Vagner Vale do Nascimento

Decodificação Iterativa de Códigos Turbo-Produto q-ários em um Canal FFH-CDMA

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Orientador: Jaime Portugheis

Campinas, SP
2007

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

Nascimento, Vagner Vale do
N17d Decodificação iterativa de códigos turbo-produto q-ários em um Canal FFH-CDMA / Vagner Vale do Nascimento. – Campinas, SP: [s.n.], 2007.

Orientador: Jaime Portugheis.
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Teoria da codificação. I. Portugheis, Jaime. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título

Título em Inglês: Iterative decoding of q-ary turbo-product codes in FFH-CDMA systems
Palavras-chave em Inglês: Coding theory
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca Examinadora: Daniel C. Cunha, Danilo Zanatta Filho e Renato Baldini Filho
Data da defesa: 19/12/2007
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Vagner Vale do Nascimento

Data da Defesa: 19 de dezembro de 2007

Título da Tese: "Decodificação Iterativa de Códigos Turbo-Produto Q-ários em um Canal FFH-CDMA"

Prof. Dr. Jaime Portugheis (Presidente): Jaime Portugheis

Prof. Dr. Daniel Carvalho da Cunha:

Dr. Danilo Zanatta Filho: Danilo Zanatta Filho

Prof. Dr. Renato Baldini Filho: R. Baldini Jr.

Resumo

Este trabalho apresenta um estudo sobre a decodificação turbo q-ária em sistemas FFH-CDMA. Para alcançar o principal objetivo do estudo, foi desenvolvido um algoritmo para implementar a decodificação suave q-ária. Este algoritmo se baseia em uma adaptação do método de decodificação de Chase para suportar alfabetos q-ários. A proposta de decodificação iterativa (turbo) define um procedimento para viabilizar a realimentação dos símbolos q-ários decodificados e de suas respectivas confiabilidades em cada iteração do decodificador. Simulações foram realizadas considerando canais gaussianos e FFH-CDMA. Os resultados obtidos nas simulações demonstram uma considerável melhoria no desempenho dos sistemas com a decodificação turbo q-ária, sem comprometer a complexidade. Entretanto, a utilização dos códigos produto reduzem a eficiência espectral do sistema, sendo necessário, assim, compensá-la através do aumento do alfabeto do código q-ário.

Palavras-chave: Códigos q-ários, Códigos Produto, Decodificação Suave, Decodificação Turbo, SISO, Sistemas FFH-CDMA.

Abstract

This work presents a study of q-ary turbo decoding applied to FFH-CDMA systems. To attain the main objective of this dissertation, an algorithm to implement the q-ary soft decoding was designed. Specifically, an adaptation of the Chase decoding method was proposed to comply with q-ary symbols requirements. The proposal of iterative (turbo) decoding makes possible the feedback of the decoded q-ary symbols and its reliabilities in each decoder iteration. Simulations were done considering Gaussian and FFH-CDMA channels. The results obtained in the simulations indicate a better system's performance when the q-ary turbo decoding is applied, without compromising complexity. However, the use of product codes decreases the system's spectral efficiency. Thus, it is necessary to increase the q-ary code's alphabet in order to mitigate this decrease.

Keywords: Non-Binary Codes, Product Codes, Soft Decoding, Turbo Decoding, SISO, FFH-CDMA.

Agradecimentos

Ao meu orientador, Prof. Jaime Portugheis, sou grato pela orientação e pela compreensão das minhas decisões particulares que postergaram a conclusão deste mestrado.

Ao colega Daniel Carvalho da Cunha, pela ajuda em algumas simulações realizadas neste trabalho.

Aos amigos Giuliano Gadioli La Guardia, João Coelho, João Henrique Kleinschmidt e Lourival Aparecido de Góis, pelos estudos em grupo na Unicamp e pelas diversões.

Aos meus pais, pelo apoio e incentivo durante toda esta jornada.

À minha esposa, pela compreensão e apoio neste cansativo período, mas compensador, dividindo-me entre o mestrado e o trabalho.

À CAPES, pelo apoio financeiro durante parte deste período.

Sumário

Lista de Figuras **xiii**

Lista de Tabelas **xvii**

1 Introdução **1**

1.1 Comunicações Digitais 1

1.2 Contexto e objetivos do trabalho 2

1.3 Organização da tese 3

1.4 Resumo das principais contribuições 4

2 Fundamentação Teórica **5**

2.1 Introdução 5

2.2 Elementos de um Sistema de Comunicação Digital 6

2.2.1 Fonte de Informação 6

2.2.2 Codificador de Canal 7

2.2.3 Modulador Digital 7

2.2.4 Canal de Transmissão 8

2.3 Modelos de Canais 8

2.3.1 Canal Discreto no Tempo 8

2.3.2 Canal AWGN 9

2.3.3 Outros Tipos de Canais 9

2.4 Capacidade de Canal 10

2.4.1 Entropia e Informação Mútua 10

2.4.2 Teorema da Capacidade de Canal 13

2.4.3 Taxa de corte 14
2.5 Códigos de Bloco Lineares 15
2.5.1 Matriz de Verificação de Paridade 17
2.5.2 Distância Mínima 17
2.5.3 Códigos Cíclicos 18
2.5.4 Decodificação de Códigos de Bloco 20

3 Decodificação Suave q-ária 23
3.1 Introdução 23
3.2 Decodificação de Chase Binária - Método Original 26
3.2.1 Conceitos da Decodificação de Chase 26
3.2.2 Geração dos Padrões de Teste 32
3.2.3 Cálculo da Confiabilidade 35
3.2.4 Resultados 37
3.3 Algoritmo de Chase q-ário - Método Adaptado 41
3.3.1 Geração dos Padrões de Teste q-ários 44
3.3.2 Cálculo da Confiabilidade q-ária 46
3.4 Análise dos Resultados 48

4 Sistemas FFH-CDMA 53
4.1 Sistemas com Espalhamento Espectral 53
4.1.1 Modelo de um Sistema de Comunicação com Espalhamento Espectral 56
4.1.2 Espalhamento Espectral por Seqüência Direta 57
4.1.3 Espalhamento Espectral por Saltos em Freqüência 58
4.2 Canal FFH-CDMA 59
4.2.1 Descrição do Sistema FFH-CDMA 59
4.2.2 Taxa de Corte do Canal FFH-CDMA Não-Quantizado 64
4.3 Decodificação q-ária Suave em Sistemas FFH-CDMA 66
4.3.1 Algoritmo de Chase q-ário 66
4.3.2 Resultados 67

5 Decodificação Turbo q-ária **73**
5.1 Introdução 73
5.2 Códigos Concatenados 75
5.2.1 Concatenação Serial 75
5.2.2 Concatenação Paralela 76
5.2.3 Entrelaçamento 76
5.3 Códigos Turbo 78
5.3.1 Códigos Produto 80
5.4 Decodificação Turbo 82
5.4.1 Logaritmo da Razão de Verossimilhança 82
5.4.2 Decodificação Iterativa 83
5.4.3 Confiabilidade da Decisão 85
5.4.4 Decodificação Turbo de Códigos Produto 88
5.5 Decodificação Turbo q-ária em Canais FFH-CDMA 91
5.5.1 Proposta para a Decodificação Turbo q-ária 91
5.5.2 Resultados 95

6 Conclusões **101**

Referências bibliográficas **104**

Lista de Figuras

2.1 Comparação de um sistema de comunicação com e sem codificação. . 6
2.2 Diagrama de blocos de um sistema de comunicação digital. 7
2.3 Código RS(7,3). 21

3.1 Diagrama em blocos de um sistema de comunicação digital que utiliza informação de medição de canal. 24
3.2 Representação geométrica da decodificação com informação de medição de canal. 28
3.3 Diagrama de fluxo para a decodificação suave de Chase. 31
3.4 Receptores de forma de onda. 36
3.5 Desempenho do código RS(15,9) utilizando decodificação algébrica de Berlekamp-Massey e decodificação suave de Chase-II. 38
3.6 Desempenho do código RS(31,25) utilizando decodificação algébrica de Berlekamp-Massey e decodificação suave de Chase-II. 39
3.7 Desempenho do código RS(31,19) utilizando decodificação algébrica de Berlekamp-Massey e decodificação suave de Chase-II. 40
3.8 Exemplo de erro em uma palavra do código RS(7,3) transmitida. . . . 42
3.9 Desempenho do código RS(15,9) utilizando decodificação suave de Chase-II e sua adaptação para símbolos q-ários. 49
3.10 Desempenho do código RS(31,25) utilizando decodificação suave de Chase-II e sua adaptação para símbolos q-ários. 50
3.11 Desempenho do código RS(31,19) utilizando decodificação suave de Chase-II e sua adaptação para símbolos q-ários. 51

4.1 Diagrama em blocos de um sistema com espalhamento espectral. . . . 57
4.2 Espalhamento espectral por seqüência direta. 57
4.3 Modelo do transmissor para espalhamento espectral por salto em freqüência. 59
4.4 Sistema de comunicação FFH-CDMA codificado. 60
4.5 nl-ésimo Detector de Energia. 62
4.6 Taxa de Corte do Canal R_0^{MV} versus número de usuários U para dois sistemas FFH-CDMA com taxas de codificação diferentes. 65
4.7 Desempenho do código RS(15,9) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$. 68
4.8 Desempenho do código RS(15,9) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$. 69
4.9 Desempenho do código RS(28,20) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$. 70
4.10 Desempenho do código RS(28,20) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$. 71

5.1 Esquema geral da concatenação serial. 75
5.2 Esquema geral da concatenação paralela. 76
5.3 Representação matemática do entrelaçador. 77
5.4 Ilustração da capacidade de aumento de peso do entrelaçador. 78
5.5 Construção do código $P = C_1 \otimes C_2$. 81
5.6 Decodificação SISO pata códigos sistemáticos. 84
5.7 Código produto bidimensional. 85
5.8 Diagrama esquemático do decodificador turbo de bloco. 89
5.9 Sistema de comunicação FFH-CDMA com decodificação turbo q-ária. 92
5.10 Decodificador turbo q-ário para o caso $Y_{ij}^{(t)} = D_{ij}^{(t)}$. 93
5.11 Decodificador turbo q-ário para o caso $Y_{ij}^{(t)} \neq D_{ij}^{(t)}$. 94
5.12 Projeções cartesianas para a decodificação turbo a) binária e b) q-ária. 95
5.13 Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando o número de usuários simultâneos, com $M = 16$, $L = 9$ e $P_a = (225, 81, 49)$. 96

5.14 Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando a eficiência espectral, com $M = 16$, $L = 9$ e $P_a = (225, 81, 49)$. 97
5.15 Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando o número de usuários simultâneos, com $M = 32$, $L = 6$ e $P_b = (784, 400, 81)$. 98
5.16 Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando a eficiência espectral, com $M = 32$, $L = 6$ e $P_b = (784, 400, 81)$. 99

Lista de Tabelas

5.1 Efeito das permutações nas seqüências do codificador da figura 5.4. . 78
5.2 Tabela comparativa das formas de concatenação de códigos de bloco. 80

Capítulo 1

Introdução

1.1 Comunicações Digitais

Na atual época, onde temos a chamada *sociedade da informação*, o armazenamento, a transmissão e o processamento de mensagens é uma necessidade evidente. Desta maneira, é bastante natural o interesse de um grande número de pessoas pelo conceito de informação, bem como pelo conhecimento das técnicas necessárias para sua aplicação.

Tendo em vista o enorme crescimento da demanda pela utilização da informação, cada vez mais os fabricantes põem no mercado soluções que não demoram para se tornarem obsoletas devido às novas necessidades que surgem. Os recursos computacionais são os grandes impulsionadores deste crescimento, que, paralelamente, requerem que a capacidade de transmissão dos sistemas seja constantemente aumentada. Porém, não apenas a capacidade dos sistemas de comunicações digitais deve ser considerada, a confiabilidade da transmissão da informação também é um fator importante para os usuários.

Em projetos de sistemas de comunicações, os parâmetros que limitam a capacidade de transmissão e a confiabilidade da informação recebida são: a largura de faixa espectral e a potência do sinal transmitido. A limitação da largura de faixa espectral ocorre principalmente devido às restrições impostas por órgãos reguladores que têm a função de evitar/controlar as interferências entre os diversos sistemas

existentes. A limitação da potência do sinal se deve por questões econômicas, regulamentadoras e tecnológicas. Com isso, os sistemas de comunicações são projetados para trabalharem com valores limítrofes da relação sinal-ruído.

Uma solução utilizada para projetar sistemas de comunicações que trabalhem cada vez mais próximos de seus limites de relação sinal-ruído é a codificação de canal. Porém, a aplicação de técnicas de codificação de canal refletem em uma maior complexidade computacional no sistema, principalmente devido aos algoritmos de decodificação. Desta maneira, existe um grande esforço da comunidade científica em pesquisas relacionadas com as técnicas de codificação de canal, que buscam soluções ótimas que satisfaçam o compromisso entre eficiência e complexidade para cada aplicação específica.

1.2 Contexto e objetivos do trabalho

Na década de 90, Berrou, Glavieux e Thitimajshima propuseram um novo esquema de codificação e decodificação de canal, chamado de codificação turbo, com a intenção de obter melhores resultados comparados àqueles obtidos com as técnicas clássicas de codificação [1]. Ao longo de quase dez anos, a comunidade científica mostrou que o uso desta técnica trouxe resultados significativos. Pyndiah estendeu o conceito de decodificação para códigos produto, mas considerou a aplicação apenas em canais gaussianos [2]. D. C. Cunha estudou a aplicação da decodificação turbo de códigos produtos em canais não-gaussianos, especialmente em canais CDMA com saltos rápidos em freqüência [3].

O estudo realizado em [3] levou em consideração os códigos binários. Estes códigos, quando utilizados para a composição de códigos produtos, apesar de apresentarem uma capacidade de correção boa, não demonstraram um boa eficiência espectral para o sistema FFH-CDMA. Também foi levantada a hipótese das conversões bit-símbolos utilizadas degradarem o desempenho do sistema.

Diante destas evidências, pensamos na utilização de códigos q-ários, com elementos em $GF(q > 2)$, para aplicação em canais FFH-CDMA. Portanto, o objetivo deste presente trabalho é estudar o comportamento dos códigos produto q-ários, conjuntamente com a decodificação turbo q-ária, em sistemas FFH-CDMA. Espera-

se que, com esta aplicação, o sistema apresente uma melhoria tanto na capacidade de correção quanto na eficiência espectral.

1.3 Organização da tese

No capítulo 2, é apresentada uma fundamentação teórica abordando os conceitos básicos sobre teoria de informação e codificação, mais especificamente, códigos de bloco lineares e algoritmos de decodificação que utilizam técnicas algébricas e informação do canal. Estes conceitos são fundamentais para o entendimento do objetivo principal deste trabalho.

Em seguida, no capítulo 3 abordamos a decodificação suave q-ária, onde, inicialmente, é apresentado os algoritmos de Chase original, ou seja, para símbolos binários. Neste capítulo é também descrita uma proposta com as adaptações necessárias para realizarmos a decodificação suave de códigos q-ários, baseando-se nos algoritmos de Chase. Uma comparação de desempenho será realizada com o método original de Chase em um canal AWGN.

No capítulo 4, é apresentada a aplicação da decodificação suave q-ária em sistemas FFH-CDMA. Iniciamos com as técnicas de espalhamento espectral, para em seguida realizarmos uma descrição detalhada do canal CDMA com saltos rápidos em freqüência. As adequações da proposta de decodificação suave q-ária ao canal FFH-CDMA também são descritas.

No capítulo 5, são apresentadas as contribuições relacionadas à decodificação turbo q-ária em sistemas FFH-CDMA. Os conceitos de códigos produto e decodificação iterativa são descritos de acordo com as técnicas já conhecidas para símbolos binários. Após esta introdução, são apresentados os procedimentos propostos para a implantação da decodificação turbo q-ária em canais FFH-CDMA. Por fim, os resultados são comentados.

Finalizando o trabalho, no capítulo 6 são apresentadas as conclusões do estudo realizado e as perspectivas de trabalhos futuros.

1.4 Resumo das principais contribuições

As principais contribuições desta dissertação foram:

- A generalização dos algoritmos de decodificação suave de Chase, utilizados em códigos binários, a fim de suportar símbolos q-ários com elementos pertencentes a $GF(q > 2)$. Para alcançar este resultado, foram realizadas adaptações no cálculo do Logaritmo da Razão de Verossimilhança e na geração dos padrões de testes, onde foi levado em consideração o compromisso entre complexidade e capacidade de correção.

- A criação de um método capaz de realizar a decodificação turbo de códigos q-ários em canais FFH-CDMA. Para isso, foram realizadas manipulações no cálculo da confiabilidade da decisão do decodificador e nos parâmetros adicionais de entrada do decodificador turbo. Além disso, foi desenvolvido um novo procedimento iterativo para o decodificador turbo, definindo os critérios de sua realimentação.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão apresentados os conceitos básicos sobre sistemas de comunicações digitais, especialmente sobre a teoria de codificação que é o principal pré-requisito para o entendimento dos capítulos seguintes, onde serão apresentados os objetos de estudo desta dissertação.

2.1 Introdução

A aplicação da codificação de canal, ou seja, de códigos corretores de erros, proporciona um grande benefício em sistemas de comunicações: o incremento da confiabilidade da informação recebida. Esta vantagem pode ser claramente observada na figura 2.1, onde os desempenhos de um sistema codificado e não-codificado são comparados. O desempenho é avaliado através da curva $BER \times E_b/N_o$, onde BER (do inglês, *Bit Error Rate*) é a taxa de erro de bit, que representa a probabilidade de erro de bit quando a fonte de informação é conhecida, e E_b/N_o, a relação sinal-ruído normalizada, onde E_b é a energia por bit e N_0 é a densidade espectral do ruído. Observe que o desempenho de um sistema não-codificado pode ser alcançado por um sistema codificado com uma necessidade menor de potência transmitida. Porém, a codificação contribui com o aumento da complexidade do sistema, principalmente na decodificação, o que evidencia uma desvantagem para esta técnica.

O projeto de um sistema digital de comunicação se concentra em torno da transmissão da informação de uma fonte a um destinatário de uma forma rápida e con-

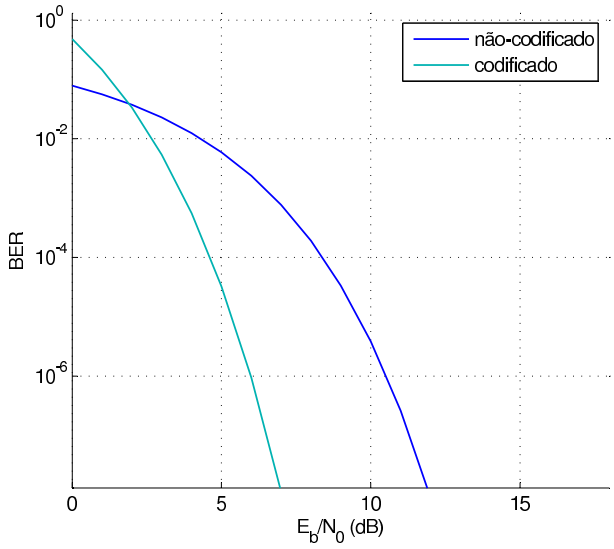

Fig. 2.1: Comparação de um sistema de comunicação com e sem codificação.

fiável, segundo as necessidades do usuário. Para isso, parâmetros como a largura de faixa, a potência do sinal e a complexidade de implementação precisam ser levados em conta. A largura de faixa é um fator normalmente limitado pelo meio de transmissão, enquanto a potência do sinal e a complexidade de implementação são características do sistema que representam fatores de custo ao projetista. Desta maneira, surge a grande importância de sabermos balancear estes parâmetros de modo a chegarmos a uma relação custo-benefício ideal.

