$  é dito ser unicamente decodificável em um 2-BAC quase-síncrono (QSUD) se ele pode ser unicamente decodificável sob todos os possíveis *slippages*  $s$ . Um par de códigos QSUD garante correta decodificação em um 2-BAC quase-síncrono.

Este capítulo introduz uma condição de decodibilidade única para uma classe de códigos  $(C_1, C_2)$ , em que  $C_1 = \{0^n, 1^n\}$ , transmitindo por meio de um 2-BAC quase-síncrono. A abordagem que leva a esta condição sugere que  $C_2$  deve ser um código de treliça. Primeiramente será usado um exemplo para demonstrar as principais idéias, então será mostrado o caso geral.

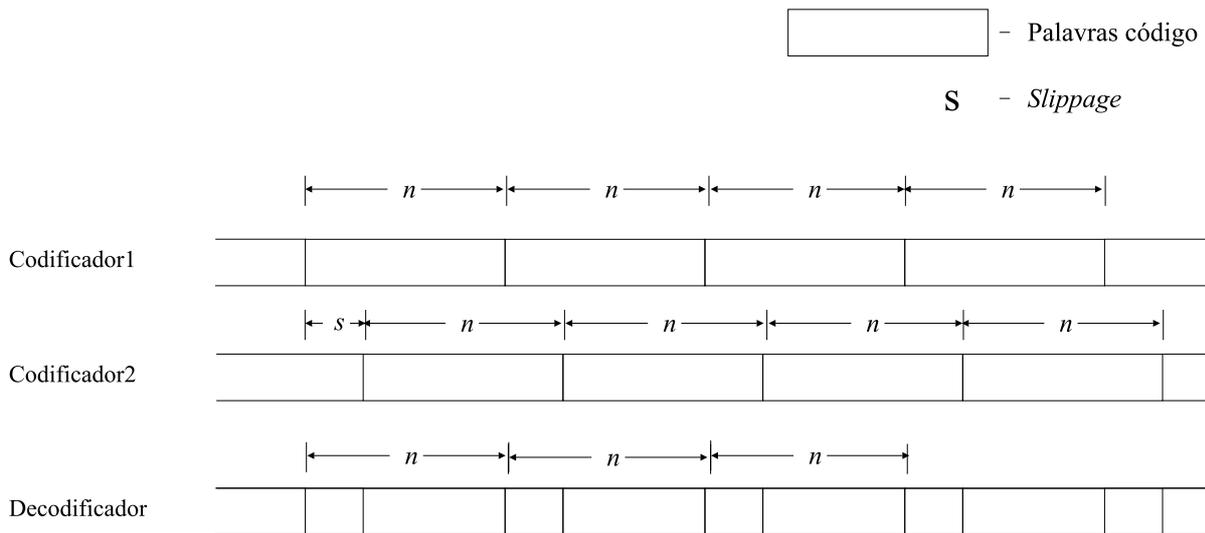


Figura 9.1: 2-BAC quase-síncrono. A diferença de fase  $s$  entre duas palavras-código de dois códigos de comprimento de bloco  $n$  é chamada de *slippage*.

Tabela 9.1:  $C_1 = \{00, 11\}$  e  $C_2 = \{00, 01, 10\}$  é um par de códigos de bloco unicamente decodificáveis para o 2-BAC síncrono.

$C_2 \downarrow \setminus C_1 \rightarrow$	00	11
00	00	11
01	01	12
10	10	21

## 9.1 O Caso para $n = 2$

Considere o esquema ilustrado na Figura 2.6. Seja  $C_1 = \{00, 11\}$  e  $C_2 = \{00, 01, 10\}$  dois códigos de bloco com comprimento de bloco  $n = 2$ . Suponha primeiramente, que os codificadores estão em sincronismo de bloco. Uma vez que todas as palavras são distintas, como ilustrado na Tabela 9.1, o par  $(C_1, C_2)$  é unicamente decodificável no 2-BAC. Suponha agora, que os codificadores não estão em sincronismo de bloco entre si como ilustrado na Figura 9.2. O decodificador recebe o vetor  $(r_k, r_{k+1})$ , em que  $r_k = v_k + w_k$  e  $r_{k+1} = v_{k+1} + w_{k+1}$ . O par  $(v_k, v_{k+1})$  é uma palavra código de  $C_1$ , mas o par  $(w_k, w_{k+1})$  é constituído por um símbolo da palavra código  $(w_{k-1}, w_k) \in C_2$  e um símbolo da palavra código  $(w_{k+1}, w_{k+2}) \in C_2$ . A diferença de fase entre as palavras código dos dois codificadores é  $s = 1$ . A Figura 9.3 mostra todos os possíveis vetores recebidos  $(r_k, r_{k+1})$  no decodificador e as palavras código correspondentes do codificador 1  $(v_k, v_{k+1})$  e os pares correspondentes  $(w_k, w_{k+1})$  do codificador 2. Como o par  $(C_1, C_2)$  não pode ser decodificado de maneira única sob todos os possíveis *slippages*  $s$ , este

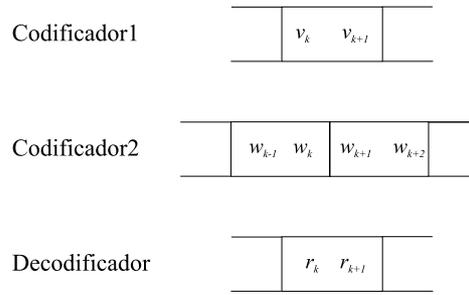


Figura 9.2: 2-BAC quase-síncrono para código com comprimento de bloco 2. A diferença de fase entre duas palavras-código dos dois códigos é  $s = 1$ . O par  $(v_k, v_{k+1})$  representa uma palavra código do usuário 1. Os pares  $(w_{k-1}, w_k)$  e  $(w_{k+1}, w_{k+2})$  representam duas palavras-código do usuário 2.

código para dois usuários não é QSUD. Este fato acontece porque o vetor  $(r_k = 1, r_{k+1} = 1) = (v_k = 0, v_{k+1} = 0) + (w_k = 1, w_{k+1} = 1) = (v_k = 1, v_{k+1} = 1) + (w_k = 0, w_{k+1} = 0)$  então o decodificador não consegue resolver a ambigüidade de entregar  $(v_k = 0, v_{k+1} = 0)$  para o usuário 1 e  $(w_k = 1, w_{k+1} = 1)$  para o usuário 2 ou entregar  $(v_k = 1, v_{k+1} = 1)$  para o usuário 1 e  $(w_k = 0, w_{k+1} = 0)$  para o usuário 2.

Suponha agora que em vez de  $C_2$  ser o código de bloco  $C_2 = \{00, 01, 10\}$ , o codificador para  $C_2$  tem a estrutura de treliça mostrada na Figura 9.4. Como o vetor recebido  $(r_k = 1, r_{k+1} = 1)$  resulta de  $(v_k = 1, v_{k+1} = 1) + (w_k = 0, w_{k+1} = 0)$ , e a soma  $(v_k = 0, v_{k+1} = 0) + (w_k = 1, w_{k+1} = 1)$  nunca poderá acontecer, não existe ambigüidade. Assim, o par  $(C_1, C_2)$  é unicamente decodificável em relação a todos os possíveis *slippages*  $s = 1$  e neste caso o código para dois usuários é QSUD.

## 9.2 O Caso Geral

Seja  $C_1 = \{0^n, 1^n\}$  um código de bloco com comprimento de bloco  $n$ , formado por duas  $n$ -uplas binárias, em que  $0^n$  é uma palavra-código com todos os elementos iguais a 0 e  $1^n$  é uma palavra código com todos os elementos iguais a 1. Seja  $C_2$  um código de bloco com comprimento  $n$  constituído por todas as  $n$ -uplas binárias, com exceção da  $n$ -upla  $1^n$ . É bem conhecido que este par é unicamente decodificável para o 2-BAC, contudo este par não é QSUD.

**Proposição 9.2.1** *Considere um par de códigos  $(C_1, C_2)$ , com comprimento  $n$ , em que  $C_1 = \{0^n, 1^n\}$ . Seja  $t$  o número de símbolos 1's consecutivos em cada par concatenado de palavras código de  $C_2$ . O par  $(C_1, C_2)$  é QSUD se e somente se  $t < n$ .*