2.2 Elementos de um Sistema de Comunicação Digital

Os elementos básicos de um sistema de comunicação digital são ilustrados no diagrama de blocos da figura 2.2. A seguir, apresentaremos um descrição de cada elemento do diagrama.

2.2.1 Fonte de Informação

A fonte de informação representa as informações a serem transmitidas, podendo ser analógica ou discreta. Consideraremos apenas as fontes discretas, que podem

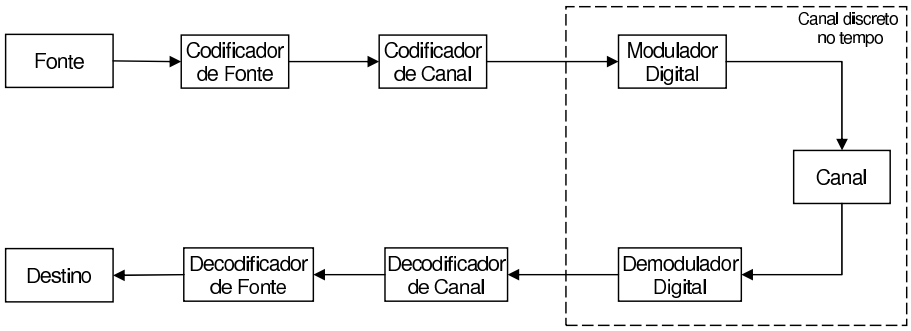

Fig. 2.2: Diagrama de blocos de um sistema de comunicação digital.

ser caracterizadas por um alfabeto da fonte e uma distribuição de probabilidade de ocorrência de cada símbolo do alfabeto. Então, é possível determinar um modelo probabilístico para a fonte e a taxa de informação da mesma é medida em bits por segundo. Em seguida, a seqüência de símbolos será processada pelo codificador de fonte, cuja finalidade é reduzir a redundância da fonte, diminuindo a largura de faixa necessária para a transmissão daquela informação. No receptor, o decodificador de fonte tem a função de recuperar a seqüência de símbolos que será passada ao destinatário.

2.2.2 Codificador de Canal

Uma vez que a redundância foi extraída pelo codificador de fonte, a seqüência de bits em sua saída pode sofrer erros durante a transmissão da informação. A função do codificador de canal é introduzir uma redundância controlada na seqüência de informação de forma a tornar a transmissão confiável. No receptor, o decodificador de canal tem a função de tentar recuperar a seqüência original dos bits de informação.

2.2.3 Modulador Digital

A função do modulador é adequar a saída do codificador de canal ao canal de transmissão. Ele converte a seqüência de entrada em uma seqüência de sinais apropriados à transmissão pelo meio físico, que é analógico, de acordo com o diagrama de constelação da modulação. Por exemplo, para a modulação binária, ele simplesmente converte o dígito binário 0 ou 1 em um sinal $s_0(t)$ ou $s_1(t)$, respectivamente,

de duração T_s segundos cada. Para a modulação M-ária, os M símbolos possíveis são convertidos em um conjunto de M sinais $s_0(t), s_1(t), ..., s_{M-1}(t)$. No receptor, o demodulador digital tem a função inversa do modulador, isto é, a partir do sinal analógico recebido, gerar o símbolo do diagrama de constelação correspondente àquele sinal.

2.2.4 Canal de Transmissão

O termo canal de transmissão inclui todas as operações necessárias para preparar o sinal modulado para a transmissão no canal físico, o meio de transmissão propriamente dito e as operações de recepção que recuperam o sinal no ponto anterior à demodulação.

É exatamente no canal de transmissão que ruído e interferências, que são os principais fatores limitante de desempenho dos sistemas de comunicações, atuam no sinal transmitido, prejudicando a boa recepção da informação por parte do destinatário. Entre as diversas formas de ruído que podemos encontrar, temos o ruído térmico, que possui uma distribuição de probabilidade gaussiana e sempre está presente nas transmissões.

2.3 Modelos de Canais

O canal de transmissão de um sistema de comunicação pode ser descrito por um modelo, que é uma representação matemática definida para descrever como os sinais são processados e afetados pelo ambiente de comunicação real. A seguir, vamos mencionar alguns modelos clássicos de canais utilizados em sistemas de comunicações.

2.3.1 Canal Discreto no Tempo

O canal discreto no tempo é definido pelo conjunto de entradas do modulador, o conjunto de saídas do demodulador e as estatísticas que relacionam as possíveis saídas com cada possível entrada. Destacamos um modelo de canal chamado canal discreto sem memória (do inglês, *Discrete Memoryless Channel*), que é definido

por um conjunto de símbolos M-ários de entrada $\{x_i\}$, um conjunto de símbolos q-ários de saída $\{y_j\}$ e um conjunto de probabilidades condicionais $p(y_j|x_i)$, chamadas probabilidades de transição, onde $i = 0, 1, ..., (M-1)$ e $j = 0, 1, ..., (q-1)$. A denominação de canal sem memória se refere ao fato de que o símbolo de saída a qualquer instante depende estatisticamente apenas do símbolo de entrada naquele mesmo instante.

2.3.2 Canal AWGN

O canal AWGN (do inglês, *Additive White Gaussian Noise*) pode ser definido como um tipo de canal discreto sem memória, caracterizado por sua saída ser a entrada adicionada de um ruído gaussiano. Ele é descrito em termos da entrada x e da saída y da seguinte forma:

$$y = x + n_0, \tag{2.1}$$

onde n_0 é uma variável aleatória gaussiana de média *zero*, variância σ^2 e a entrada x pode assumir um de M valores discretos, tal que $M \geq 2$, sendo M os símbolos do modulador. A função densidade de probabilidade condicional da saída y, dada a entrada x_i, é expressa por:

$$p(y|x = x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(y-x_i)^2}{2\sigma^2}}. \tag{2.2}$$

2.3.3 Outros Tipos de Canais

Existem diversos modelos matemáticos que representam canais não-gaussianos. Um tipo de degradação de sinal bastante conhecido em comunicações móveis são as flutuações que reduzem o nível do sinal recebido, abaixo do seu valor mediano, a uma determinada distância do transmissor, chamada de desvanecimento [4]. Este fenômeno é causado pelo multipercurso do sinal transmitido, ou seja, o receptor recebe ondas secundárias provenientes da reflexão e/ou difração do sinal transmitido em obstáculos, tais como: prédios, árvores, montanhas, etc. O desvanecimento ainda se divide em duas categorias: a) o desvanecimento lento ou de larga escala,

em que a flutuação do nível do sinal ocorre de uma maneira suave comparada ao comprimento de onda; e b) o desvanecimento rápido ou de pequena escala, onde a flutuação do nível do sinal ocorre em distâncias em torno de meio comprimento de onda, geralmente ocasionadas por curtos deslocamentos do receptor. O desvanecimento suave pode ser representado pela distribuição de probabilidade log-normal. O desvanecimento rápido pode ser modelado pela equação de distribuição de probabilidade de Rice quando o sinal da linha de visada direta também é captado pelo receptor, ou pela distribuição de probabilidade de Rayleigh quando apenas os sinais refletidos e/ou difratados são captados pelo receptor.

2.4 Capacidade de Canal

Nesta seção, vamos apresentar alguns conceitos básicos de teoria de informação [5], visando destacar o que vem a ser a capacidade de canal como limite ideal a se atingir no desenvolvimento de um sistema de comunicações, concluindo com o conceito de um limitante mais prático e atingível, conhecido por taxa de corte de canal.

2.4.1 Entropia e Informação Mútua

No início dos estudos sobre teoria da informação, buscava-se uma caracterização matemática da quantidade de informação que poderia ser transmitida de maneira confiável sobre um dado canal físico. Para solucionar este problema, era necessário estabelecer um medida matemática da informação, que encontrou uma solução natural na função logarítmica. Desta forma, o conteúdo de informação de uma mensagem gerada a partir de um conjunto de M mensagens igualmente prováveis é $\log M$, em que a base do logaritmo depende da unidade básica de informação, que usualmente é considerada o bit.

Uma vez definido como medir a quantidade de informação transmitida, é possível determinar a taxa da fonte de informação como sendo $\log_2 M$ bits por símbolo, em que esta fonte produza M símbolos equiprováveis e cada um deles seja independente de um intervalo de símbolo para outro. Porém, não podemos atribuir este conceito

de medida de informação para as fontes cujos símbolos não são equiprováveis.

Para estabelecer uma medida de informação para esta classe de fontes, Shannon utilizou o conceito de entropia de uma fonte de informação. Ela fornece uma medida apropriada para a incerteza a priori de qualquer símbolo produzido por uma fonte discreta e é dada por:

$$H = -\sum_{i=1}^{M} p_i \log_2 p_i. \tag{2.3}$$

Observe que a equação (2.3) também enquadra as fontes equiprováveis, pois neste caso, $H = -\log p_i = \log M$, conforme descrito anteriormente. Sendo assim, a função entropia fornece uma medida da quantidade média de informação produzida por símbolo da fonte.

Conhecido o conceito de entropia, este pode ser estendido a alfabetos conjuntos X e Y, caracterizando a definição de entropia conjunta:

$$H(X,Y) = -\sum_{i=1}^{M}\sum_{j=1}^{q} p(x_i, y_j) \log_2 p(x_i, y_j). \tag{2.4}$$

Lembrando que $[-\log_2 p(x_i)]$ mede a incerteza a priori associada à transmissão do símbolo x_i e $[-\log_2 p(x_i|y_j)]$ mede a incerteza final sobre a transmissão do símbolo de entrada x_i após se ter recebido o símbolo y_j, o ganho de informação sobre o símbolo de entrada x_i após a observação do símbolo de saída y_j é dado por:

$$I(x_i; y_j) = \log_2 p(x_i|y_j) - \log_2 p(x_i) = \log_2 \left(\frac{p(x_i|y_j)}{p(x_i)} \right). \tag{2.5}$$

Utilizando a regra de Bayes e manipulando as equações (2.4) e (2.5), temos:

$$I(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{q} I(x_i; y_j) p(x_i, y_j). \tag{2.6}$$

A equação (2.5) define a informação mútua entre os acontecimentos $X = x_i$ e $Y = y_j$. A informação mútua média entre os alfabetos X e Y é dada por:

$$I(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{q} p(x_i,y_j)\log_2\left(\frac{p(x_i,y_j)}{p(x_i)p(y_j)}\right). \tag{2.7}$$

Para interpretarmos o conceito de informação mútua média, vamos definir primeiramente a entropia condicional como:

$$H(X|Y) = -\sum_{i=1}^{M}\sum_{j=1}^{q} p(x_i,y_j)\log_2 p(x_i|y_j). \tag{2.8}$$

Reescrevendo (2.6):

$$\begin{aligned} I(X,Y) &= \sum_{i=1}^{M}\sum_{j=1}^{q} p(x_i,y_j)[\log_2 p(x_i|y_j) - \log_2 p(x_i)] \\ &= \sum_{i=1}^{M}\sum_{j=1}^{q} p(x_i,y_j)\log_2 p(x_i|y_j) - \sum_{i=1}^{M}\log_2 p(x_i)\sum_{j=1}^{q} p(x_i,y_j) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{q} p(x_i,y_j)\log_2 p(x_i|y_j) - \sum_{i=1}^{M}[\log_2 p(x_i)]p(x_i). \end{aligned} \tag{2.9}$$

Comparando com (2.8), verificamos que a primeira parcela do lado direito de (2.9) é exatamente $-H(X;Y)$ enquanto que a segunda é a entropia do alfabeto de entrada. Então, temos que:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X). \tag{2.10}$$

A relação anterior pode ser escrita de outra forma, baseada na definição de entropia conjunta. Manipulando a equação (2.4) chegamos à:

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{2.11}$$

2.4.2 Teorema da Capacidade de Canal

O teorema da capacidade de canal, ou $3^{\underline{o}}$ teorema de Shannon, é um dos resultados mais importantes da teoria da informação. Seja um processo estacionário $X(t)$, de média zero e limitado a uma faixa de B Hz, vamos denominar X_k, $k = 1, 2, ..., n$, as V.A.s contínuas obtidas a partir da amostragem uniforme de $X(t)$ a uma taxa de $2B$ amostras por segundo. Estas amostras são transmitidas em T segundos sobre um canal ruidoso, também limitado a B Hz. Desta forma, o número de amostras n é dado por:

$$n = 2BT. \tag{2.12}$$

X_k representa uma amostra do sinal transmitido, enquanto a saída do canal é perturbada por ruído AWGN de média zero e densidade espectral de potência $N_0/2$. A V.A. contínua Y_k, $k = 1, 2, ..., n$, representa as amostras do sinal recebido, conforme:

$$Y_k = X_k + N_k, \qquad k = 1, 2, ..., n \tag{2.13}$$

e são consideradas estatisticamente independentes.

O Teorema da Capacidade de Canal é estabelecido da seguinte maneira:

"A capacidade de um canal de largura de faixa B Hz, perturbado por ruído AWGN de densidade espectral de potência $N_0/2$ e limitado em B Hz é dada por:

$$C = B \log_2 \left(1 + \frac{P}{N_0 B}\right), \quad (bits/s) \tag{2.14}$$

onde P é a potência média transmitida".

Este teorema relata outro importante resultado da teoria da informação ao relacionar três parâmetros chave do sistema: largura de faixa do canal, potência média do sinal transmitido e densidade espectral do ruído na saída do canal.

Vamos agora analisar o caso do sistema ideal para entendermos mais uma importante contribuição de Shannon às comunicações. Sistema ideal é definido como aquele que transmite à taxa máxima, que é a capacidade de canal C. Podemos expressar a potência média transmitida como:

$$P = E_b C, \tag{2.15}$$

onde E_b é a energia transmitida por bit. Substituindo (2.15) em (2.14), temos:

$$\frac{C}{B} = \log_2 \left(1 + \frac{E_b}{N_0}\frac{C}{B}\right). \tag{2.16}$$

Isolando o termo E_b/N_0, temos que:

$$\frac{E_b}{N_0} = \frac{2^{\frac{C}{B}} - 1}{\frac{C}{B}}. \tag{2.17}$$

Analisando (2.17) para uma largura de faixa infinita, encontramos um valor limite para E_b/N_0:

$$\left(\frac{E_b}{N_0}\right)_\infty = \lim_{B\to\infty}\left(\frac{E_b}{N_0}\right) = \ln 2 = 0,693 = -1,6dB. \tag{2.18}$$

Este valor é conhecido como limite fundamental de Shannon. O limite de Shannon pode ser definido ainda como o valor mínimo de E_b/N_0 necessário para obtermos um sistema ideal livre de erros de transmissão.

2.4.3 Taxa de corte

Baseado na existência de limitações práticas para se atingir a capacidade de canal, foi definida uma medida, chamada de taxa de corte de canal, na tentativa de orientar os projetistas de sistemas de comunicações. A taxa de corte R_0 é um parâmetro definido para qualquer canal DMC, cujo valor é sempre inferior ao de C. Ela fornece um limite prático para a taxa com que a informação pode ser transmitida confiavelmente através do canal. Matematicamente, a taxa de corte para um canal DMC, de G entradas e Q saídas é dada por [17]:

$$R_0 = -\log_2 \left\{ \min_{P(x_i)} \sum_{j=0}^{Q-1} \left[\sum_{i=0}^{G-1} \sqrt{p(y_j|x_i)} P(x_i) \right]^2 \right\}, \tag{2.19}$$

onde $p(y_j|x_i)$ representa as probabilidades de transição do canal e a minimização é

tomada sobre todas as possíveis distribuições de probabilidades a priori $P(x_i)$.

Para o caso do canal AWGN, temos que as entradas do canal são $x_0 = -a$ e $x_1 = +a$, com probabilidades iguais a $P(x_0) = P(x_1) = \frac{1}{2}$. A taxa de corte R_0 é escrita como:

$$R_0 = -\log_2 \int_{-\infty}^{+\infty} \left(\frac{1}{2}\sqrt{p(y|x_0)} + \frac{1}{2}\sqrt{p(y|x_1)} \right)^2 dy, \tag{2.20}$$

em que as probabilidades de transição $p(y|x_0)$ e $p(y|x_1)$ são dadas por:

$$p(y|x_0) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{(y+a)^2}{2\sigma^2}}$$

e

$$p(y|x_1) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{(y-a)^2}{2\sigma^2}}.$$

Desta maneira, o valor mínimo de E_b/N_0 para atingir R_0 é:

$$\frac{E_b}{N_0} = 2\ln 2 = 1,4dB. \tag{2.21}$$

Comparando este valor com o resultado anterior, vemos que a taxa de corte está exatamente $3dB$ acima do limite de Shannon, que vale $-1,6dB$.

2.5 Códigos de Bloco Lineares

Os códigos de bloco constituem uma das principais técnicas clássicas de codificação e controle de erros em sistemas de comunicações. Abordaremos os códigos de bloco lineares q-ários, pois estes serão utilizados em diversas aplicações deste trabalho. O objetivo desta seção é oferecer uma descrição matemática geral dos códigos de bloco $C(n, k)$.

Os códigos de bloco lineares podem ser definidos como o conjunto de todos os vetores de n símbolos formados pela combinação linear, sobre o $GF(q)$, de k vetores de base linearmente independentes $\mathbf{g_1}, \mathbf{g_2}, ..., \mathbf{g_k}$. Um vetor $\mathbf{v}$ é considerado uma palavra-código em C se e somente se ele se encontrar no espaço de vetores k-

dimensional formado por $\{\mathbf{g_i}\}$. Alternativamente, se $\{\mathbf{g_i}\}$ for arranjado como linhas de uma matriz geradora $\mathbf{G}_{k\times n}$, uma palavra-código $\mathbf{v}$ pode ser expressa como:

$$\mathbf{v} = (u_1, u_2, ..., u_k) \begin{bmatrix} g_1 \\ g_2 \\ . \\ . \\ . \\ g_k \end{bmatrix} = \mathbf{uG}, \tag{2.22}$$

onde $\mathbf{u}$ é um vetor de k símbolos de informação e $\mathbf{G}$ é a chamada matriz geradora do código.

Outra abordagem pode ser feita em relação aos códigos lineares $C(n,k)$ ao considerá-los como um subgrupo de ordem q^k extraído de todos os q^n vetores possíveis. Dentro desse contexto, os códigos de bloco lineares podem ser considerados códigos de grupo, herdando certas propriedades, por exemplo: a) a soma de duas palavras-código também é uma palavra-código; e b) cada código linear contém o vetor nulo.

A matriz geradora do código $C(n,k)$ pode ser representada da seguinte forma:

$$\mathbf{G} = [\mathbf{I_k}; \mathbf{P}], \tag{2.23}$$

onde $\mathbf{I_k}$ é a matriz identidade $k \times k$ e $\mathbf{P}$ é a matriz de paridade $k \times (n-k)$ cujos elementos determinam os símbolos de paridade que são relacionados com os símbolos de informação. Os códigos cuja matriz geradora é representada segundo (2.23) são chamados códigos sistemáticos.

Os códigos de bloco não-sistemáticos são equivalentes aos sistemáticos, dado que permutações sobre colunas e operações elementares sobre as linhas da matriz possibilitam a conversão de uma forma para outra. A preferência pelos códigos sistemáticos é percebida na facilidade das operações de codificação e decodificação.

2.5.1 Matriz de Verificação de Paridade

Vimos que a matriz geradora constitui uma das formas de se relacionar os símbolos de informação aos símbolos de paridade. Também podemos realizar esta relação através da matriz $\mathbf{H}_{(\mathbf{n-k})\times\mathbf{n}}$, dada por:

$$\mathbf{H} = [\mathbf{P^T}; \mathbf{I_{n-k}}], \tag{2.24}$$

onde $\mathbf{P^T}$ é a transposta da matriz de paridade $\mathbf{P}$ e $\mathbf{I_{n-k}}$ é a matriz identidade de dimensão $n-k \times n-k$. A matriz $\mathbf{H}$ é chamada matriz de verificação de paridade e toda palavra-código $\mathbf{v}$ deve satisfazer à seguinte equação:

$$\mathbf{vH^T} = 0. \tag{2.25}$$

A relação entre a matriz geradora e a matriz de verificação de paridade de um código de bloco linear deve obedecer à seguinte equação:

$$\mathbf{GH^T} = 0. \tag{2.26}$$

Esta relação pode ser interpretada como uma necessidade que cada linha de $\mathbf{G}$ tem que satisfazer todas as equações de verificação de paridade, por ser uma palavra-código.

2.5.2 Distância Mínima

Uma importante característica dos códigos de bloco é a sua capacidade de detecção e correção de erros. Entretanto, faz-se necessário definir alguns conceitos que permitirão a abordagem deste parâmetro.

A distância de Hamming entre dois vetores é definida como o número de posições onde os elementos de cada vetor diferem.

O peso de Hamming de um vetor é o número de elementos não-nulos do vetor. Podemos defini-lo também como a distância de Hamming entre o vetor considerado e o vetor nulo.

A distância mínima de um código é a menor distância de Hamming entre todas

as combinações possíveis de pares de palavras do código. A distância mínima de um código C e a estrutura da matriz $\mathbf{H}$ podem ser relacionadas, bastando para isso, que esta última seja escrita da seguinte maneira:

$$\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n}], \tag{2.27}$$

onde $\mathbf{h_i}$ é a i-ésima coluna de $\mathbf{H}$. Observe que a distância mínima do código é numericamente igual à quantidade mínima de colunas de $\mathbf{H}$ linearmente dependentes [6].

2.5.3 Códigos Cíclicos

Os códigos cíclicos formam uma subclasse dos códigos de bloco lineares. Na verdade, muitos dos principais códigos de bloco lineares descobertos até hoje são também códigos cíclicos ou estritamente relacionados com os códigos cíclicos. Uma vantagem dos códigos cíclicos sobre muitos outros tipos de códigos é que eles são fáceis de implementar. Além disso, os códigos cíclicos possuem uma estrutura matemática bem definida, o que resultou no desenvolvimento de esquemas de decodificação bastante eficientes.

Um código é considerado cíclico quando ele atende a duas propriedades fundamentais [6]:

1. Linearidade: A soma de duas palavras-código também é uma palavra-código.

2. Cíclica: Qualquer deslocamento cíclico de uma palavra-código também é uma palavra-código.

A propriedade 1 enfatiza que um código cíclico é um código de bloco linear. Para representar a propriedade 2 em termos matemáticos, considere a seqüência $(x_0, x_1, ..., x_{n-1})$ uma palavra-código de um código de bloco linear de parâmetros (n, k). O código será cíclico se as seqüências:

$$(x_{n-1}, x_0, ..., x_{n-2}),$$
$$(x_{n-2}, x_{n-1}, ..., x_{n-3}),$$
$$.$$
$$.$$
$$.$$
$$(x_1, x_2, ..., x_0)$$

são todas palavras códigos.

A seguir serão apresentados alguns códigos cíclicos conhecidos.

Códigos de Bose-Chaudhuri-Hocquenqhem (BCH)

Uma das classes de códigos de bloco lineares mais importantes são os códigos BCH, que são códigos cíclicos com uma ampla variedade de parâmetros. Os códigos BCH mais comuns são caracterizados conforme a seguir. Para qualquer inteiro positivo m, maior ou igual a 3, e t, menor que $(2^m - 1)/2$, existe um código BCH binário com os seguintes parâmetros:

Tamanho da palavra-código:	$n = 2^m - 1$
Número de bits de informação:	$k \geq n - mt$
Distância mínima:	$d_{min} \geq 2t + 1$

Os códigos BCH possuem a capacidade de detectar e corrigir até t erros por palavra-código. Os códigos BCH oferecem flexibilidade na escolha dos parâmetros do código, ou seja, no tamanho da palavra-código e na taxa do código. Além disso, os códigos BCH com tamanho de palavra-código menor que algumas centenas estão entre os melhores códigos conhecidos, considerando a mesma taxa de código e tamanho de palavra-código.

Códigos de Reed-Solomon (RS)

Os códigos de Reed-Solomon (RS) [7] são uma importante subclasse dos códigos BCH não-binários. O codificador para um código RS difere de um codificador binário pelo fato de operar com múltiplos bits, ao invés de bits individualmente.

Especificamente, o codificador para um código $RS(n, k)$, com símbolos de m bits, agrupa a sequência binária de entrada em blocos, com tamanho igual a km bits cada. Cada bloco é tratado como k símbolos, com cada símbolo contendo m bits. O algoritmo de codificação expande o bloco de k símbolos para n símbolos, adicionando $n - k$ símbolos redundantes. Quando m é uma potência de dois, os símbolos de m bits são chamados de *bytes*. Um valor comum para m é 8.

Um código RS capaz de corrigir até t erros possui os seguintes parâmetros:

Tamanho da palavra-código:	$n = 2^m - 1$
Símbolos de informação:	k
Símbolos de paridade:	$n - k = 2t$
Distância mínima:	$d_{min} = 2t + 1$

O tamanho da palavra-código de um código RS é igual ao alfabeto do código subtraído de uma unidade, e a distância mínima é igual ao número de símbolos de paridade adicionado de uma unidade. É fácil mostrar que nenhum código de bloco linear com parâmetros (n, k) pode obter uma distância mínima maior que $n - k + 1$ [8]. Um código de bloco linear (n, k) que possui distância mínima igual a $n - k + 1$ é chamado de código separável pela distância máxima. Portanto, todo código RS é um código separável pela distância máxima. Desta maneira, os códigos RS utilizam eficientemente a redundância, permitindo que o tamanho da palavra-código e do alfabeto do código possam ser ajustados para suportar diversos tamanhos de seqüência de informação k. Além disso, os códigos RS fornecem uma variedade de taxas de código que podem ser escolhidas para otimizar o desempenho. Finalmente, técnicas de decodificação eficientes estão disponíveis para serem usadas com códigos RS, sendo mais um motivo de sua ampla aplicação.

2.5.4 Decodificação de Códigos de Bloco

A distância mínima d_{min} de um código de bloco linear é o parâmetro fundamental que determina a capacidade de detectar e corrigir erros. Um código é capaz de corrigir até t erros em qualquer palavra recebida, se a distância mínima d_{min} entre as palavras for, pelo menos, $2t + 1$. Na figura 2.3 é ilustrada uma representação

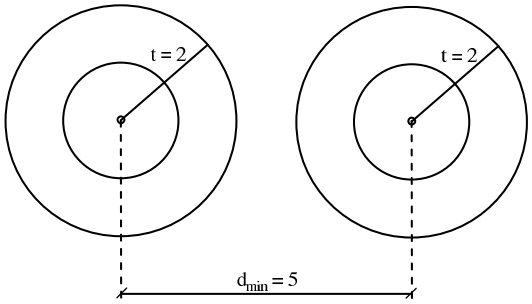

Fig. 2.3: Código RS(7,3).

geométrica de um código RS(7,3), ressaltando sua capacidade de correção. O centro das esferas da figura 2.3 representa uma das possíveis palavras do código, cuja capacidade de correção é indicada pelo raio da esfera.

Os exemplos mencionados servem para introduzirmos uma regra de decodificação que se caracteriza por corrigir todos os padrões de erro de peso menor ou igual a l, onde $l \leq t = \left\lfloor \frac{(d_{min}-1)}{2} \right\rfloor$. Esta regra é chamada decodificação BDD (do inglês, *Bounded Distance Decoding*). Se o código possuir distância mínima d_{min} ímpar, t assumirá $\frac{(d_{min}-1)}{2}$ e o código é dito corretor de t erros. Caso possua distância mínima par, t assumirá $\frac{(d_{min}-2)}{2}$ e o código é denominado corretor de t erros e detector de $(t+1)$ erros.

A maioria dos decodificadores BDD utiliza decodificação por síndrome, na qual a síndrome $\mathbf{s}$ é calculada a partir da palavra recebida $\mathbf{r}$ da seguinte forma:

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathbf{T}}. \tag{2.28}$$

Cada síndrome representa um padrão de erro específico de peso menor ou igual a l ou a ocorrência de algum padrão de erro indefinido, o que resulta na recepção de uma palavra que não está contida na esfera de raio l centrada em uma palavra-código. Assim, podemos descrever um código corretor de t erros como um código que fornece síndromes suficientes para identificar unicamente cada possível padrão de erro de peso menor ou igual a t.

Conforme mencionado, a regra de decodificação BDD necessita que as palavras recebidas estejam dentro da esfera de raio t centrada em uma possível palavra-código. Então, vemos que a decodificação BDD é um procedimento de decodificação

incompleta, pois existem palavras recebidas que não podem ser decodificadas por não estarem em nenhuma esfera relacionada a uma palavra-código, isto acontece para vários códigos de bloco.

Na tentativa de suprir esta lacuna, recorremos aos algoritmos de decodificação completa, onde cada palavra recebida é decodificada em uma das possíveis palavras-código transmitidas. Uma regra de decodificação completa, chamada decodificação pelo vizinho mais próximo (do inglês, *Nearest-Neighbor Decoding*), decodifica cada palavra recebida na palavra-código mais próxima, em termos de distância de Hamming. Acontece que, para muitos códigos, a decodificação pelo vizinho mais próximo não é uma estratégia prática, pois necessita de inúmeras comparações da palavra recebida com um conjunto de possíveis palavras-código, o que exige um grande recurso computacional.

Capítulo 3

Decodificação Suave q-ária

3.1 Introdução

No capítulo 2, foram apresentados métodos utilizados para a decodificação das palavras recebidas através de decisão abrupta (do inglês, *Hard Decision Decoding*). Embora apresente bons resultados em algumas aplicações, o desempenho deste tipo de decodificação é superado por outros algoritmos de decodificação que dispõem da informação do canal utilizado na transmissão, ao invés de empregar apenas as propriedades algébricas do código corretor de erros na decodificação. Este método é conhecido como decodificação por decisão suave (do inglês, *Soft Decision Decoding*).

Na figura 3.1, é apresentado um diagrama em blocos de um sistema de comunicação digital que utiliza um decodificador baseado em decisão suave. Uma seqüência de K bits (ou símbolos) de informação, denotado por $\mathbf{I} = I_1, I_2, ..., I_K$, é codificada em um bloco de N bits (ou símbolos) denotado por $\mathbf{X} = X_1, X_2, ..., X_N$. Estes bits (ou símbolos) codificados alimentam o modulador, que determina a forma de onda $x(t)$ que será transmitida pelo canal. Quando o decodificador é baseado em decisão abrupta, o demodulador fornece apenas uma seqüência de N bits (ou símbolos), $\mathbf{Y} = Y_1, Y_2, ..., Y_N$, que é determinada de acordo com a forma de onda $y(t)$. Porém, quando temos um decodificador baseado em decisão suave, o demodulador irá fornecer, além da seqüência $\mathbf{Y}$, uma seqüência de N números positivos denotados por $\mathbf{\Lambda} = \Lambda_1, \Lambda_2, ..., \Lambda_N$. Estes valores são conhecidos como informação de medição

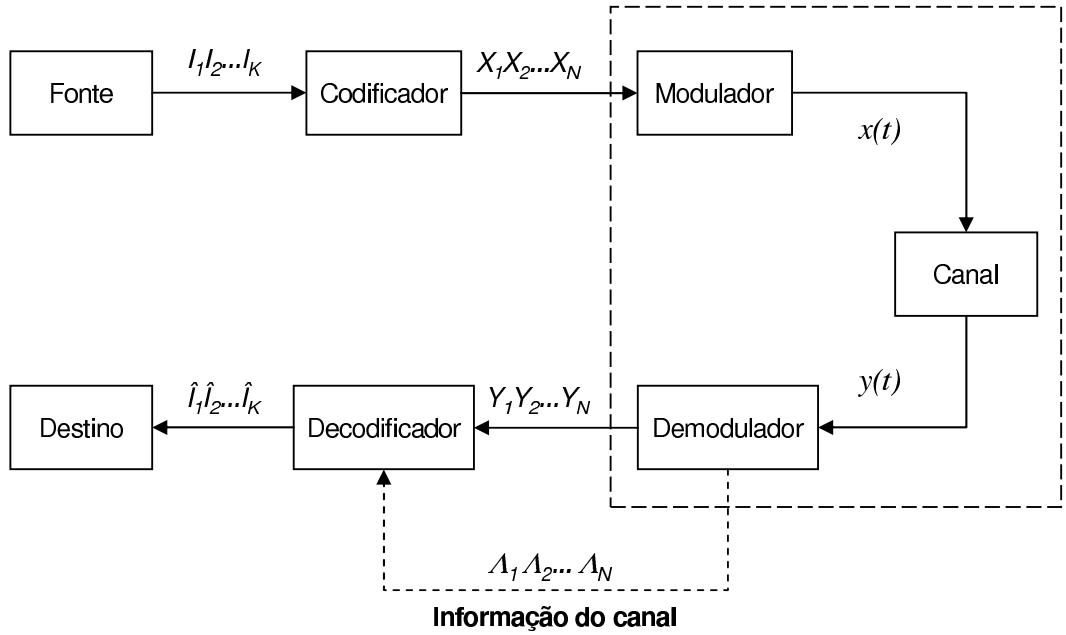

Fig. 3.1: Diagrama em blocos de um sistema de comunicação digital que utiliza informação de medição de canal.

de canal e são utilizados para indicar a confiabilidade relativa dos bits (ou símbolos) recebidos. Para $\Lambda_i > \Lambda_j$, temos que o bit (ou símbolo) da posição i da palavra recebida $\mathbf{Y}$, denotado por Y_i, tem uma maior probabilidade de ter sido recebido corretamente do que o bit (ou símbolo) Y_j.

Uma das primeiras aplicações que utilizou a informação de medição de canal para decodificar códigos de bloco binários foi o algoritmo de Wagner [9] e [10], onde a utilização da informação de medição de canal aumenta em 1(um) a capacidade de correção de um código cuja distância mínima é um número par. Métodos mais sofisticados de codificação que utilizam a informação de medição de canal apareceram posteriormente. Dois métodos bastante conhecidos são: o GMD [11] (do inglês, *Generalized Minimum Distance*), que pode ser aplicado em códigos de blocos binários e não-binários, e o método de Chase [12], conhecido pela sua flexibilidade na relação complexidade versus capacidade de correção.

No algoritmo de decodificação GMD, os símbolos da palavra recebida com menores confiabilidades, geradas a partir da informação de medição de canal, são apagados e a nova palavra é submetida a uma decodificação por apagamento e

erro. A quantidade de símbolos que serão apagados em cada palavra varia entre 0 e $(d_{min} - 1)$, em ordem crescente, ou até que a desigualdade $\mathbf{Y} \cdot \mathbf{c}_r > n - d_{min}$ seja satisfeita, onde $\mathbf{Y}$ é a palavra recebida (decisão abrupta) na entrada do decodificador e $\mathbf{c}_r$ é a palavra de saída do decodificador de apagamento e erro, de tamanho n e distância mínima d_{min}. Sendo que:

$$\mathbf{Y} \cdot \mathbf{c}_r = \sum_{i=0}^{n-1} \Lambda_i \delta\left(y_i, c_{ri}\right) \tag{3.1}$$

onde,

$$\delta\left(v, v'\right) = \begin{cases} +1, & v = v' \\ -1, & v \neq v' \end{cases} \tag{3.2}$$

Em 1972, David Chase propôs uma classe de algoritmos para decodificação com informação de medição de canal [12], podendo ser empregada em três formas, ou algoritmos, que se diferenciam pela relação entre complexidade e capacidade de correção. Esta dissertação utiliza os algoritmos de Chase como método de decodificação por decisão suave devido ao seu melhor desempenho em relação ao algoritmo GMD. Desta maneira, o método de Chase será melhor detalhado na seção 3.2.

Quando utilizamos códigos de bloco sobre o $GF(q)$, ou seja, com símbolos q-ários, por exemplo, os códigos de Reed-Solomon, o uso de algoritmos de decodificação binária por decisão suave exige que conversões símbolo-bit e bit-símbolo sejam realizadas. Supomos que estas conversões podem ocasionar perda de desempenho no algoritmo de decodificação binária, pois as propriedades do código q-ário podem não ser aproveitadas da melhor maneira possível. Quando utilizamos métodos de modulação digital espectralmente mais eficientes que o BPSK, ou seja, que possuem mais de dois símbolos no diagrama de constelação, por exemplo, 16-PSK e 64-QAM, o uso de algoritmos de decodificação binária por decisão suave também pode resultar em perda de rendimento. Neste caso, a informação de medição de canal, que foi obtida considerando o símbolo da modulação transmitido, deverá ser manipulada a fim de se obter uma confiabilidade por bit. Supomos que esta manipulação pode resultar em perdas de desempenho.

Sabemos que os algoritmos de Chase possuem um ótimo desempenho, porém não suportam a utilização de símbolos q-ários. Para tentar suprir esta deficiência dos algoritmos de Chase, propomos nesta dissertação uma modificação no método original de Chase a fim de se trabalhar diretamente com símbolos q-ários, ou seja, com elementos do $GF(q)$, evitando conversões símbolo-bit e bit-símbolo desnecessárias. Com isso, espera-se observar uma melhoria no desempenho do sistema utilizado com a aplicação do algoritmo modificado.

Na seção 3.2, os algoritmos de Chase na sua versão binária (método original) são detalhados. As modificações realizadas no algoritmo de Chase para suportar símbolos q-ários são abordadas na seção 3.3. Por último, na seção 3.4, os resultados das simulações realizadas com as modificações no algoritmo de Chase são apresentados.

3.2 Decodificação de Chase Binária - Método Original

Esta seção é baseada em [12]. Os algoritmos de decodificação com informação de medição de canal propostos por Chase são apresentados. O desenvolvimento original destes algoritmos possibilita apenas o tratamento de palavras compostas de símbolos binários. Porém, mesmo para códigos q-ários sobre o $GF(q)$, com o uso de conversões símbolo-bit e bit-símbolo, estes algoritmos apresentam um desempenho melhor que os métodos algébricos de decodificação, por exemplo: o algoritmo de Berlekamp-Massey para decodificar códigos de Reed-Solomon.