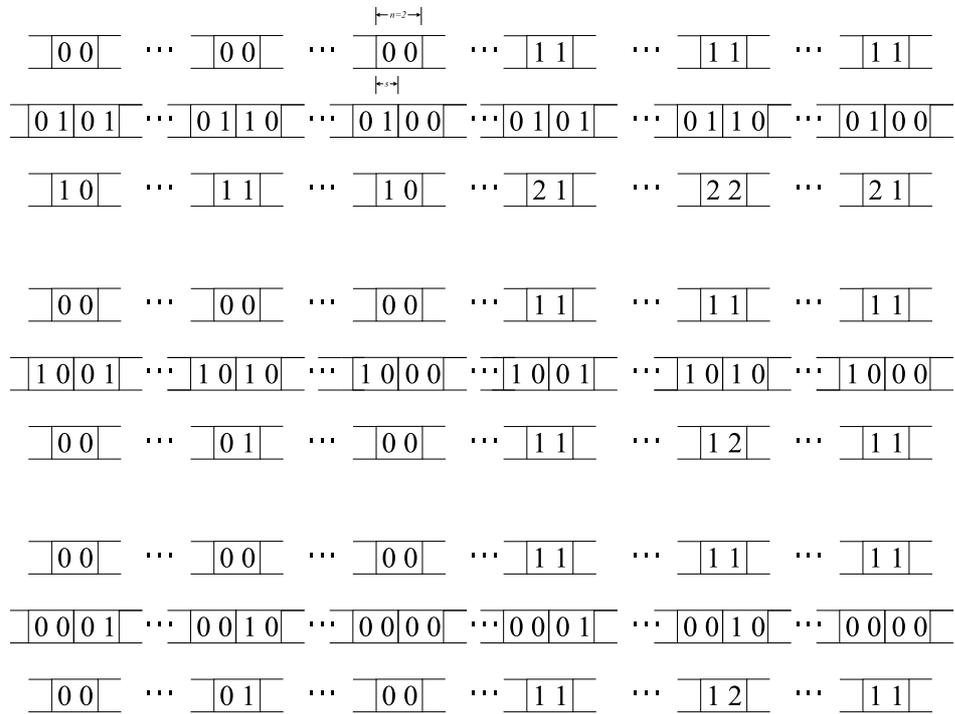


Figura 9.3: Todos os possíveis vetores recebidos  $(r_k, r_{k+1})$  no decodificador e as palavras código correspondentes do codificador 1  $(v_k, v_{k+1})$  e os pares correspondentes  $(w_k, w_{k+1})$  do codificador 2.

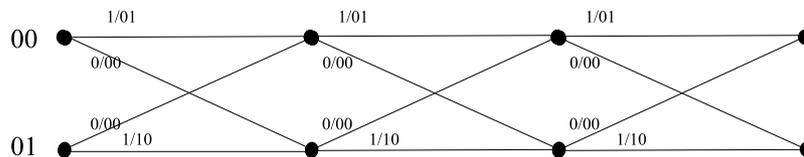


Figura 9.4: Código de treliça para  $C_2$  com sub-blocos de comprimento 2.

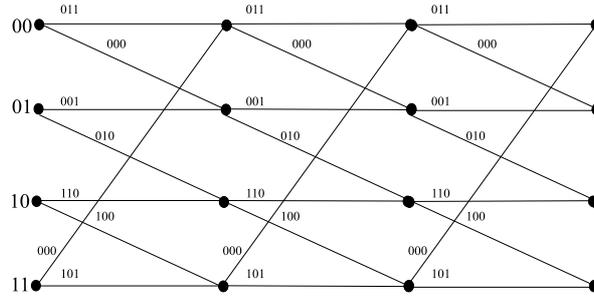


Figura 9.5: Código de treliça para  $C_2$  com comprimento de bloco 3.

**Prova:**

A prova segue mostrando primeiramente que contrariando a hipótese, se  $t \geq n$  e  $0 \leq s < n$ , então para alguns valores de  $s$  tem-se as possibilidades  $(r_k = 1, r_{k+1} = 1, \dots, r_{k+n} = 1) = (v_k = 0, v_{k+1} = 0, \dots, v_{k+n} = 0) + (w_k = 1, w_{k+1} = 1, \dots, w_{k+n} = 1) = (v_k = 1, v_{k+1} = 1, \dots, v_{k+n} = 1) + (w_k = 0, w_{k+1} = 0, \dots, w_{k+n} = 0)$  e o decodificador não consegue resolver a ambigüidade de entregar  $(v_k = 0, v_{k+1} = 0, \dots, v_{k+n} = 0)$  para o usuário 1 e  $(w_k = 1, w_{k+1} = 1, \dots, w_{k+n} = 0)$  para o usuário 2 ou entregar  $(v_k = 1, v_{k+1} = 1, \dots, v_{k+n} = 1)$  para o usuário 1 e  $(w_k = 0, w_{k+1} = 0, \dots, w_{k+n} = 0)$  para o usuário 2.

Suponha agora que  $t < n$ . Segue que  $(r_k = 1, r_{k+1} = 1, \dots, r_{k+n} = 1)$  sempre resultará de  $(v_k = 1, v_{k+1} = 1, \dots, v_{k+n} = 1) + (w_k = 0, w_{k+1} = 0, \dots, w_{k+n} = 0)$  uma vez que a seqüência  $(w_k = 1, w_{k+1} = 1, \dots, w_{k+n} = 1)$  não existe, então nao haverá ambigüidade e o par  $(C_1, C_2)$  é QSUD. □

A *Proposição 9.2.1* especifica uma condição necessária e suficiente para decodibilidade única para um 2-BAC quase-síncrono se  $C_1 = \{0^n, 1^n\}$ . A abordagem que leva à *Proposição 9.2.1* sugere que  $C_2$  deva ser um código de treliça, isto é, um código com memória, não restrito a ser linear.