3.2.1 Conceitos da Decodificação de Chase

Relembrando alguns conceitos apresentados no capítulo 2, temos que a função de um decodificador algébrico é encontrar uma palavra-código $\mathbf{X}_m = X_{m1}, X_{m2}, ..., X_{mN}$ que se diferencia no menor número de posições, ou seja, possui a menor distância de Hamming da palavra recebida $\mathbf{Y} = Y_1, Y_2, ..., Y_N$, definida por decisão abrupta no modulador. O decodificador algébrico terá sucesso apenas se o número de posições em que os símbolos se diferenciam em $\mathbf{X}_m$ e $\mathbf{Y}$ for menor ou igual a $\lfloor (d_{min} - 1)/2 \rfloor$, ou seja, menor ou igual à parte inteira do quociente da

divisão $(d_{min} - 1)/2$. A seqüência de erros é a palavra que contém o bit de valor 1 nas posições em que $\mathbf{X}_m$ e $\mathbf{Y}$ se diferenciam e é dada por:

$$\mathbf{Z}_m = \mathbf{Y} \oplus \mathbf{X}_m = Y_1 \oplus X_{m1}, Y_2 \oplus X_{m2}, \ldots, Y_N \oplus X_{mN}, \tag{3.3}$$

onde a notação $\oplus$ representa a soma módulo-2. Desta maneira, o peso binário de $\mathbf{Z}_m$, ou seja, a distância de Hamming entre $\mathbf{Y}$ e $\mathbf{X}_m$ é definida por:

$$W(\mathbf{Z}_m) = \sum_{i=1}^{N} Z_{mi}, \tag{3.4}$$

Então, matematicamente, a função do decodificador algébrico é encontrar a palavra-código, ou equivalentemente, a seqüência de erros que satisfaça:

$$W(\mathbf{Z}_m) \leq \lfloor (d_{min} - 1)/2 \rfloor. \tag{3.5}$$

O decodificador algébrico encontrará uma única palavra-código se a desigualdade (3.5) for satisfeita. Caso contrário, nenhuma palavra-código será escolhida. Neste caso, dependendo da aplicação, algum método de retransmissão de dados pode ser utilizado, por exemplo: o ARQ (do inglês, *Automatic Repeat reQuest*).

Um decodificador baseado em decisão abrupta também pode ser projetado para encontrar a palavra-código que satisfaça

$$\min_m W(\mathbf{Y} \oplus \mathbf{X}_m), \tag{3.6}$$

onde m varia dentre todas as palavras-código possíveis. Este decodificador, conhecido como decodificador completo de decisão abrupta, difere do decodificador algébrico convencional dado em (3.5), pois uma palavra-código sempre será escolhida, independentemente da quantidade de erros existentes na palavra recebida.

De uma maneira similar, podemos definir um decodificador completo de decisão suave como aquele que é capaz de encontrar uma palavra-código que satisfaça:

$$\min_m W_a(\mathbf{Y} \oplus \mathbf{X}_m). \tag{3.7}$$

Neste caso, a função do decodificador é encontrar o padrão de erro $\mathbf{Z}_m = \mathbf{Y} \oplus \mathbf{X}_m$

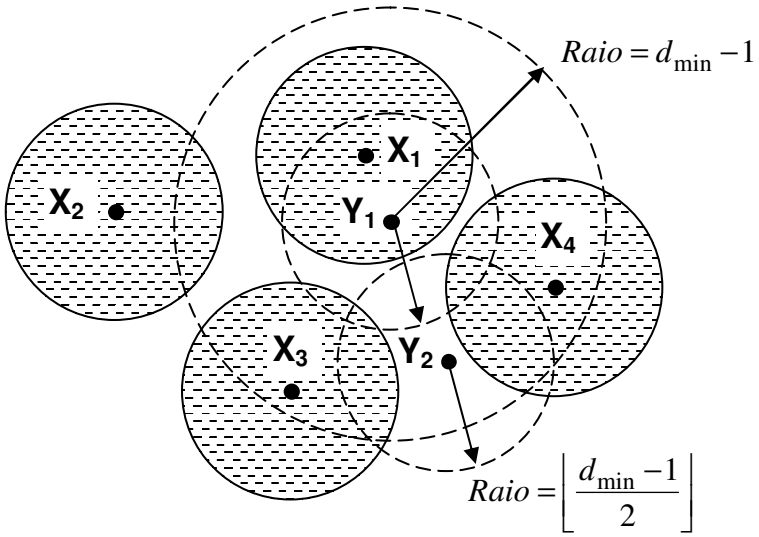

Fig. 3.2: Representação geométrica da decodificação com informação de medição de canal.

de menor peso analógico. Por definição, o peso analógico de uma seqüência $\mathbf{Z}_m$ é dado por:

$$W_a(\mathbf{Z}_m) = \sum_{i=1}^{N} \Lambda_i \cdot Z_{mi}, \tag{3.8}$$

onde $\mathbf{\Lambda} = \Lambda_1, \Lambda_2, ..., \Lambda_N$ representa a medida de confiabilidade dos símbolos da palavra recebida $\mathbf{Y}$ terem sido recebidos corretamente. A confiabilidade de um símbolo recebido é calculada baseada na informação de medição de canal e será detalhada na seção 3.2.3.

Tendo em vista estes três tipos de decodificadores, podemos fazer a seguinte analogia. Da mesma forma que o decodificador algébrico dado em (3.5) procura obter um desempenho mais próximo possível do decodificador completo de decisão abrupta em (3.6), os algoritmos de decodificação com informação de medição de canal desenvolvidos por Chase procuram alcançar um desempenho próximo do decodificador completo de decisão suave em (3.7).

Na figura 3.2, o conceito básico da decodificação com informação de medição de canal é ilustrado. A representação geométrica apresentada na figura 3.2, denota a distância de Hamming entre as quatro palavras-código $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4$ e as palavras recebidas $\mathbf{Y}_1$ e $\mathbf{Y}_2$. Cada palavra-código é limitada por uma região circular de raio

$\lfloor (d_{min}-1)/2 \rfloor$. Desta maneira, uma única palavra-código, ou equivalentemente um único padrão de erro, é obtido pelo decodificador algébrico caso a palavra recebida esteja dentro de uma das regiões circulares.

No caso de utilização de um decodificador algébrico para a palavra recebida $\mathbf{Y}_1$ da figura 3.2, concluímos que a palavra-código escolhida seria $\mathbf{X}_1$, pois temos um único padrão de erro $\mathbf{Z}_1 = \mathbf{Y}_1 \oplus \mathbf{X}_1$ dentro da região circular de raio $\lfloor (d_{min}-1)/2 \rfloor$ que envolve $\mathbf{Y}_1$. Observe que, com a utilização de um decodificador algébrico, não seria possível decodificar a palavra recebida $\mathbf{Y}_2$, pois ela não pertence a nenhuma das regiões circulares que envolvem as palavras-códigos $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ e $\mathbf{X}_4$.

De uma forma geral, o objetivo do decodificador por decisão suave de Chase é obter um conjunto de padrões de erro, através da decodificação algébrica, que são possíveis candidatos a decodificar corretamente a palavra recebida, ao invés de termos apenas uma opção. O padrão de erro candidato escolhido será aquele que possuir o menor peso analógico definido em (3.7). Com isso, a palavra recebida $\mathbf{Y}_2$, ilustrada na figura 3.2, provavelmente seria decodificada pela palavra-código mais próxima e que estivesse dentro de uma outra região circular de raio maior que $\lfloor (d_{min}-1)/2 \rfloor$.

O conjunto de padrões de erros, obtido com o auxílio de um decodificador algébrico, será originado a partir de perturbações programadas que serão realizadas na palavra recebida $\mathbf{Y} = Y_1, Y_2, ..., Y_N$ da saída do demodulador. Estas perturbações serão definidas através de uma seqüência binária $\mathbf{T}_t = T_{t1}, T_{t2}, ..., T_{tN}$ chamada de padrão de teste. As posições do padrão de teste $\mathbf{T}_t$ que possuírem o bit 1, indica que os valores dos bits contidos nas mesmas posições relativas em $\mathbf{Y}$ serão invertidos. Esta nova seqüência gerada pode ser obtida, simplesmente, realizando a operação de adição módulo-2 entre a palavra recebida e o padrão de teste, ou seja:

$$\mathbf{Y}' = \mathbf{Y} \oplus \mathbf{T}_t. \tag{3.9}$$

Desta maneira, um novo padrão de erro $\mathbf{Z}'$ será obtido com o auxílio do decodificador algébrico binário, onde:

$$\mathbf{Z}' = \mathbf{Y}' \oplus \mathbf{X}_m. \tag{3.10}$$

Entretanto, o padrão de erro real relativo à palavra recebida $\mathbf{Y}$ será dado por:

$$\mathbf{Z_T} = \mathbf{T}_t \oplus \mathbf{Z}'. \tag{3.11}$$

O padrão de erro real $\mathbf{Z_T}$, poderá ser diferente ou não do padrão de erro original $\mathbf{Z}_m$ dependendo se a nova seqüência $\mathbf{Y}'$ cairá, ou não, dentro de uma outra região circular de raio $\lfloor (d_{min} - 1)/2 \rfloor$ de uma palavra-código diferente.

Os algoritmos de Chase irão definir os padrões de testes que perturbarão a palavra recebida $\mathbf{Y}$ dentro de uma região circular de raio $d_{min} - 1$. Os detalhes de como estes padrões de testes são obtidos e a sua influência na complexidade da decodificação serão apresentados na seção 3.2.2.

Na figura 3.3, é ilustrado um diagrama de fluxo para a decodificação com informação de medição de canal de Chase. Independentemente do algoritmo de Chase utilizado para gerar os padrões de testes, este diagrama de fluxo deve ser obedecido.

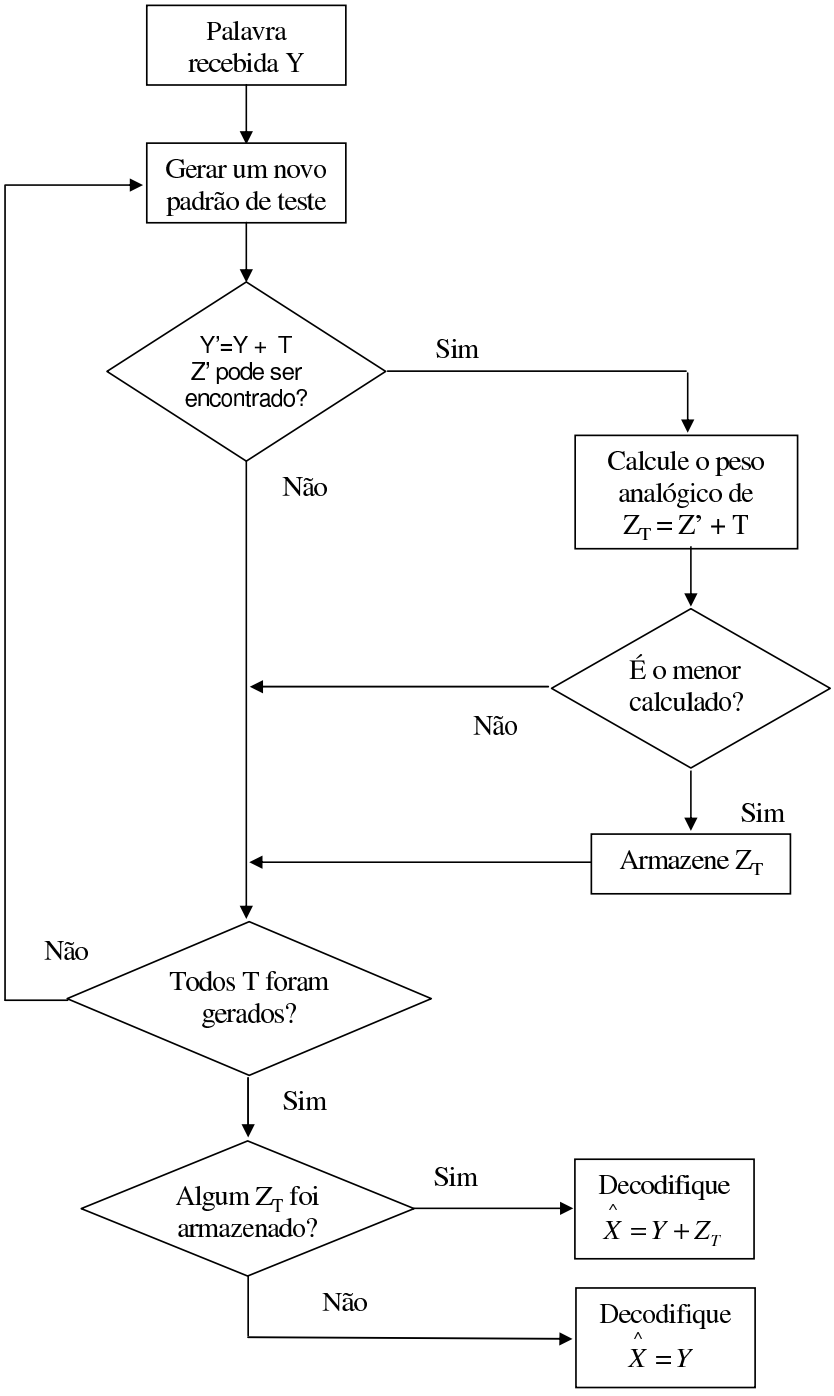

Fig. 3.3: Diagrama de fluxo para a decodificação suave de Chase.

3.2.2 Geração dos Padrões de Teste

O decodificador completo de decisão suave em (3.7) torna-se inviável para algumas aplicações devido a sua complexidade. Para um código de bloco binário (N, K), seriam necessárias 2^K iterações até que o decodificador determinasse a palavra escolhida. Estimulado por esta complexidade na decodificação, Chase definiu uma classe de algoritmos para geração de um conjunto de padrões de teste com cardinalidade menor que 2^K. Assim, o espaço amostral para a procura da palavra escolhida é reduzido, tornando a decodificação menos complexa.

Os algoritmos de Chase, nomeados de *Algoritmo I*, *Algoritmo II* e *Algoritmo III*, diferenciam-se pela complexidade de decodificação, ou seja, a cardinalidade do conjunto de padrões de teste diminui de um algoritmo para o outro. Como estes algoritmos são utilizados com o auxílio do decodificador algébrico binário, capaz de encontrar uma palavra-código apenas se (3.5) for satisfeita, ainda é possível que nenhum padrão de erro seja encontrado. A seguir, os três algoritmos de Chase são apresentados.

Algoritmo I

Dentre os algoritmos de Chase, este é o que possui o conjunto com maior cardinalidade, ou seja, maior número de padrões de teste. Observe a figura 3.2, podemos obter um conjunto com todos os padrões de erros possíveis dentro da região circular centrada na palavra recebida $\mathbf{Y}$ e de raio $(d_{min} - 1)$. Baseado neste conjunto, o *Algoritmo I* de Chase tenta encontrar uma palavra seguindo o diagrama da figura 3.3. Desta maneira, todos padrões de erros de peso binário menor ou igual a $(d_{min} - 1)$ são considerados. Portanto, como o padrão de erro escolhido é determinado pelo peso analógico (3.8), é possível que um padrão de erro com peso binário maior que $\lfloor (d_{min} - 1)/2 \rfloor$ seja selecionado, aumentando assim a capacidade de correção do código.

Se tivermos um conjunto com todas as combinações de seqüências de N-elementos que contêm $\lfloor d_{min}/2 \rfloor$ bits de valor 1, seremos capazes de gerar todos os padrões de erros de peso binário menor que d_{min} com o auxílio do decodificador algébrico binário. Desta maneira, o decodificador algébrico binário capaz de obter padrões

de erro com pesos até $\lfloor (d_{min} - 1)/2 \rfloor$, quando combinado com os padrões de teste apropriados, será capaz de obter qualquer padrão de erro com peso menor ou igual a $(d_{min} - 1)$.

Utilizando o conjunto de padrões de teste mencionado no parágrafo anterior, a quantidade de padrões de teste utilizada neste algoritmo será igual à seguinte combinação:

$$\begin{pmatrix} N \\ \lfloor d_{min}/2 \rfloor \end{pmatrix}.$$

Portanto, o *Algoritmo I* de Chase é aplicável apenas a códigos que possuem uma distância mínima pequena ou a códigos curtos. Uma redução na quantidade de padrões de teste pode ser obtida eliminando os padrões de teste que resultam em padrões de erros idênticos. No entanto, o *Algoritmo I* provavelmente não seria implementado em uma aplicação, já que outros algoritmos mais simples conseguem um desempenho próximo a este.

Apesar da complexidade do *Algoritmo I* influir contra a sua implementação prática, este algoritmo é de grande interesse, pois, segundo [12], ele pode ser modificado para prover um limitante inferior para o decodificador completo de decisão suave e servir como comparativo de desempenho para outros algoritmos de decodificação mais simples.

Algoritmo II

Este algoritmo é um dos mais utilizados devido a sua relação complexidade × desempenho. Uma quantidade menor de padrões de erros é considerada, quando comparado ao *Algoritmo I*. Apenas são considerados os padrões de erro com, no máximo, $\lfloor (d_{min} - 1)/2 \rfloor$ erros que não podem estar localizados nas $\lfloor d_{min}/2 \rfloor$ posições de menor confiabilidade. Assim, os padrões de erro testados contém, no máximo, $(d_{min} - 1)$ erros, mas nem todos os padrões com $d_{min} - 1$ erros, ou menos, serão testados como é feito no *Algoritmo I*.

Um conjunto de padrões de teste capaz de gerar todos os padrões de erro requeridos por este algoritmo pode ser encontrado fazendo com que cada padrão de

teste represente uma amostra de todas combinações possíveis que possuem o bit de valor 1 localizado, no máximo, nas $\lfloor d_{min}/2 \rfloor$ posições de menor confiabilidade. Dado que teremos $2^{\lfloor d_{min}/2 \rfloor}$ padrões de teste possíveis, incluindo a seqüência com o bit de valor 0 em todas a posições, serão considerados neste algoritmo, no máximo, $2^{\lfloor d_{min}/2 \rfloor}$ padrões de erro.

Observe que, se o padrão de erro inicial $\mathbf{Z}$ tiver um peso binário igual a 1, então todos os padrões de teste de peso binário menor ou igual a $\lfloor (d_{min}-3)/2 \rfloor$ resultarão em $\mathbf{Z}_T = \mathbf{Z}$, pois o decodificador algébrico binário utilizado decodificará qualquer $\mathbf{Y}'$, da equação (3.9), com distância de Hamming de X_m menor ou igual a $\lfloor (d_{min}-1)/2 \rfloor$ para a mesma palavra-código X_m.

Algoritmo III

Este algoritmo é bastante semelhante ao *Algoritmo II*, diferenciado-se pelo número de padrões de teste considerado. Utilizamos apenas $\lfloor (d_{min}/2)+1 \rfloor$ padrões de teste. Cada padrão de teste possui i bits de valor 1 localizados nas i posições com menores confiabilidades. Para um código com d_{min} de valor par, os valores que i assume são: $i = 0, 1, 3, ..., d_{min}-1$. Quando d_{min} é um valor ímpar, os valores assumidos por i são: $i = 0, 2, 4, ..., d_{min}-1$. Observe que estas seqüências de padrões de teste são equivalentes às seqüências de apagamento utilizadas na decodificação GMD descrita, ligeiramente, na seção 3.1 e detalhada em [11].

O conjunto de padrões de erro considerado neste algoritmo é o de menor cardinalidade, quando comparado com os *Algoritmos I* e *II*, porém ainda possui um desempenho assintótico de erro semelhante aos observados nos algoritmos anteriores, segundo [12]. Apesar de possuir um desempenho inferior aos dois algoritmos anteriores, este algoritmo é o menos complexo. Assim, para aplicações que utilizam códigos longos com distâncias mínimas grandes, o *Algoritmo III* de Chase pode ser interessante, pois o número de padrões de teste cresce linearmente com a distância mínima do código.

3.2.3 Cálculo da Confiabilidade

Nesta seção, o método utilizado para a obtenção da informação de medição de canal, baseado na forma de onda recebida $y(t)$, é abordado. Vimos na figura 3.1 que o demodulador é responsável por calcular a informação de canal e repassá-la ao decodificador. Na figura 3.4, são mostrados os receptores de forma de onda pertencentes ao demodulador da figura 3.1. Observe que, na saída dos filtros de decisão, temos as estatísticas de decisão ponderadas. A estatística de decisão, denotada por v_i, determinará a regra de decisão que definirá o valor do bit recebido Y_i, segundo as seguintes desigualdades:

$$\begin{aligned} &para \quad v_i \geq 0, \qquad \text{faça} \quad Y_i = 1 \\ &para \quad v_i \leq 0, \qquad \text{faça} \quad Y_i = 0. \end{aligned} \tag{3.12}$$

O projeto dos filtros de decisão dependerá do tipo do sinal de transmissão e do tipo do canal considerado nesta transmissão. Os filtros de decisão não são os objetivos desta dissertação, portanto seu projeto não será abordado.

O Logaritmo da Razão de Verossimilhança (LLR, do inglês *Log-Likelihood Ratio*) de Y_i condicionado a v_i é definido como:

$$L(Y_i|v_i) = \ln \frac{P(Y_i = 1|v_i)}{P(Y_i = 0|v_i)}. \tag{3.13}$$

Utilizando o Teorema de Bayes, o logaritmo da razão de verossimilhança pode ser escrito da seguinte forma:

$$\begin{aligned} L(Y_i|v_i) &= \ln \left(\frac{p(v_i|Y_i = 1)}{p(v_i|Y_i = 0)} \frac{P(Y_i = 1)}{P(Y_i = 0)} \right) \\ &= \ln \left(\frac{p(v_i|Y_i = 1)}{p(v_i|Y_i = 0)} \right) + \ln \left(\frac{P(Y_i = 1)}{P(Y_i = 0)} \right). \end{aligned} \tag{3.14}$$

Considerando que a fonte de informação gera bits equiprováveis ou que um embaralhador seja utilizado para fazer tal função, teremos $P(Y_i = 0) = P(Y_i = 1)$. Assim, a equação (3.14) se reduz a:

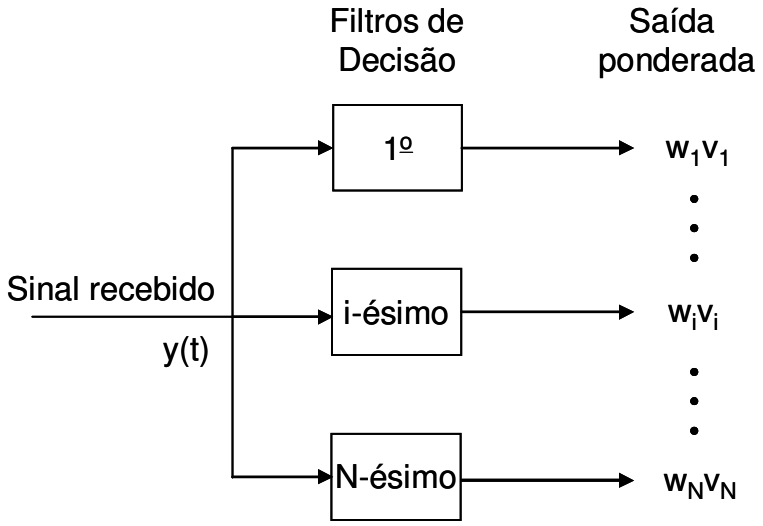

Fig. 3.4: Receptores de forma de onda.

$$L(Y_i|v_i) = \ln\left(\frac{p(v_i|Y_i=1)}{p(v_i|Y_i=0)}\right). \tag{3.15}$$

Para um canal gaussiano com média μ, variância σ^2 e função densidade de probabilidade $f(z)$ igual a:

$$f(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(z-\mu)^2}{2\sigma^2}}. \tag{3.16}$$

Podemos mostrar que:

$$\begin{aligned} L(Y_i|v_i) &= \ln\left(\frac{p(v_i|Y_i=1)}{p(v_i|Y_i=0)}\right) \\ &= \ln\left(\frac{\exp^{-\frac{E_b}{N_o}(v_i-a)^2}}{\exp^{-\frac{E_b}{N_o}(v_i+a)^2}}\right) \end{aligned}$$

$$= \ln\left(\frac{\exp^{+\frac{E_b}{N_o}2av_i}}{\exp^{-\frac{E_b}{N_o}2av_i}}\right)$$

$$= 4\frac{E_b}{N_o}a\,v_i, \tag{3.17}$$

onde $\frac{E_b}{N_o}$ é a relação sinal ruído por bit e a é a amplitude de desvanecimento. Para canais gaussianos sem desvanecimento, temos $a = 1$.

O sinal de $L(Y_i|v_i)$, positivo ou negativo, pode ser usado para indicar a decisão abrupta Y_i, bit 1 ou bit 0. Observe que à medida que $L(Y_i|v_i)$ cresce, podemos dizer que $P(Y_i = 1|v_i)$ também cresce. Similarmente, à medida que $L(Y_i|v_i)$ decresce, implica no crescimento de $P(Y_i = 0|v_i)$. Desta maneira, $L(Y_i|v_i)$ fornece uma medida de confiabilidade do símbolo recebido Y_i. Portanto, se normalizarmos a equação (3.17), podemos utilizar como confiabilidade Λ_i do símbolo recebido Y_i, transmitido através de um canal gaussiano, a seguinte expressão:

$$\Lambda_i = |v_i| \tag{3.18}$$

Então, comparando com a saída dos filtros de decisão da figura 3.4 e fazendo o fator de ponderação $w_i = 1$, podemos obter a confiabilidade Λ_i do símbolo recebido Y_i apenas extraindo o valor absoluto da decisão estatística v_i.

3.2.4 Resultados

Nesta seção, é apresentado o desempenho do procedimento de Chase como método de decodificação com a informação da medição de canal. O *Algoritmo II* de Chase foi o escolhido para a implementação do decodificador suave devido a sua boa relação complexidade × capacidade de correção. O desempenho do método de Chase é comparado com o desempenho da decodificação algébrica binária.

As simulações foram baseadas no diagrama em blocos ilustrado na figura 3.1. A fonte de informação utilizada fornece uma seqüência de bits equiprováveis e com distribuição uniforme. A ordem de grandeza da quantidade de bits de informação

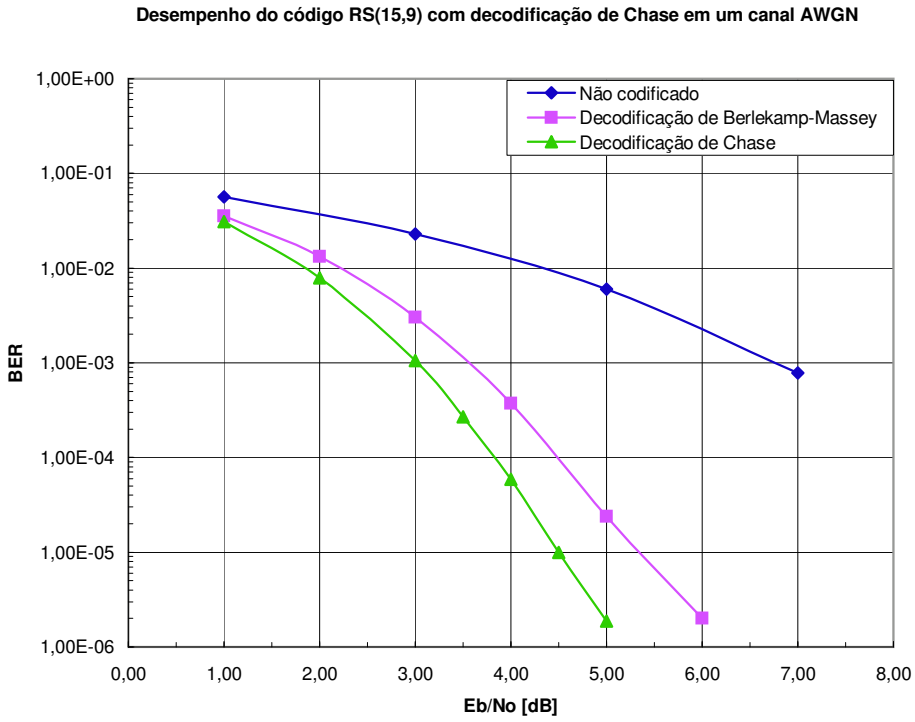

Fig. 3.5: Desempenho do código RS(15,9) utilizando decodificação algébrica de Berlekamp-Massey e decodificação suave de Chase-II.

gerados foi de 10^7, a quantidade exata dependeu dos parâmetros do código corretor utilizado. Desta maneira, taxas de erro (BER, do inglês *Bit Error Rate*) com ordem de grandeza de 10^{-6} puderam ser obtidas.

A escolha do código de Reed-Solomon como o código corretor de erro utilizado nas simulações foi baseada nas seguintes condições:

- Por ser um código de bloco: condição necessária para comparação com os resultados obtidos no capítulo 5 - Códigos Turbo de Bloco;
- Por permitir trabalhar no GF(q), ou seja, um código q-ário: condição necessária para comparação com os resultados que serão apresentados na próxima seção.

Foram utilizados três códigos Reed-Solomon diferentes: RS(15,9), RS(31,25) e RS(31,19), motivados pelas mesmas justificativas de comparação de desempenho mencionadas nos itens acima.

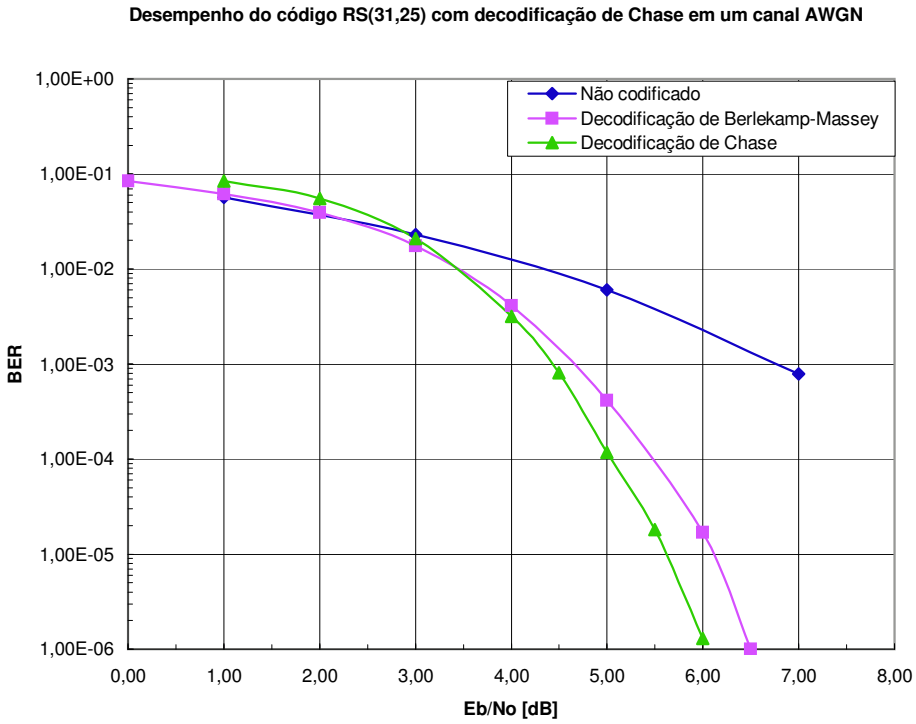

Fig. 3.6: Desempenho do código RS(31,25) utilizando decodificação algébrica de Berlekamp-Massey e decodificação suave de Chase-II.

O modulador irá modular o sinal de entrada de acordo com a constelação BPSK (do inglês, *Binary Phase Shift Keying*) e, em seguida, transmitirá o sinal modulado por um canal AWGN (do inglês, *Additive White Gaussian Noise*). Como o canal adotado é o AWGN, quando a decodificação de Chase for empregada, a equação (3.18) será utilizada para a obtenção da confiabilidade do símbolo transmitido.

Na figura 3.5, são apresentadas as curvas de $BER \times E_b/N_o$ que representam o desempenho do código RS(15,9) com decodificação algébrica de Berlekamp-Massey [13] e com decodificação suave de Chase-II (*Algoritmo II*) em um canal AWGN. Observe que a decodificação algébrica binária apresenta um enorme ganho de desempenho quando comparada com o sistema não-codificado. O desempenho do código RS(15,9) é melhorado ainda mais quando utilizamos a decodificação suave de Chase-II, obtendo um ganho de $1dB$ sobre a decodificação algébrica para uma taxa de erro de aproximadamente $2 \cdot 10^{-6}$.

Como esperado, o mesmo comportamento de desempenho é observado na

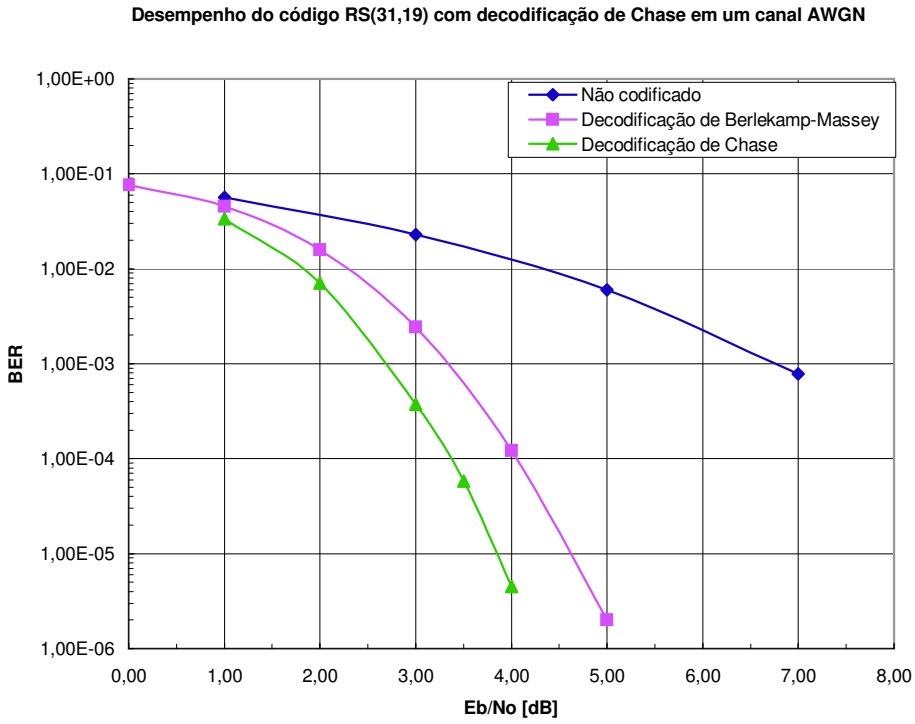

Fig. 3.7: Desempenho do código RS(31,19) utilizando decodificação algébrica de Berlekamp-Massey e decodificação suave de Chase-II.

figura 3.6 com a utilização do código RS(31,25). Porém, o ganho obtido com a decodificação de Chase-II, sobre o decodificador algébrico, é menor que o observado para o código RS(15,9), apesar de terem a mesma $d_{min} = 7$. Para um taxa de erro de 10^{-6}, temos um ganho de $0,5dB$. Observe que, para obter uma taxa de erro da ordem de 10^{-6}, o sistema com o RS(31,25) e decodificação suave de Chase-II necessita de uma relação $E_b/N_o = 6dB$, enquanto a mesma decodificação para o RS(15,9) é necessária uma $E_b/N_o = 5dB$. Esta queda de desempenho do código RS(31,25) é justificada por ele possuir uma taxa de codificação $r \approx 0,806$ maior que a encontrada no RS(15,9), $r = 0,6$, o que implica em uma diferença na proporção de bits redundantes.

Dentre os códigos analisados, o RS(31,19) é o que apresenta o melhor desempenho com a utilização da decodificação suave de Chase-II. Na figura 3.7, esta melhoria fica evidente com a observação do aumento da inclinação das curvas de $BER \times E_b/N_o$. O RS(31,19) com decodificação de Chase-II possui um ganho sobre a decodificação

algébrica de Berlekamp-Massey muito próximo do obtido com o código RS(15,9), apesar de possuir uma distância mínima bem maior. Observe que a taxa de codificação do RS(31,19) é $r \approx 0,613$, ou seja, próxima da taxa do RS(15,9). Porém, para obter uma taxa de erro de $4 \cdot 10^{-6}$ o código RS(31,19) com decodificação suave de Chase-II necessita apenas de uma relação $E_b/N_o = 4dB$.

Com estes resultados, podemos concluir que o ganho da decodificação suave de Chase-II, sobre a decodificação algébrica de Berlekamp-Massey, depende da taxa do código utilizado. Assim, quanto menor a taxa do código, maior será o ganho. Por outro lado, o valor de E_b/N_o necessário para se obter uma determinada taxa de erro, depende tanto da distância mínima do código quanto de sua taxa de codificação.

3.3 Algoritmo de Chase q-ário - Método Adaptado

Na seção 3.2, vimos que a utilização dos algoritmos de Chase como método de decodificação com informação de medição de canal aumenta a capacidade de correção do código, resultando em um ganho de desempenho sobre os métodos de decodificação algébrica. Porém, o método de decodificação desenvolvido por Chase é aplicável apenas em seqüências de símbolos binários.

Algumas aplicações de sistemas de transmissão digital requerem a utilização de códigos q-ários, sobre o $GF(q)$, devido à possibilidade de se obter uma maior distância mínima entre as palavras-códigos, ou seja, uma maior capacidade de correção de erros, sem onerar a eficiência espectral do sistema, que é prejudicada com o uso de taxas de codificação menores. Além disso, os códigos de blocos q-ários são bastante utilizados em aplicações que necessitam uma maior proteção para erros em rajada (do inglês, *Burst Errors*).