**Exemplo 9.2.1** *Seja  $C_1 = \{000, 111\}$  um código de bloco com comprimento de bloco 3 e seja  $C_2$  um código de treliça, cuja treliça está ilustrada na Figura 9.5. Uma vez que  $t < 3$ , o par  $(C_1, C_2)$  é QSUD.*

### 9.3 Resultado das Simulações

A Figura 9.6 mostra as curvas da probabilidade de erro versus relação sinal ruído para comparação do caso em que  $C_1 = \{00, 11\}$  e  $C_2 = \{00, 01, 10\}$  para um 2-BAC quase-síncrono e

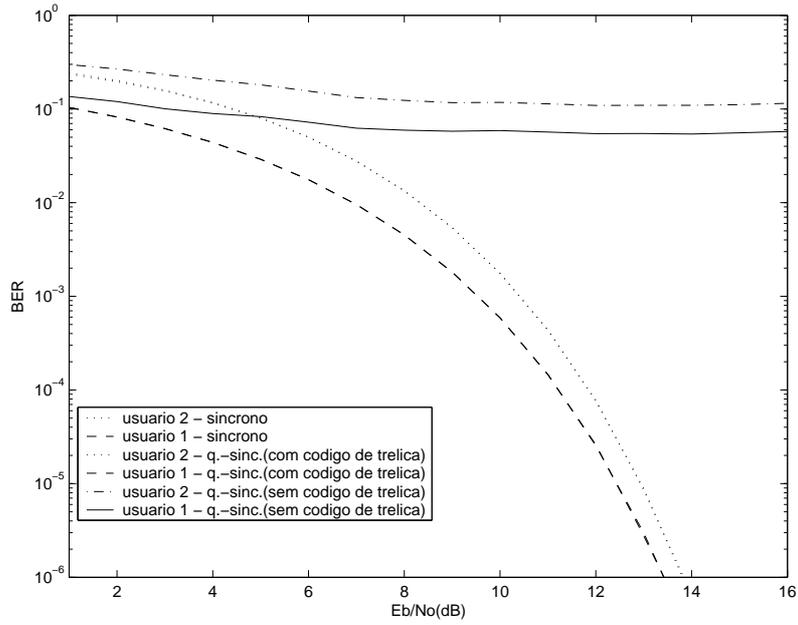


Figura 9.6: 2-BAC síncrono  $\times$  2-BAC quase-síncrono. Curvas da probabilidade de erro versus relação sinal ruído para comparação do caso em que  $C_1 = \{00, 11\}$  e  $C_2 = \{00, 01, 10\}$  para um 2-BAC quase-síncrono e o caso em que  $C_1 = \{00, 11\}$  e o codificador para  $C_2$  tem a estrutura de treliça mostrada na Figura 9.4 para um 2-BAC síncrono e um 2-BAC quase- síncrono.

o caso em que  $C_1 = \{00, 11\}$  e o codificador para  $C_2$  tem a estrutura de treliça mostrada na Figura 9.4 para um 2-BAC síncrono e um 2-BAC quase- síncrono.

# Capítulo 10

## Conclusões

Nesta tese, as condições de decodibilidade única encontradas, juntamente com os sistemas de codificação/decodificação turbo permitem a utilização prática de códigos para o 2-BAC, isto é, possibilitam a obtenção de baixas probabilidades de erro na saída do decodificador, com baixa complexidade. Até então, não encontrava-se na literatura técnica específica, artigos descrevendo tal tipo de aplicação.

No *Capítulo 5*, foi introduzida uma condição para decodibilidade única com uso de códigos de treliça para o 2-BAC. Foi apresentada uma construção de códigos de treliça baseada em um par de códigos convolucionais  $(\mathcal{C}_1, \mathcal{C}_2)$  e um par de códigos de bloco  $(\mathbf{C}_1, \mathbf{C}_2)$ . O uso de um par de códigos de bloco unicamente decodificáveis para o 2-BAC, elimina caminhos através das treliças dos códigos convolucionais empregados, evitando assim problemas de ambigüidade na decodificação. Essa abordagem em princípio não limita a taxa resultante para o 2-BAC. A partir desta condição é possível uma visualização clara da utilização de códigos convolucionais (lineares) ou de códigos de treliça (não-lineares) para o 2-BAC. Ressalta-se que a baixa taxa obtida no *Exemplo 5.3.1* foi resultado, principalmente da taxa  $1/2$  do código convolucional empregado. Apenas para ilustração, se for utilizado um código convolucional sistemático com taxa  $4/5$  e o mesmo par de códigos de bloco  $\mathbf{C}_1 = \{00, 11\}$  e  $\mathbf{C}_2 = \{00, 01, 10\}$  a taxa resultante para o 2-BAC será 1,032.

No *Capítulo 6* foi introduzida a decodificação iterativa para o 2-BAC. O esquema de codificação descrito no Capítulo 5 (Figura 5.3) é adaptado com a substituição do par de códigos convolucionais por um par de códigos turbo, isto é, cada código convolucional é substituído por um par de códigos convolucionais recursivos sistemáticos separados por um entrelaçador. O decodificador utilizado, usa a decodificação iterativa para estimar a seqüência ternária mais

provável e em seguida usa um segundo decodificador para separar as informações relativas aos dois usuários. São apresentados resultados de simulações (*Exemplos 6.2.1 e 6.2.2*) nos quais podem ser observadas melhorias quando do uso da decodificação iterativa, bem como quando do aumento da memória dos codificadores convolucionais empregados.