Quando utilizamos os algoritmos de Chase binários para decodificar códigos q-ários, conversões símbolo-bit e bit-símbolo devem ser realizadas para possibilitar a decodificação. Estas conversões podem comprometer o desempenho da decodificação de Chase. Ilustraremos, como exemplo dessa perda de desempenho, uma aplicação com o código RS(7,3) que possui $d_{min} = N - K + 1 = 7 - 3 + 1 = 5$ e, conseqüentemente, capacidade de correção de t erros, sendo:

Palavra-código transmitida:

$$X_A = [\ 3 \quad 1 \quad 7 \quad 7 \quad 5 \quad 1 \quad 5\ ]$$

$$X_{Abits} = [011\ 001\ 111\ 111\ 101\ 001\ 101]$$

Palavra recebida:

$$Y = [\ 4 \quad 1 \quad 7 \quad 0 \quad 5 \quad 6 \quad 2\ ]$$

$$Y_{bits} = [100\ 001\ 111\ 000\ 101\ 110\ 010]$$

Erros:

$$Z = X_A \oplus Y = [\ 7 \quad 0 \quad 0 \quad 7 \quad 0 \quad 7 \quad 7\ ]$$

$$Z_{bits} = X_{Abits} \oplus Y_{bits} = [111\ 000\ 000\ 111\ 000\ 111\ 111]$$

Fig. 3.8: Exemplo de erro em uma palavra do código RS(7,3) transmitida.

$$t = \left\lfloor \frac{(d_{min} - 1)}{2} \right\rfloor = \left\lfloor \frac{(5 - 1)}{2} \right\rfloor = 2.$$

Observe a figura 3.8, suponha que uma palavra-código $\mathbf{X}_A$ seja transmitida e a palavra recebida na entrada do decodificador seja $\mathbf{Y}$. Fazendo as conversões símbolo-bit das seqüências $\mathbf{X}_A$ e $\mathbf{Y}$, teremos $\mathbf{X}_{Abits}$ e $\mathbf{Y}_{bits}$, respectivamente. Utilizando o *Algoritmo II* de Chase para aumentar a capacidade de correção de erros, podemos manipular a palavra-recebida $\mathbf{Y}_{bits}$ através de $2^{\lfloor d_{min}/2 \rfloor} = 2^{\lfloor 5/2 \rfloor} = 4$ padrões de teste que possibilitam modificações, no máximo, nas $\lfloor d_{min}/2 \rfloor = \lfloor 5/2 \rfloor = 2$ posições de menores confiabilidades da palavra recebida $\mathbf{Y}_{bits}$. Porém, mesmo aplicando este algoritmo de Chase a $\mathbf{Y}_{bits}$ e fazendo a conversão bit-símbolo, para obter a nova seqüência $\mathbf{Y}'$, o decodificador algébrico não conseguiria decodificar $\mathbf{Y}'$ corretamente, pois ele é capaz de corrigir apenas $t = 2$ erros de símbolos. Para decodificar a palavra-recebida $\mathbf{Y}$ corretamente, utilizando o método de Chase, precisaríamos manipular a palavra $\mathbf{Y}_{bits}$ pelo menos em 2 seqüências de 3 posições seguidas, totalizando 6 posições. É possível desenvolver um algoritmo que atenda esta necessidade, porém ele seria mais complexo e com uma quantidade grande de padrões de teste.

Com o intuito de melhorar o desempenho da decodificação de Chase para códigos q-ários, é proposta uma modificação no método de Chase para generalizar a sua aplicação. Observe novamente a figura 3.8, se trabalhássemos diretamente com a palavra recebida $\mathbf{Y}$, ao invés de realizarmos a conversão símbolo-bit para obter $\mathbf{Y}_{bits}$, e conseguíssemos manipulá-la através de padrões de teste que permitissem a substituição dos símbolos q-ários das posições com baixa confiabilidade por outros símbolos q-ários mais adequados, seriam necessárias substituições apenas em duas posições de $\mathbf{Y}$ para que o decodificador algébrico conseguisse decodificar a nova sequência $\mathbf{Y}^{'}$ em uma palavra válida do código de RS(7,3). Exemplificando, se tivermos o padrão de teste $\mathbf{T} = [7\ 0\ 0\ 7\ 0\ 0\ 0]$ e o somarmos, no GF(8), à palavra recebida $\mathbf{Y}$, teremos uma nova palavra:

$$\begin{aligned} \mathbf{Y}^{'} &= \mathbf{Y} \oplus \mathbf{T} \\ &= [4\ \ 1\ \ 7\ \ 0\ \ 5\ \ 6\ \ 2] \oplus [7\ \ 0\ \ 0\ \ 7\ \ 0\ \ 0\ \ 0] \\ &= [3\ \ 1\ \ 7\ \ 7\ \ 5\ \ 6\ \ 2]. \end{aligned}$$

Note que a seqüência q-ária $\mathbf{Y}^{'}$ agora pode ser decodificada pelo decodificador algébrico q-ário, pois possui apenas dois erros de símbolo quando comparada à $\mathbf{X}_A$.

De acordo com o exemplo anterior, o método de Chase modificado para suportar a decodificação de códigos q-ários, sem a necessidade de conversões bit-símbolo e símbolo-bit, deve seguir a mesmo esquema de decodificação ilustrado no diagrama de fluxo da figura 3.3. Porém, a obtenção do conjunto de padrões de teste difere das formas utilizadas pelos três algoritmos originais de Chase, pois agora trabalharemos com símbolos q-ários. Na seção 3.3.1, a geração dos padrões de teste para utilizar na decodificação de Chase q-ária é abordada. O cálculo das confiabilidades dos símbolos recebidos também deve ser modificado para atender o novo algoritmo de obtenção dos padrões de teste q-ários. Na seção 3.3.2, esta nova forma de calcular a confiabilidade será apresentada.

3.3.1 Geração dos Padrões de Teste q-ários

Devido a boa relação entre complexidade e desempenho alcançado pelo *Algoritmo II* de Chase, conforme apresentado na seção 3.2.2, a geração do conjunto de padrões de teste para a adaptação da decodificação de Chase à códigos q-ários será baseada apenas naquele algoritmo.

Relembrando, o *Algoritmo II* de Chase binário gera um conjunto de padrões de teste que possui todas as combinações de seqüências que possuem o bit de valor 1, no máximo, nas $\lfloor d_{min}/2 \rfloor$ posições de menores confiabilidades da palavra recebida $\mathbf{Y}$. Quando trabalhamos com códigos q-ários, vimos que o desempenho da decodificação de Chase pode ser degradado, conforme o exemplo ilustrado na figura 3.8. Isto ocorre porque, após a conversão símbolo-bit, a seqüência binária $\mathbf{Y}_{bits}$, que será manipulada pelos padrões de teste binários, torna-se $\log_2(q)$ vezes maior que a seqüência q-ária $\mathbf{Y}$, porém a quantidade de posições de $\mathbf{Y}_{bits}$ que serão modificadas se mantêm em $\lfloor d_{min}/2 \rfloor$.

Poderíamos alterar o *Algoritmo II* de Chase simplesmente aumentando a quantidade de posições da palavra $\mathbf{Y}_{bits}$ que seriam modificadas pelos padrões de teste, proporcionalmente ao aumento do número de posições da seqüência $\mathbf{Y}$ para a seqüência $\mathbf{Y}_{bits}$, ou seja, $\log_2(q)$ vezes. Desta maneira, manipularíamos $\lfloor d_{min}/2 \rfloor \times \log_2(q)$ posições de $\mathbf{Y}_{bits}$, resultando em uma quantidade de padrões de teste igual a:

$$\begin{aligned} 2^{\lfloor d_{min}/2 \rfloor \times \log_2(q)} &= \\ &= \left(2^{\log_2(q)}\right)^{\lfloor d_{min}/2 \rfloor} \\ &= q^{\lfloor d_{min}/2 \rfloor}. \end{aligned} \tag{3.19}$$

Apesar desta ser uma solução simples, note que a quantidade de padrões de teste cresceria exponencialmente de acordo com a dimensão do corpo de Galois do código q-ário utilizado. Com isso, a aplicação deste algoritmo de Chase modificado poderia ser inexeqüível devido a sua complexidade. Exemplificando, com a utilização do código RS(7,3) sobre o GF(8) ilustrado na figura 3.8, teríamos um aumento na quantidade de padrões de teste de $2^{\lfloor d_{min}/2 \rfloor} = 2^{\lfloor 5/2 \rfloor} = 4$ para $q^{\lfloor d_{min}/2 \rfloor} = 8^{\lfloor 5/2 \rfloor} = 64$.

Vale lembrar que os códigos de Reed-Solomon mais utilizados estão sobre os GF(32), GF(64) e GF(128), o que inviabiliza ainda mais o uso deste algoritmo de Chase modificado.

Observando a equação (3.19), podemos pensar que, ao invés de fazermos substituições nas seqüências binárias, poderíamos trabalhar com as próprias seqüências q-árias. Então, o conjunto de padrões de teste utilizado possuiria todas as combinações possíveis de seqüências q-árias que possuem os elementos do GF(q) nas $\lfloor d_{min}/2 \rfloor$ posições de menor confiabilidade da palavra recebida $\mathbf{Y}$. Apesar desta mudança eliminar as conversões símbolo-bit e bit-símbolo, o problema da complexidade permaneceria, pois ainda teríamos $q^{\lfloor d_{min}/2 \rfloor}$ padrões de teste.

Sabemos que a seqüência recebida $\mathbf{Y}$ é gerada no demodulador através de decisão abrupta e que o símbolo escolhido $Y_i = m$ obedece a seguinte desigualdade:

$$P(Y_i = m|v_i) > P(Y_i = j|v_i), \quad \text{para todo } j \neq m, \tag{3.20}$$

onde: v_i é a decisão estatística, ilustrada na figura 3.4, correspondente ao i-ésimo símbolo transmitido e j e $m \in \{0,\ 1,\ 2,\ ...,\ (q-1)\}$.

Como o auxílio da desigualdade (3.20), podemos obter uma seqüência $\mathbf{Y}^a$ que contenha o símbolo $Y_i^a = k$ tal que:

$$P(Y_i = m|v_i) > P(Y_i = k|v_i) > P(Y_i = j|v_i), \quad \text{para todo } j \neq m \neq k, \tag{3.21}$$

ou seja, a seqüência $\mathbf{Y}^a$ conterá os símbolos Y_i^a que possuem a segunda maior probabilidade de terem sido transmitidos dado que v_i foi observado.

De posse da seqüência $\mathbf{Y}^a$, podemos encontrar um conjunto de padrões de teste que possua todas as combinações possíveis capazes de fazer a seqüência $\mathbf{Y}'$, da equação (3.9), assumir alternativamente os símbolos das seqüências $\mathbf{Y}$ e $\mathbf{Y}^a$ pertencentes às $\lfloor d_{min}/2 \rfloor$ posições de menores confiabilidades. Note que os padrões de teste devem possuir símbolos nulos nas posições que não estão entre as $\lfloor d_{min}/2 \rfloor$ menores confiabilidades, pois aquelas posições devem assumir os mesmos valores encontrados em $\mathbf{Y}$. Portanto, os valores dos símbolos dos padrões de teste devem seguir:

$$T_i = \begin{cases} Y_i^a - Y_i, & \textit{quando } Y_i^{'} = Y_i^a \\ 0, & \textit{quando } Y_i^{'} = Y_i \end{cases} \tag{3.22}$$

Observe que os padrões de teste gerados para suportar a decodificação de Chase q-ária assemelha-se bastante com os padrões de teste binários do *Algoritmo II* de Chase, sendo que, ao invés de possuir o bit de valor 1 nas posições onde haverá inversão, agora os valores assumidos obedecem a equação (3.22). Desta maneira, apesar de utilizarmos a decodificação de Chase para símbolos q-ários, a quantidade de padrões de teste utilizada na decodificação continua a mesma do *Algoritmo II* binário, ou seja, $2^{\lfloor d_{min}/2 \rfloor}$, o que torna a sua aplicação viável.

3.3.2 Cálculo da Confiabilidade q-ária

A proposta de adaptação do método original da decodificação de Chase, apresentada anteriormente, para suportar o uso de uma decodificação suave que trabalha diretamente com símbolos q-ários, requer também uma modificação na metodologia para calcular a confiabilidade de cada símbolo, agora q-ário, da palavra recebida.

Para símbolos binários, vimos que a confiabilidade pode ser obtida conforme a equação (3.13). Neste caso, como temos apenas dois símbolos, é preciso apenas calcular o logaritmo neperiano da razão entre as probabilidades a posteriori dos bits 0 e 1. Para o caso q-ário, é necessário o cálculo das probabilidades a posteriori dos símbolos de todo o alfabeto. Além disso, deve-se ordenar as probabilidades a posteriori de tal modo que a maior delas seja identificada. Desta maneira, o Logaritmo da Razão de Verossimilhança, LLR, de um símbolo q-ário Y_i pode ser calculado conforme a equação abaixo:

$$L(Y_i|V_i) = \ln\left(\frac{P(Y_i^{(1)}|V_i)}{\sum_{j=2}^{q} P(Y_i^{(j)}|V_i)}\right), \tag{3.23}$$

onde: $Y_i^{(j)}$ representa o símbolo do alfabeto q-ário que gera a j-ésima maior probabilidade a posteriori condicionada a V_i, ou seja, $P(Y_i^{(j)}|V_i)$.

Observe que o símbolo $Y_i^{(1)}$ representará a posição i da palavra recebida $\mathbf{Y}$ e o símbolo $Y_i^{(2)}$ será o valor correspondente a posição i da seqüência $\mathbf{Y}^a$, que será

utilizada para gerar o conjunto dos padrões de teste q-ários, conforme apresentado na seção 3.3.1.

Aplicando o Teorema de Bayes à equação (3.23), o logaritmo da razão de verossimilhança pode ser reescrito da seguinte forma:

$$L(Y_i|V_i) = \ln\left(\frac{P(V_i|Y_i^{(1)}) \cdot P(Y_i^{(1)})}{\sum_{j=2}^{q} P(V_i|Y_i^{(j)}) \cdot P(Y_i^{(j)})}\right). \tag{3.24}$$

Considerando que a fonte de informação gera símbolos q-ários equiprováveis ou que um embaralhador seja utilizado para fazer tal função, teremos $P(Y_i^{(j)}) = cte$, para $j = 1, 2, ..., q$. Assim, a equação (3.24) se reduz a:

$$L(Y_i|V_i) = \ln\left(\frac{P(V_i|Y_i^{(1)})}{\sum_{j=2}^{q} P(V_i|Y_i^{(j)})}\right). \tag{3.25}$$

Observe que, diferentemente do caso binário, não podemos utilizar o valor absoluto do logaritmo da razão de verossimilhança como confiabilidade do símbolo recebido, pois podem ocorrer casos em que o valor do denominador da equação (3.25) seja maior que o valor do numerador, ou seja, apesar de $Y_i^{(1)}$ ser o símbolo que gera a maior $P(V_i|Y_i)$, esta pode ser superada pelo somatório das probabilidades para os outros símbolos. Neste caso, $L(Y_i|V_i)$ será negativo e quanto menor for o seu valor, menor será a confiabilidade do símbolo recebido.

Mesmo utilizando uma fonte de informação q-ária e codificação de canal q-ária, haverão aplicações que utilizarão uma conversão símbolo-bit para adequar os símbolos de entrada do modulador à constelação utilizada. O processo inverso deve ser realizado na recepção, ou seja, um conversor bit-símbolo deve ser utilizado após o demodulador para adequar os símbolos à entrada do decodificador q-ário. Nestes casos, a equação (3.25) deve ser utilizada equivalentemente na seguinte forma:

$$L(Y_i|V_i) = \ln\left(\frac{\prod_{k=1}^{\log_2 q} P(v_{ik}|y_{ik}^{(1)})}{\sum_{j=2}^{q} \prod_{k=1}^{\log_2 q} P(v_{ik}|y_{ik}^{(j)})}\right), \tag{3.26}$$

onde as conversões binárias dos símbolos Y_i e V_i são representadas, respectivamente, por $[y_{i0} \;\; y_{i1} \;\; ... \;\; y_{i\log_2 q}]$ e $[v_{i0} \;\; v_{i1} \;\; ... \;\; v_{i\log_2 q}]$.

Observe que as conversões realizadas para adequar os símbolos q-ários ao canal

de transmissão (modulador + meio de transmissão + demodulador) não impactam no desempenho do método q-ário adaptado da decodificação de Chase, pois são transparentes ao decodificador, mudando apenas a forma de calcular a confiabilidade do símbolo de entrada do decodificador.

3.4 Análise dos Resultados

Nesta seção, são apresentados os resultados das simulações realizadas para avaliar o desempenho da adaptação proposta do método de decodificação de Chase para símbolos q-ários. Os procedimentos propostos na seção 3.3 foram implementados para a obtenção dos resultados. O algoritmo original de Chase II para símbolos binários foi escolhido para comparação com método proposto, pois possuem a mesma complexidade, ou seja, mesma quantidade de padrões de testes.

Basicamente, as simulações foram baseadas no diagrama em blocos da figura 3.1, modificando apenas o tipo de entrada do decodificador, que agora serão símbolos q-ários. A ordem de grandeza da quantidade de bits gerados pela fonte de informação foi de 10^7, a quantidade exata dependeu dos parâmetros do código corretor de erro utilizado. Desta maneira, taxas de erro de bit com ordem de grandeza de 10^{-6} puderam ser obtidas.

Foram utilizados três códigos de Reed-Solomon diferentes: RS(15,9), RS(31,25) e RS(31,19). Estes códigos foram escolhidos por motivos de comparação com os resultados do método original de Chase apresentados na seção 3.2.4.

Na figura 3.9, são apresentadas as curvas $BER \times E_b/N_o$ que representam o desempenho do código RS(15,9) com a decodificação original de Chase II e com a adaptação proposta do método de Chase para símbolos q-ários em um canal AWGN. Observe que o método adaptado para a decodificação de Chase de símbolos q-ários apresenta um desempenho melhor que o método original de Chase II. Perceba que quanto maior a relação E_b/N_o, maior será a melhoria de desempenho do método proposto em relação ao método original. Para uma $E_b/N_o = 5dB$, o método proposto apresenta uma $BER \approx 4 \times 10^{-7}$, enquanto que o método original resulta em uma $BER \approx 2 \times 10^{-6}$ para a mesma relação E_b/N_o, o que implica em uma melhoria de cinco vezes na BER.

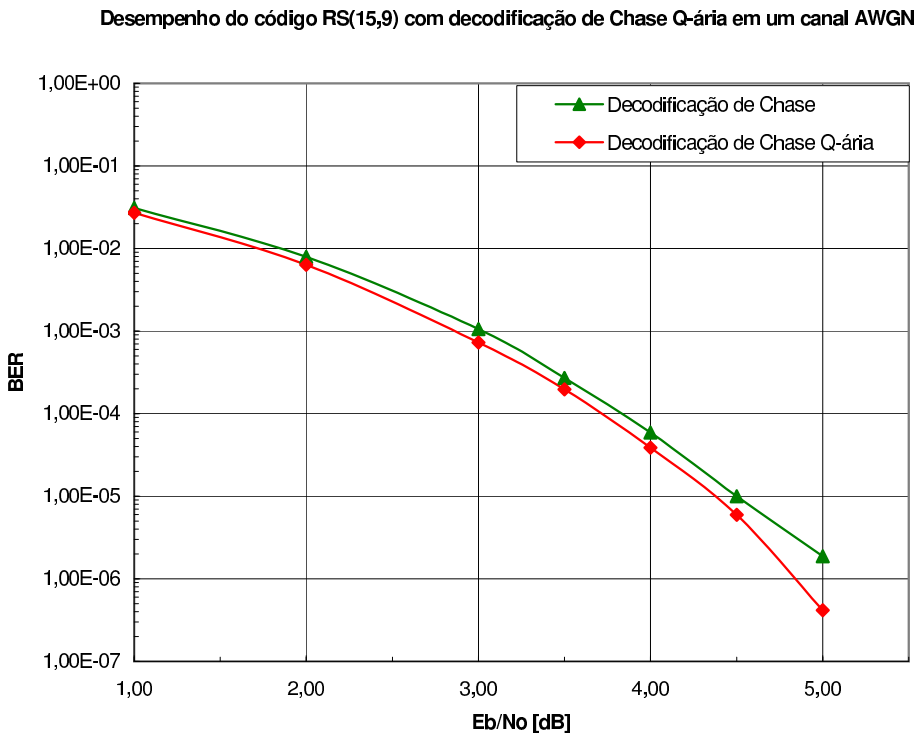

Fig. 3.9: Desempenho do código RS(15,9) utilizando decodificação suave de Chase-II e sua adaptação para símbolos q-ários.

Conforme esperado, também pode-se observar na figura 3.10 uma melhoria de desempenho do sistema com a utilização do método proposto para um código RS(31,25). O ganho obtido com o método adaptado proposto sobre o método original de decodificação de Chase II foi de, aproximadamente, $0,5dB$, para uma $BER \approx 1 \times 10^{-6}$. Como também observado na figura 3.9, o ganho de desempenho do método adaptado proposto sobre o método original aumenta conforme o crescimento da relação E_b/N_o. Observe que para uma $E_b/N_o = 6dB$, tem-se uma $BER \approx 1 \times 10^{-6}$ com o método original de decodificação de Chase II, já para o método adaptado proposto, tem-se uma $BER \approx 1 \times 10^{-7}$ para a mesma E_b/N_o, o que resulta em uma melhoria de, aproximadamente, dez vezes na taxa de erro de bit.

Dentre os códigos analisados, o RS(31,19) é o que possui maior capacidade de correção. Apesar de seu comportamento se assemelhar com o apresentado nas figuras 3.9 e 3.10, ou seja, o ganho de desempenho do método adaptado proposto

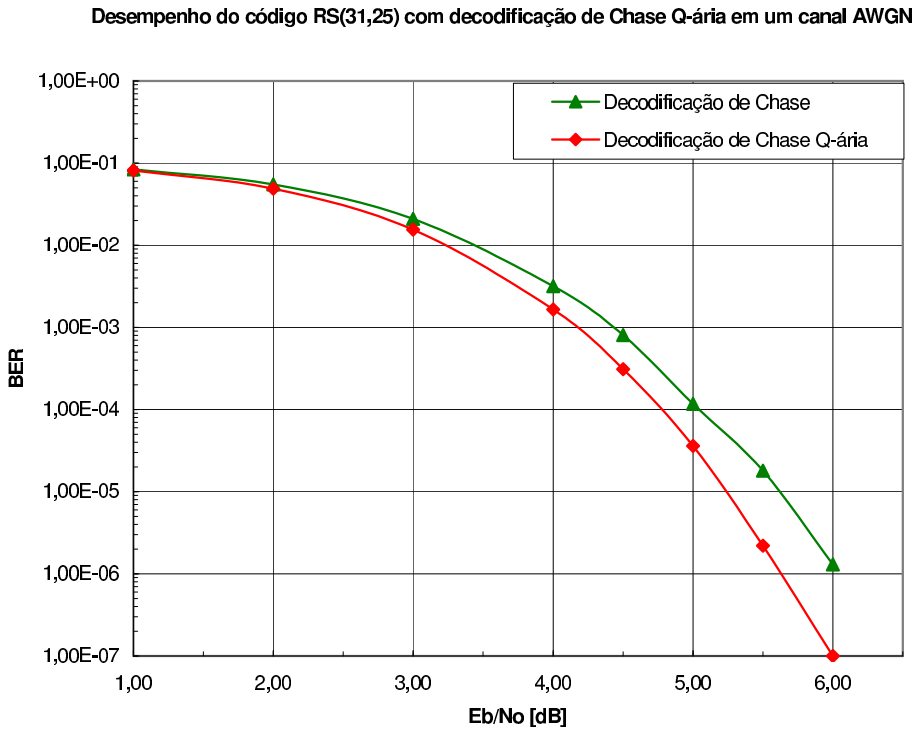

Fig. 3.10: Desempenho do código RS(31,25) utilizando decodificação suave de Chase-II e sua adaptação para símbolos q-ários.

sobre o método original de Chase II aumenta conforme o crescimento da relação E_b/N_o, observa-se na figura 3.11 uma melhoria de desempenho do método proposto sobre o método original para relações E_b/N_o mais baixas. Para uma relação $E_b/N_o = 3,75dB$ é observada uma $BER \approx 2 \times 10^{-5}$ com o uso da decodificação original de Chase II. Com a implementação do método proposto adaptado para símbolos q-ários, observa-se uma $BER \approx 1 \times 10^{-6}$ para a mesma relação Eb/No. Este ganho de desempenho representa uma melhoria de, aproximadamente, vinte vezes na taxa de erro de bit.

De acordo com os resultados apresentados nas figuras 3.9, 3.10 e 3.11, observamos que o método de decodificação adaptado para símbolos q-ários proposto na seção anterior supera em desempenho o método original de Chase para os três códigos RS apresentados, sem incrementar a complexidade da decodificação. Também podemos observar que a melhoria de desempenho torna-se mais significativa quanto maior for o alfabeto q-ário utilizado, neste caso, para os códigos do GF(32). Esta observação

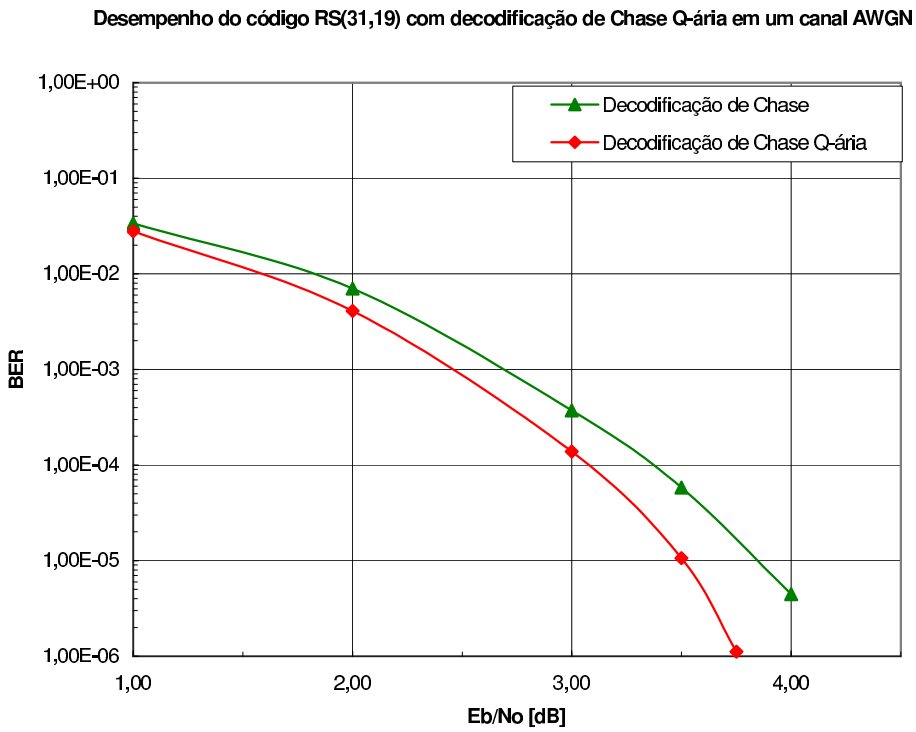

Fig. 3.11: Desempenho do código RS(31,19) utilizando decodificação suave de Chase-II e sua adaptação para símbolos q-ários.

é justificada pela queda de desempenho que o método original de decodificação de Chase sofre ao trabalhar com símbolos q-ários, que é proveniente das conversões símbolo-bit e bit-símbolo para adequar o método original à decodificação de palavras-código q-árias.

Capítulo 4

Sistemas FFH-CDMA

4.1 Sistemas com Espalhamento Espectral

Inicialmente, a aplicação das técnicas de espalhamento espectral dirigiu-se para o desenvolvimento de sistemas de guiamento e de comunicações militares. Estes sistemas de comunicações eram conhecidos como "Modulação de Ruído", pois os sinais espalhados espectralmente sobre uma grande largura de faixa pareciam mais um ruído para os receptores AM/FM convencionais. Após a Segunda Guerra Mundial, o conceito de espalhamento espectral já era familiar entre os engenheiros de radar e nos anos seguintes a investigação sobre o assunto foi motivada primordialmente pelo desejo de obter sistemas de comunicação altamente resistentes a interferência intencional (do inglês, *jamming*).

De acordo com [14], as técnicas de espalhamento espectral não se caracterizam apenas pelo fato da largura de faixa empregada ser muito maior que a mínima largura de faixa necessária para transmitir a informação. Um sistema é definido como sendo de espalhamento espectral se preencher os seguintes requisitos:

1. O sinal ocupa uma largura de banda muito maior que a mínima largura de faixa necessária para enviar a informação. Valores típicos para esta relação estão entre cem e um milhão;

2. O espalhamento é obtido a partir de um sinal de espalhamento, também conhecido como sinal código, que é independente da informação;

3. No receptor, a recuperação da informação original é feita pela correlação entre o sinal recebido e uma réplica sincronizada do sinal de espalhamento usado no processo de espalhamento no transmissor.

Esquemas de modulação em freqüência ou por código de pulso também espalham o espectro do sinal, mas não se qualificam como sistemas de espalhamento espectral por não satisfazerem as condições enunciadas acima.

O principal parâmetro em sistema de espalhamento espectral é o ganho de processamento, que é definido pela razão entre larguras de faixas de transmissão e de informação:

$$GP = \frac{W_t}{W_i}, \tag{4.1}$$

que representa basicamente o "fator de espalhamento". O ganho de processamento determina o número de usuários que podem ser alocados em um sistema, o grau de redução do efeito de multipercurso, o grau de imunidade à interferência e à interceptação no sinal etc. Quanto maior o ganho de processamento, maior será a robustez do sistema de espalhamento espectral.

Existem inúmeras razões para se espalhar o espectro e muitos benefícios podem ser obtidos se isto for realizado adequadamente. Abaixo, são descritas algumas vantagens da utilização da técnica de espalhamento espectral:

- **Rejeição à interferência intencional ("jamming")** pois as transmissões alheias na mesma faixa de freqüência do sinal de interesse, e descorrelacionadas deste, são fortemente atenuadas no receptor;

- **Rejeição à interferência natural** devido ao ganho de processamento inerente ao processo;

- **Baixa probabilidade de interceptação**, por dois aspectos: faixa mais larga a ser monitorada e densidade de potência do sinal a ser detectado reduzida pelo processo de espalhamento de espectro;

- Possibilidade de **comunicação multiusuário de acesso aleatório por endereçamento seletivo** que consiste na transmissão de sinais de vários

usuários simultaneamente no tempo, ocupando a mesma faixa de freqüência;

- **Resistência ao efeito do multipercurso** (quando o sinal transmitido percorre mais de um caminho até o receptor, devido a reflexões na superfície, prédios, árvores e atmosfera, produzindo interferências no sinal recebido). Por exemplo: supondo-se a resposta impulsiva do canal com múltiplo percurso invariante no tempo. Uma vez que o receptor intercepte o sinal recebido pelo caminho direto e obtenha a correlação deste com a seqüência de código previamente conhecida, os outros sinais que chegam por múltiplos percursos, ainda que possuam o mesmo código, estarão defasados no tempo e, conseqüentemente, descorrelacionados com o primeiro;

- **Medidas de distância com alta resolução** em sistemas de radar que utilizam técnicas de espalhamento espectral conseguem resolução melhor que os convencionais devido ao ganho de processamento inerente e à capacidade desses sistemas distinguirem pulsos muito próximos oriundos de reflexões no mesmo alvo, recebidos por múltiplos percursos;

- **Referência de tempo universal precisa** apesar de problemas de múltiplo percurso. A distinção de pulsos muito próximos recebidos por múltiplos percursos aliado ao ganho de processamento proporcionado pelas técnicas de espalhamento espectral, torna possível a um transmissor informar o mesmo referencial de tempo (com uma precisão tanto maior quanto mais elaborado for o sistema de espalhamento espectral), a receptores que conheçam o código utilizado no espalhamento.

De acordo com as vantagens citadas de utilização das técnicas de espalhamento espectral, vários sistemas de comunicações comerciais fazem uso das características inerentes do espalhamento espectral. Dentre as aplicações comerciais desta técnica podemos citar a mais conhecidas:

1. GPS (do inglês, *Global Positioning System*), onde um receptor utiliza sinais espalhados espectralmente transmitidos de alguns satélites de acordo com um formato pré-definido para determinar sua latitude, longitude e altitude com boa precisão.

2. WLAN (do inglês, *Wireless Local Area Network*), no qual os padrões IEEE 802.11 e 802.11b para redes sem fio utilizam técnicas de espalhamento espectral como padrão de suas camadas físicas.

3. Telefones sem fio residenciais e industriais, onde os fabricantes aplicam as técnicas de espalhamento espectral devido à segurança, imunidade ao ruído, maior alcance e densidade espectral de potência reduzida.

4. Sistema de telefonia celular CDMA (do inglês, *Code Division Multiple Access*), as técnicas de espalhamento espectral são úteis para as comunicações de acesso múltiplos onde vários usuários compartilham o mesmo meio de transmissão. A associação de uma seqüência de espalhamento espectral única para cada usuário permite o compartilhamento do canal com uma baixa interferência multi-usuário. O sistema CDMA será abordado com mais detalhes na seção 4.2, pois será utilizado para comparação de desempenho dos algoritmos de decodificação propostos nesta dissertação.

As principais técnicas de implementação do espalhamento espectral são definidas como: Seqüência Direta (do inglês, *DSSS - Direct Sequence Spread Spectrum*), Salto de Freqüência (do inglês, *FHSS - Frequency Hopping Spread Spectrum*) e Híbrida (onde há uma combinação entre a Seqüência Direta e o Salto de Freqüência). As duas primeiras técnicas serão detalhadas nas seções 4.1.2 e 4.1.3.

4.1.1 Modelo de um Sistema de Comunicação com Espalhamento Espectral

Na figura 4.1, é apresentado o diagrama em blocos de um sistema de espalhamento espectral. Observe na figura que o modelo apresenta dois elementos inerentes a um sistema com espalhamento espectral, que são o espalhador e o desespalhador. Estes elementos consistem de dois geradores idênticos de seqüência aleatória que se conectam com o modulador na transmissão e com o demodulador na recepção. Desta maneira, faz-se necessário a sincronização dos geradores de seqüência aleatória para a correta demodulação do sinal recebido.

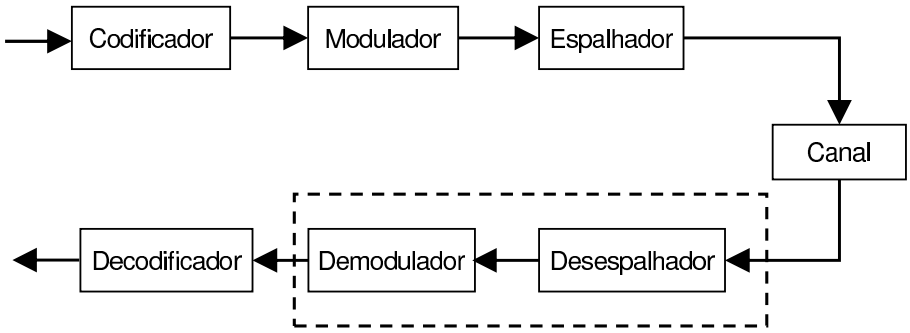

Fig. 4.1: Diagrama em blocos de um sistema com espalhamento espectral.

Os tipos de modulação utilizados em sistemas de espalhamento espectral são: PSK e FSK. A determinação da modulação a ser utilizada dependerá do tipo de aplicação da técnica de espalhamento espectral. Caso seja utilizado o espalhamento espectral por seqüência direta, a modulação PSK é utilizada em conjunto com as seqüências pseudo-aleatórias. Quando na aplicação a coerência de fase for temporalmente comprometida no canal de comunicação, a modulação FSK é recomendada. Neste caso, a seqüência pseudo-aleatória será responsável por definir a freqüência a ser transmitida, o que caracteriza o espalhamento espectral por salto em freqüência.

4.1.2 Espalhamento Espectral por Seqüência Direta

Conforme ilustrado na figura 4.2, o espalhamento espectral por seqüência direta é realizado modulando a portadora com uma seqüência de código que possui uma taxa

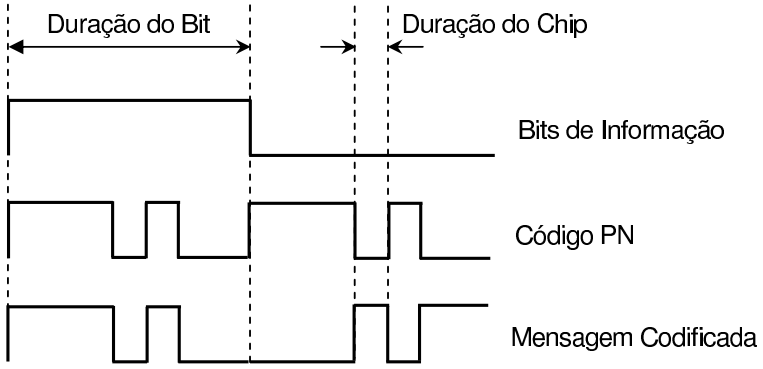

Fig. 4.2: Espalhamento espectral por seqüência direta.

de bit muito maior que a taxa da mensagem a ser enviada. Esta seqüência de código é normalmente um código binário pseudo-aleatório, chamado de pseudo-ruído (do inglês, *Pseudo-Noise* - PN), especialmente escolhido por propriedades estatísticas desejáveis. Na verdade, é transmitido um ruído faixa larga como um sinal que contém mensagem de informação. A duração de um bit do código pseudo-aleatório é chamada de *chip*.

No receptor, o sinal recebido é multiplicado pelo mesmo código pseudo-aleatório utilizado pelo transmissor correspondente. Esta operação extrai o código do sinal recebido e recupera a seqüência de informação que foi codificada para transmissão.

O principal problema do espalhamento espectral por seqüência direta é o efeito *Near-Far*. Este efeito é observado quando um transmissor interferente está muito mais próximo do receptor que o transmissor pretendido. Embora a correlação cruzada entre os códigos pseudo-aleatórios do transmissor pretendido e do transmissor interferente seja baixa, o código pseudo-aleatório do receptor pode obter uma correlação maior com o sinal recebido do transmissor interferente do que com o sinal recebido do transmissor pretendido devido à potência dos sinais recebidos. O resultado será uma detecção não possível.

4.1.3 Espalhamento Espectral por Saltos em Freqüência

No espalhamento espectral por salto em freqüência, a portadora a ser transmitida é chaveada periodicamente entre diferentes *slots* de freqüência de largura de faixa igual a B Hz, que é uma pequena fração comparada com a largura de faixa total W_t Hz utilizada pelo sistema para o espalhamento espectral. Estes saltos periódicos da portadora entre os *slots* de freqüências disponíveis caracteriza o espalhamento espectral. A determinação de quais *slots* de freqüência serão utilizados para a transmissão é feita conforme a seqüência pseudo-aleatória (código PN).