O *Capítulo 7* introduz um novo esquema de codificação CCMA para o 2-BAC (Figura 7.4), empregando uma concatenação serial de códigos de bloco. O código resultante da adição das palavras de dois códigos produto (um de cada usuário), é empregado como um código turbo, isto é, as mensagens são codificadas horizontalmente, entrelaçadas e então codificadas verticalmente. Também foi introduzido um esquema de codificação com uso de um par de códigos de bloco unicamente decodificáveis para o 2-BAC e um código de bloco ternário (Figura 7.6), neste caso, o código ternário é empregado como um código turbo. Para ambas construções o decodificador horizontal entrega as probabilidades *a priori* para o decodificador vertical, similarmente, o decodificador vertical entrega valores *a priori* para o decodificador horizontal, permitindo um substancial aumento de desempenho. Foram apresentados exemplos bastante simples para demonstrar uma aplicação destas técnicas, resultando em um ganho de 1dB (*Exemplos 7.3.1 e 7.3.2*) com apenas duas iterações. Códigos produto mais potentes podem ser empregados, podendo ser usado o algoritmo BCJR [40].

No *Capítulo 8* foi proposto um esquema de codificação que permite um acréscimo do tráfego total em redes *Ethernet*, mantendo compatibilidade com os usuários já existentes. Novos usuários serão adicionados usando o esquema de codificação proposto, enquanto os usuários existentes não precisarão mudar de *hardware*.

O *Capítulo 9* estabelece uma condição de decodibilidade única para o 2-BAC quase-síncrono sendo apresentada uma classe de códigos satisfazendo esta condição. Resultados de simulações foram apresentados indicando desempenho idêntico para os casos síncrono e quase-síncrono na presença de ruído branco gaussiano.

## 10.1 Sugestões para Trabalhos Futuros

Os esquemas de codificação com concatenação serial, apresentados nesta tese, são importantes uma vez que possibilitam o uso da decodificação iterativa para o 2-BAC conforme introduzido nos Capítulos 5, 6 e 7, fazendo uso de códigos de treliça ou de códigos produto. Entretanto, na prática há situações em que é desejável operar com mais de dois usuários. A generalização destes esquemas, para o caso em que há  $T$  ( $T > 2$ ) usuários é possível, conforme ilustrado na

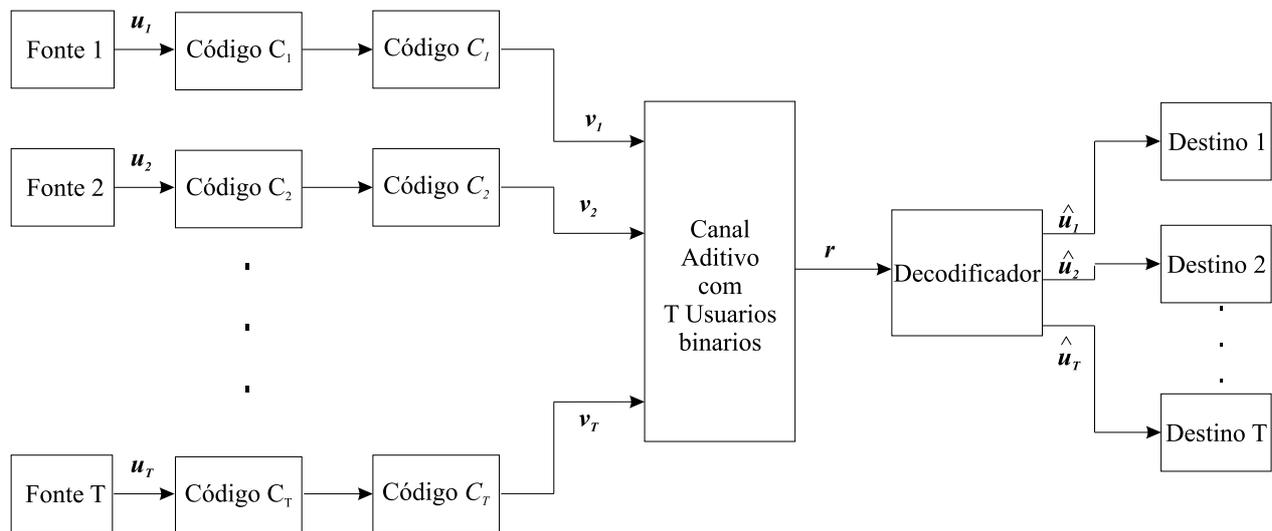


Figura 10.1: Esquema de concatenação serial para uso em canal aditivo com  $T$  usuários binários.

Figura 10.1. Neste caso, o codificador para cada usuário será composto por uma concatenação de um código de bloco com códigos Turbo ou códigos produto. A complexidade no decodificador entretanto, cresce bastante, uma vez que se a treliça para 1 usuário tiver  $M$  estados, a treliça resultante para  $T$  usuários terá  $M^T$  estados. Em outras palavras, a simples extensão do modelo atual aumenta bastante a complexidade da treliça. Uma alternativa seria expandir o modelo atual usando espalhamento espectral. Cada par de usuários receberia uma mesma seqüência de espalhamento. Teríamos um sistema com uso conjunto de esquemas CCMA e CDMA. Uma outra possibilidade para expansão para  $T$  usuários seria obtida combinando CCMA com *Coded Orthogonal Frequency Division Multiplexing* (COFDM).

# Referências Bibliográficas

- [1] SHANNON, C.E. A Mathematical Theory of Communication. *Bell Syst. Tec. Jour.*, v. 27, pp.379-423, 623-656, July 1948.
- [2] LIN, S.; COSTELLO JR., D. *Error Control Coding: Fundamentals and Applications*. New Jersey, USA: Prentice-Hall Inc., 1983
- [3] MICHELSON, A. M.; LEVESQUE, A. H. *Error Control Techniques for Digital Communication*. John Wiley & Sons, 1984.
- [4] HEEGARD, C.; WICKER, S. B. *Turbo Coding*. Boston, Dorderecht, London: Kluwer Academic Publishers, 2001
- [5] JOHANNESSON, R.; ZIGANGIROV, K. SH. *Fundamentals of Convolutional Coding*. New York, USA: IEEE series on Digital & mobile communication, The Institute of Electrical and Eletronics Engineers, Inc., 1999.
- [6] VUCETIC, B.; YUAN, J. *Turbo Codes Principles and Applications*. Boston, Dorderecht: Kluwer Academic Publishers, 2001.
- [7] SHANNON, C. E. Two-way Communication Channels. In 4th BERKELEY SYMP. MATH. STAT. PROB (1961). Proceedings. v. 1, pp.611-644, 1961. Reprinted in *Key Papers in the Development of Information Theory.*, SLEPIAN, D. Ed. New York, IEEE Press, pp.339-372, 1974
- [8] CABRAL, H. A. *Codificação para Canal de Acesso Múltiplo Síncrono*. Recife, Brasil, 1994. Dissertação (Mestrado em Engenharia Elétrica)- Departamento de Eletrônica e Sistemas, UFPE.
- [9] ALCOFORADO, M. L. M. G. *Implementação Algorítmica de Códigos Lineares para o Canal Aditivo com Dois Usuários Binários*. Recife, Brasil, 1999. Dissertação (Mestrado em Engenharia Elétrica)- Departamento de Eletrônica e Sistemas, UFPE.

- [10] FAN, P. Z.; DARNELL, M.; HONARY, B. Superimposed Codes for the Multiaccess Binary Adder Channel. *IEEE Trans. Inform. Theory*, v. 41, N° 4, pp.1178-1182, 1995.
- [11] AHLWEDE, R. Multi-Way Communication Channels. In: 2nd INT. SYMP. INFORMATION THEORY (1971: Tsahkadsor, Armenian S.S.R). *Proceedings*. Tsahkadsor, Armenian S.S.R 1971.
- [12] LIAO, H. A Coding Theorem for Multiple Access Communications. In INT. SYMP. INFORMATION THEORY (1972: Asilomar, CA,). *Proceedings*. Asilomar, CA, 1972
- [13] LIAO, H. *Multiple Access Channels*. USA, 1972. Tese (Doutorado)- Dept. Elec. Eng., University of Hawaii.
- [14] GALLAGER, R. G. A Perspective on Multiaccess Channels. *IEEE Trans. Inform. Theory*, v. IT-31, N° 2, pp.124-142, 1985.
- [15] KASAMI, T.; LIN, S. Coding for a Multiple-Access Channel. *IEEE Trans. Inform. Theory*, v. IT-22, N° 2, pp.129-137, 1976.
- [16] KASAMI, T.; LIN, S. Bounds on the Achievable Rates of Block Coding for a Memoryless Multiple-Access Channel. *IEEE Trans. Inform. Theory*, v. IT-24, N° 2, pp.187-197, March 1978.
- [17] KASAMI, T.; LIN, S. Decoding of Linear Delta-Decodable Codes for a Multiple-access Channel. *IEEE Trans. Inform. Theory*, v. IT-24, N° 2, pp.187-197, March 1978.
- [18] DEAETT, M. A.; WOLF, J. K. Some Very Simple Codes for the Nonsynchronized Two-User Multiple-Access Adder Channel with Binary Inputs. *IEEE Trans. Inform. Theory*, v. IT-24, N° 5, pp.635-636, September 1978.
- [19] WELDON, E. J. Coding for a Multiple-Access Channel. *Information and Control* 36, pp.256-274, 1978.
- [20] DA ROCHA JR., V. C. Seminar About Codes for 2-BAC. *Depto. de Eletrônica e Sistemas*, UFPE, 1995.
- [21] DA ROCHA JR., V. C.; MASSEY, J. L. A New Approach to the Design of Codes for the Binary Adder Channel. *Cryptography and Coding III, IMA Conf. Series, New Series*, Oxford: Ed. M.J. Ganley N° 45, 1993, pp.179-185.

- [22] AHLWEDE, R.; BALAKIRSKY, V. B. Construction of Uniquely Decodable Codes for the Two-User Binary Adder Channel. *IEEE Trans. Inform. Theory*, v. 45, N° 1, pp. 326-330, January 1999.
- [23] VAN TILBORG, C. A. An Upper Bounds for Codes in a Two-Access Binary Erasure Channel. *IEEE Trans. Inform. Theory*, v. IT-24, pp.113-116, January 1978.
- [24] MARKARIAN, G.; HONARY, B.; BENACHOUR, P. Trellis Decoding Technique for Binary Adder Channel with M Users and its Application in LANs. In: IEE PROC. COMMUN. (April 1997). *Proceedings*, v. 144, N° 4, April 1997, pp.65-69.
- [25] SIDORENKO, V.; MARKARIAN, G.; HONARY, B. Minimal Trellis Design for Linear Codes Based on the Shannon Product. *IEEE Trans. Inform. Theory*, v. 42, N° 6, pp.2048-2053, November 1996.
- [26] RIMOLDI, B.; URBANKE, R. A Rate-Splitting Approach to the Gaussian Multiple-Access Channel. *IEEE Trans. Inform. Theory*, v. 42, N° 2, pp.364-375, March 1996.
- [27] GRANT, A. J.; RIMOLDI, B.; URBANKE, R. L.; WHITING, P. A. Rate-Splitting Multiple Access for Discrete Memoryless Channels. *IEEE Trans. Inform. Theory*, v. 47, N° 3, pp.873-890, March 2001.
- [28] CHANG, S. C.; WELDON JR., E. J. Coding for T-User Multiple-Access Channels. *IEEE Trans. Inform. Theory*, v. IT-25, pp.684-691, November 1979.
- [29] ALI, F. H.; HONARY, B. Soft Decision Decoding Technique for Collaborative Coding Multiple-Access Channels. In: THIRD IEE CONF. ON TELECOMMUNICATION (March 1991). 1991.
- [30] ALI, F. H.; HONARY, B. Low Complexity Soft Decision Decoding Technique for T-User Collaborative Coding Multiple-Access Channels. *Electronic Letters* , pp.1167-1169, 1991.
- [31] ALI, F. H.; HONARY, B. Collaborative Coding and Decoding Techniques for Multiple Access Channel. In: IEE PROC. COMMUN. (1994). *Proceedings*, v. 141, N° 2, 1994, pp.56-62.
- [32] HONARY, B.; ALI, F. H.; DARNELL, M. Capacity of T User Collaborative Coding Multiple-Access Scheme Operating Over a Noisy Channel. *Electronic Letters* ,25, (11), pp.742-744, 1989.

- [33] HANSELMAN, D.; LITTLEFIELD, B. *Matlab, Guia do Usuário*. São Paulo, Brasil: Makron Books, 1999.
- [34] FORNEY JR., G. D. *Concatenated Codes*. Cambridge, MA:MIT. Press, 1996.
- [35] YUEN, J. H. et al. Modulation and Coding for Satellite and Space Communications. *IEEE Proceedings*, v. 78, pp. 1250-65 ,July 1990.
- [36] VITERBI, A. J. Convolutional Codes and Their Performance in Communication Systems. *IEEE Trans.Commun.*, v. COM-19, pp.751-772, October 1971.
- [37] FORNEY JR., G. D. The Viterbi Algorithm. *IEEE Proceedings*, v. 68, pp.268-278, March 1973.
- [38] HAGENAUER, J.; HOHER, D. P. A Viterbi Algorithm with Soft-Decision Outputs and its Applications. In: GLOBECOM'89 (November 1989), v. 3, November 1989, pp. 1680-1686.
- [39] BERROU, C.; GLAVIEUX, A.; THITIMAJSHIMA, P. Near Shannon Limit, Error-Correcting Coding and Decoding: Turbo Codes. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC'93) (May 1993), v. 2/3, May 1993, pp.1064-1071.
- [40] BAHL, L. R.; COCKE, J.; JELINEK, F.; RAVIV, J. Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. *IEEE Trans. Inform. Theory*, v. IT-20, pp.284-287, March 1974.
- [41] PROAKIS, J. G. *Digital Communication*. New York, USA: Mc. Graw Hill Inc., 1995.
- [42] ABRAMSON, N. The ALOHA System - Another Alternative for Computer Communications. In: FALL JOINT COMPUTER CONF. (1970). *Proceedings*. AFIPS Press v. 37, 1970, pp.281-285.
- [43] TANENBAUM, A. S. *Computer Networks*. New Jersey, USA: Prentice-Hall, 1996.
- [44] CAPETANAKIS, J. A Protocol for Resolving Conflicts on ALOHA Channels. In: IEEE INT. SYMP. INFO. TH. (March 1985). *Abstracts of papers*, v. IT-31, March 1985, pp.176-184, .
- [45] TSYBAKOV, B. S.; MIKHAILOV, V. A. Slotted Multiaccess Packet Broadcasting Feedback Channel. *Probl. Peredachi Inform.*, v. 14, pp.32-59, October 1978.

- [46] KAUTZ, W. H.; SINGLETON, R. C. Nonrandom Binary Superimposed Codes. *IEEE Trans. Inform. Theory*, v. IT-10, N° 4, pp.363-377, 1964.
- [47] CHIEN, R. T.; FRAZER, W. D. No Application of Coding Theory to Document Retrieval. *IEEE Trans. Inform. Theory*, v. IT-12, N° 2, pp.92-96, 1966.
- [48] ERICSON, T.; LEVENSHTAIN, V. Superimposed Codes in Hamming Space. In: IEEE INT. SYMP. ON INFORM. THEORY (1996). v. IT-12, 1996, pp.92-96.
- [49] ELIAS, P. Error-Free Coding. *IRE Trans. Inform. Theory*, v. IT-4, pp.29-37, 1954.
- [50] FORNEY JR., G. D. Convolutional Codes I: Algebraic Structure. *IEEE Trans. Inform. Theory*, v. IT-16, N° 6, pp.720-738, November 1970 e v. IT-17, N° 3, pp.360, May 1971.
- [51] HAGENAUER, J. Iterative Decoding of Binary Block and Convolutional Codes. *IEEE Trans. Inform. Theory*, v. 42, N° 2, pp.429-445, March 1996.
- [52] BERROU, C.; GLAVIEUX, A. Near Optimum Error Correcting Coding and Decoding: Turbo-Codes. *IEEE Trans. Commun.*, v. 44, N° 10, pp.1261-1271, October 1996.
- [53] SKLAR, B. A Primer on Turbo Code Concepts. *IEEE Commun. Magazine*, pp.94-102, December 1997.
- [54] CLARK JR. G. C.; CAIN, J. B. *Error Correction Coding for Digital Communications*.
- [55] THITIMAJSHIMA, P. *Les Codes Convolutifs Récursifs Systématiques Et Leur Application À La Concaténation Parallèle*. França, 1993. Tese (Doutorado em Eletrônica)- l'Université de Bretagne Occidentale.
- [56] PETERSON, R.; COSTELLO JR., D. J. Binary Convolutional Codes for a Multiple-Access Channel. *IEEE Trans. Inform. Theory*, v. 25, N° 1, pp.101-105, January 1979.
- [57] DA ROCHA JR. V.C.; ALCOFORADO, M.L.M.G. Trellis Code Construction for the 2-User Binary Adder Channel. In: 11<sup>th</sup> INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS (August 2004: Fortaleza, Ceará, Brazil). *Proceedings*. Fortaleza, Ceará, Brazil, Editors J. Neuman and P. Dini, Springer Verlag, 1-5 August 2004, .
- [58] ALCOFORADO, M.L.M.G.; DA ROCHA JR., V. C. Construção de Códigos de Treliça para o Canal Aditivo com dois Usuários Binários. In: XXI SIMPÓSIO BRASILEIRO DE

TELECOMUNICAÇÕES (2004: Belém, Pará, Brazil). *Anais*. Belém, Pará, Brazil , 6-9 September 2004.

- [59] DA ROCHA JR., V.C.; ALCOFORADO, M.L.M.G. Uniquely Decodable Trellis Codes for the 2-User Binary Adder Channel. In: INTERNATIONAL SYMPOSIUM ON INFORMATION THEORY AND ITS APPLICATIONS, ISITA 2004 (Parma, Italy: 2004). Parma, Italy, 10-13 October 2004.
- [60] METCALFE, R.M.; BOGGS, D.R. Ethernet: Distributed Packet Switching for Local Computer Networks. In: COMMUNICATIONS OF THE ACM (1993). v.19, No.7, July 1976. Reprinted in *Multiple Access Communications*, Editor N. Abramson, IEEE Press, pp.379-398, 1993.
- [61] LATHI B.P. *Modern Digital and Analog Communication Systems*. New York: Holt-Saunders International Editions, 1983.
- [62] VERDÚ, S. The Capacity Region of the Symbol-Asynchronous Gaussian Multiple-Access Channel. *IEEE Trans. Inform. Theory*, v. 35, N° 4, pp.733-751, July 1989.
- [63] LIN, S.; WEI, V. K. Nonhomogeneous Trellis Codes for the Quasi-Synchronous Multiple Access Binary Adder Channel with Two Users. *IEEE Trans. Inform. Theory*, v. IT-32, N° 6, pp.129-137, November 1986.
- [64] DA ROCHA JR.; ALCOFORADO, M.L.M.G.; MARKARIAN, G. Iterative Decoding for the 2-User Binary Adder Channel. In: FIRST INTERNATIONAL SYMPOSIUM ON BROADBAND COMMUNICATIONS, ISBC 2004 ( December 2004: Harrogate, UK). Harrogate, UK, 2004.
- [65] DA ROCHA JR.; MARKARIAN, G.; ALCOFORADO, M.L.M.G. Coding for Higher Throughout Ethernet. In: FIRST INTERNATIONAL SYMPOSIUM ON BROADBAND COMMUNICATIONS, ISBC 2004 ( December 2004: Harrogate, UK). Harrogate, UK, 2004.