Na figura 4.3, é apresentado o diagrama em blocos do transmissor de um sistema com espalhamento espectral por saltos em freqüência. Observe que um sintetizador de freqüência é utilizado para realizar os saltos entre os slots de freqüências, sendo comandado por um gerador de códigos PN. A taxa da seqüência pseudo-aleatória definirá a taxa de saltos em freqüência, e quando aquela for muito maior que a taxa

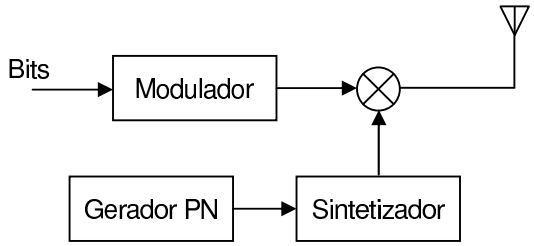

Fig. 4.3: Modelo do transmissor para espalhamento espectral por salto em freqüência.

de bits de informação, ou seja, $B \ll W$, tem-se o que é chamado de salto rápido em freqüência (do inglês, *Fast Frequency Hopping* - FFH). Entretanto, quando a taxa da seqüência pseudo-aleatória for menor do que a taxa de bits de informação, tem-se um sistema com salto lento em freqüência (do inglês, *Slow Frequency Hopping* - SFH). Quanto maior for a taxa da seqüência pseudo-aleatória, maior será o ganho de processamento.

Nos estudos desta dissertação, um sistema de espalhamento espectral com salto rápido em freqüência, conhecido como FFH-CDMA, é utilizado como canal, visto no sentido do codificador/decodificador, para as simulações. Maiores detalhes deste sistema são apresentados na seção 4.2.

4.2 Canal FFH-CDMA

4.2.1 Descrição do Sistema FFH-CDMA

O sistema de comunicação CDMA (do inglês, *Code Division Multiple Access*) utiliza as técnicas de espalhamento espectral para mitigar algumas deficiências encontradas em outros sistemas de telefonia celular. Dentre as vantagens do CDMA, podemos citar: maior robustez à interferência e ao multipercurso, baixa densidade espectral de potência, maior imunidade a interferência intencional e maior capacidade de usuários. A utilização do salto rápido em freqüência aumenta o ganho de processamento do sistema.

Considere um sistema FFH-CDMA codificado descrito em [15], onde uma quantidade U de usuários se comunica com a estação rádio base (ERB) a uma taxa R_b bits/s, compartilhando o mesmo canal de largura de faixa igual a W_{ss} Hz. A modu-

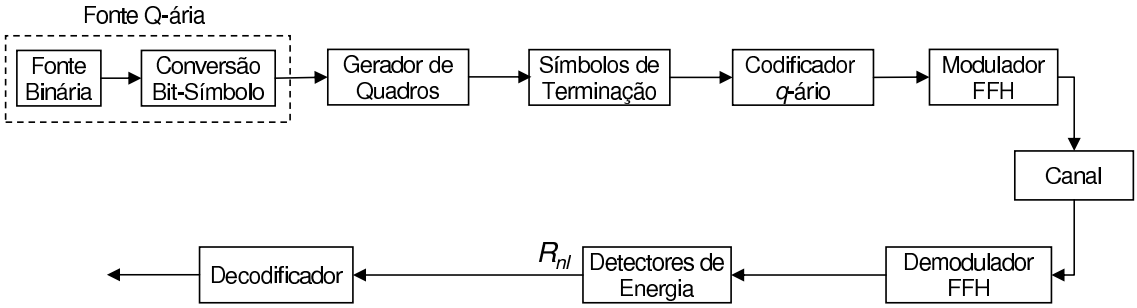

Fig. 4.4: Sistema de comunicação FFH-CDMA codificado.

lação utilizada na comunicação é o M-FSK. Cada usuário do sistema é identificado exclusivamente por um endereço local de assinante denominado padrão de salto em freqüência, que define como será realizado o espalhamento espectral da informação de cada usuário sobre toda a largura de faixa W_{ss}. O número de ramos de diversidade, ou mais conhecidamente, o número de *chips* L, é definido como o número de freqüências por padrão de salto. O intervalo da modulação M-FSK é T e a duração do *chip* obedece $T_c = T/L$, com todos os chips alinhados no tempo. O número de canais de freqüências é dado por $N = W_{ss}T_c$ e é igual ao número M de freqüências da modulação M-FSK, isto é, $N = M$.

Na figura 4.4, é apresentado o diagrama em blocos do sistema FFH-CDMA codificado proposto em [15], porém, com algumas adaptações para suportar o codificador q-ário, tais como: remoção do conversor bit-símbolo após o codificador, posicionando-o depois da fonte binária, e o uso do bloco de símbolos de terminação ao invés de bits de terminação. Seus componentes básicos são: (a) fonte q-ária, composta da fonte binária e do conversor bit-símbolo; (b) gerador de quadros; (c) bloco de símbolos de terminação; (d) codificador q-ário; (e) modulador FFH (modulação M-FSK, L *chips*); (f) canal com desvanecimento de Rayleigh seletivo em freqüência (M canais independentes) com ruído térmico do receptor e interferência multi-usuário; (g) demodulador FFH; (h) M detectores de energia não-coerentes; (i) decodificador.

O gerador de quadros recebe os símbolos de informação enviados pela fonte de dados, particionando-os em quadros de n_p símbolos. Em seguida, são adicionados n_{tail} símbolos de terminação para adequar a seqüência ao código q-ários utilizado.

Desta maneira, na saída do codificador teremos $n_{cod} = (n_p + n_{tail})/R_{CO}$ símbolos codificados por quadro, onde R_{CO} é a taxa do código. Entretanto, a taxa efetiva do código, $R_{CO}^{(eff)}$, será dada por:

$$R_{CO}^{(eff)} = \frac{n_p}{n_{cod}} = \frac{n_p}{(n_p + n_{tail})/R_{CO}}. \tag{4.2}$$

Após o particionamento e o preenchimento de símbolos de terminação, os símbolos são enviados ao codificador q-ário que irá gerar as palavras-códigos. Em seguida, estes símbolos serão enviados ao modulador e modulados através da técnica M-FSK. Para um mapeamento casado com o número de freqüências do modulador, o codificador deverá trabalhar com um alfabeto de dimensão $q = M$, assim, para todo símbolo codificado, teremos um símbolo correspondente na constelação M-FSK.

O modulador FFH recebe um símbolo M-ário $m \in 0, 1, ..., M-1$ e gera um sinal formado pela combinação de ML formas de onda de uma base:

$$s_m(t) = \sum_{n=0}^{M-1} \sum_{l=0}^{L-1} c_{nl} x_{nl}(t), \tag{4.3}$$

onde $c_{nl} = 1$ para $n = m$ e $c_{nl} = 0$ para $n \neq m$. A forma de onda x_{nl} usada no transmissor é dada por:

$$x_{nl}(t) = \sqrt{2S}\ rect_{T_c}(t - lT_c) \cos(2\pi[f_0 + \frac{n}{T_c}]t), \tag{4.4}$$

onde f_0 é um múltiplo de $\frac{1}{T_c}$ e $rect_{T_c}(.)$ é um pulso retangular de amplitude unitária e largura T_c. A energia por chip E_c está relacionada com a energia por bit E_b através da equação:

$$E_c = E_b\ R_{CO}^{eff}\ \frac{K}{L}. \tag{4.5}$$

O bloco do modulador FFH da figura 4.4 ainda possui a função do transmissor ilustrado na figura 4.3, ou seja, ele modifica a freqüência em cada chip de $s_m(t)$ antes da transmissão, de acordo com o padrão de salto em freqüência de cada usuário. O modelo a ser considerado é de padrões de salto em freqüência aleatórios.

O canal FFH se refere ao canal visto entre o codificador e o decodificador, que

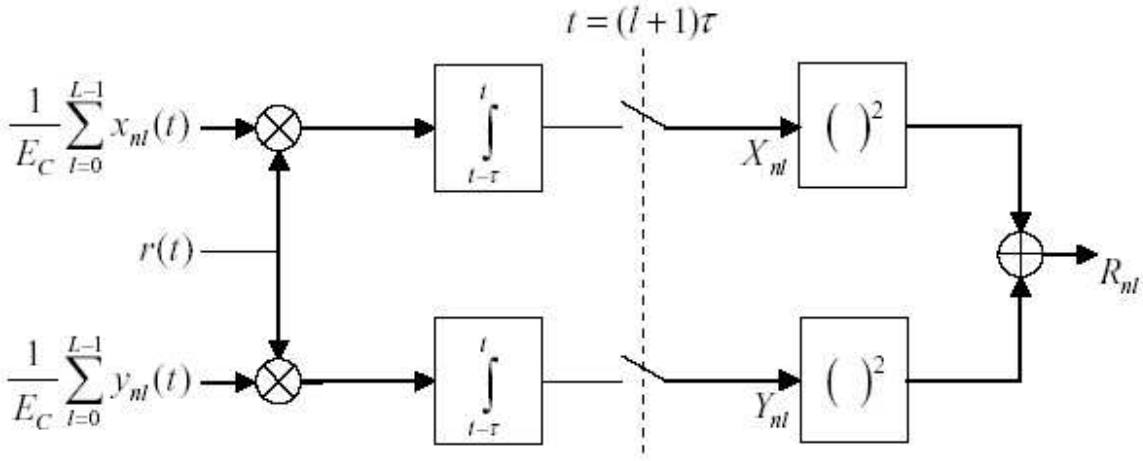

Fig. 4.5: *nl*-ésimo Detector de Energia.

compreende o modulador FFH, o canal com desvanecimento de Rayleigh seletivo em freqüência com ruído térmico no receptor e interferência multi-usuário, o demodulador FFH e, finalmente, os M detectores de energia não-coerentes.

No receptor, cada detector de energia é formado por correlatores cujas saídas são amostradas, elevadas ao quadrado e somadas, conforme a figura 4.5. Deste modo, a nl-ésima saída não-quantizada do detector de energia é dada por:

$$R_{nl} = X_{nl}^2 + Y_{nl}^2, \tag{4.6}$$

onde X_{nl} e Y_{nl} são as componentes em fase e em quadratura do sinal recebido, conforme [16].

O canal FFH-CDMA descrito até aqui também pode ser modelado por ML funções de densidade de probabilidade, $p(R_{nl}|m)$, com $0 \leq n \leq M-1$ e $0 \leq l \leq L-1$, para cada símbolo M-ário de entrada do modulador FFH [16]. Cada função $p(R_{nl}|m)$ é dada por:

$$p(R_{nl}|m) = \sum_{k=0}^{J} \frac{B(k;J,\mu)}{k+d+\delta_{nm}} \exp\left(-\frac{R_{nl}}{k+d+\delta_{nm}}\right), \tag{4.7}$$

onde $\mu = 1/M$, δ_{nm} é o delta de Kronecker, J é o número de usuários interferentes, $B(k;J,\mu) = \begin{pmatrix} J \\ k \end{pmatrix} \mu^k (1-\mu)^{J-k}$ e d é definido como o inverso da relação sinal-ruído

por chip, dado por:

$$d = \frac{N_0}{E_c} = \frac{L}{KR_{CO}^{eff}} \left(\frac{E_b}{N_0} \right)^{-1}, \tag{4.8}$$

onde E_b é a energia média por bit de informação, R_{CO}^{eff} é a taxa efetiva do código, K é o número de bits por símbolo da modulação M-FSK e N_0 é a densidade espectral de potência unilateral do ruído.

Considere o canal visto pelo decodificador, seja $\mathbf{m} = (m_0, m_1, ..., m_i, ..., m_{n_{sym}-1})$ uma seqüência de símbolos de comprimento n_{sym}, que corresponde a uma seqüência de saída do decodificador, e $\mathbf{R} = (\mathbf{R}^0, \mathbf{R}^1, ..., \mathbf{R}^i, ..., \mathbf{R}^{n_{sym}-1})$ uma seqüência recebida. Pelo fato de as saídas não-quantizadas dos detectores de energia R_{nl} serem estaticamente independentes, a densidade de probabilidade conjunta $p(\mathbf{R}|m)$ pode ser escrita como:

$$p(\mathbf{R}|m) = \prod_{i=0}^{n_{sym}-1} \prod_{n=0}^{N-1} \prod_{l=0}^{L-1} p(R_{nl}^i|m_i), \tag{4.9}$$

onde R_{nl}^i é a saída da i-ésima matriz recebida, conforme [15].

O algoritmo de decodificação de Chase q-ário proposto no capítulo 3 será aplicado ao canal FFH-CDMA. O resultado utilizado para verificar o desempenho de um esquema de codificação/decodificação em um canal FFH-CDMA é um gráfico que contém a informação da probabilidade de erro de bit do sistema em função de sua eficiência espectral η, que é definida como:

$$\eta = U \frac{K}{LM} R_{CO}^{eff}, \tag{4.10}$$

onde $U = J + 1$ é o número total de usuários no canal.

Um outro critério de desempenho passível de avaliação é a verificação de quanto o esquema de codificação/decodificação aproxima-se, ou se sobressai, da taxa de corte do canal. A seguir, será descrita a taxa de corte do canal FFH-CDMA.

4.2.2 Taxa de Corte do Canal FFH-CDMA Não-Quantizado

Quando um código não-binário é utilizado em conjunto com uma sinalização M-ária ($M = 2^K$), o sinal recebido na saída dos M filtros casados do receptor de máxima verossimilhança pode ser representado por um vetor de dimensão ML dado por:

$$\begin{aligned}\mathbf{R} = [&R_{0,0}, R_{0,1}, ..., R_{0,i}, ..., R_{0,L}, R_{1,0}, R_{1,1}, ..., R_{1,i}, ..., R_{1,L}, ..., R_{j,0}, R_{j,1}, ..., \\ &R_{j,i}, ..., R_{j,L}, ..., R_{M-1,0}, R_{M-1,1}, ..., R_{M-1,i}, ..., R_{M-1,L}].\end{aligned} \tag{4.11}$$

A taxa de corte para este canal de M entradas e ML saídas é dada em [17] por:

$$R_0^{MV} = \max_{p_j} \left\{ -\log_2 \sum_{j=0}^{M-1} \sum_{i=0}^{M-1} p_j p_i \int_{\mathbf{R}} \sqrt{p(\mathbf{R}|j)p(\mathbf{R}|i)} d\mathbf{R} \right\}, \tag{4.12}$$

onde $p(\mathbf{R}|j)$ é a função densidade de probabilidade condicional da matriz de saída $\mathbf{R}$ do demodulador dado que o j-ésimo sinal foi transmitido.

A taxa de corte de canal para um receptor de MV que utiliza um canal FFH-CDMA não-quantizado é dada em [18] por:

$$R_0^{MV} = K - \log_2 \left\{ 1 + (2^K - 1) \left[\int_0^{\infty} \sqrt{p_{i,j|i=j} p_{i,j|i\neq j}}\, dR_{nl} \right]^{2L} \right\}, \tag{4.13}$$

onde $p_{i,j|i=j}$ é a função densidade de probabilidade dada por (4.7) com $\delta_{nm} = 1$ e $p_{i,j|i\neq j}$ com $\delta_{nm} = 0$. A figura 4.6 ilustra a taxa de corte de canal R_0^{MV} em função do número de usuários simultâneos U para dois sistemas distintos, calculada por (4.13). Através desta curva, podemos constatar que o valor de R_0 diminui à medida que U cresce.

Conforme mencionado no capítulo 2, a taxa de corte do canal fornece um limite para uma transmissão confiável da informação. É interessante identificarmos onde a taxa de corte se localiza na curva $P_b \times \eta$ de um sistema FFH-CDMA. Para isso, obte-

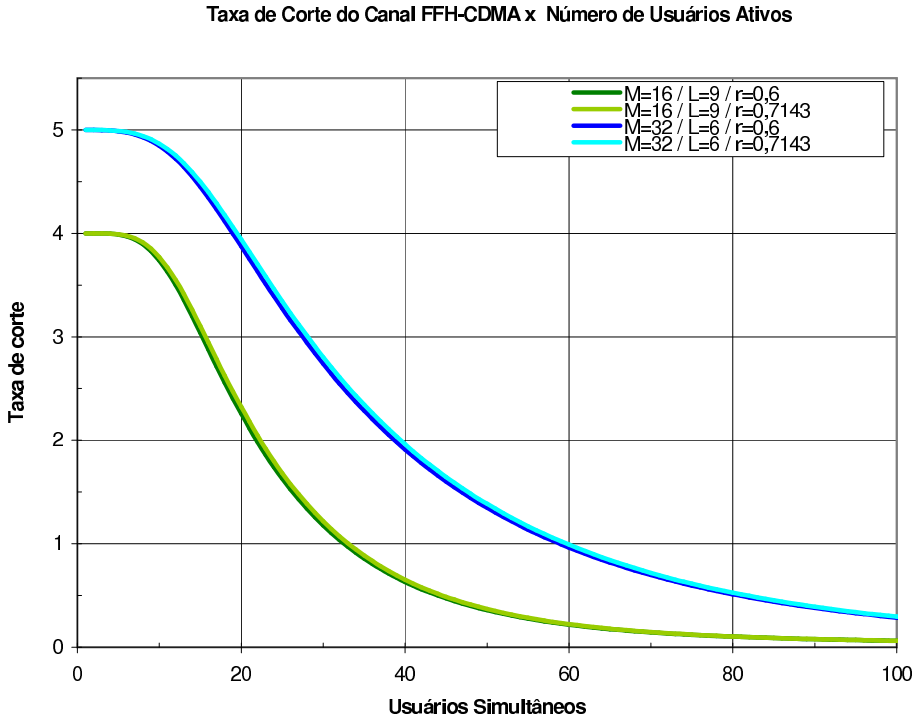

Fig. 4.6: Taxa de Corte do Canal R_0^{MV} versus número de usuários U para dois sistemas FFH-CDMA com taxas de codificação diferentes.

mos os valores de eficiência máxima, substituindo em (4.10) o número de usuários simultâneos, U, correspondente ao valor de $R_0^{max} = (\log_2 M)r$, onde r é a taxa do código. Entretanto, para valores pequenos na probabilidade de erro de bit (P_b), os valores de eficiência máxima devem ser ajustados através do cálculo de uma nova taxa de corte, R_0'. Em [19], é apresentada uma aproximação para a nova taxa do código, r', que é dada por:

$$r' \leq r\left[1 + P_b \log_2(P_b) + (1 - P_b)\log_2(1 - P_b)\right]. \tag{4.14}$$

Para um código de taxa r, calculamos r' através de (4.14), com $R_0' = Kr'$, onde K é o número de bits por símbolo da modulação M-FSK. Desta forma, a eficiência máxima para a taxa de corte de canal do receptor de MV são calculados substituindo os novos valores encontrados para o número de usuários ativos, U', em (4.10). Os valores de U' são obtidos através de interpolação linear utilizando os valores da taxa de corte de canal, R_0^{MV}, em função do número de usuários, U, obtidos da figura 4.6.

4.3 Decodificação q-ária Suave em Sistemas FFH-CDMA

4.3.1 Algoritmo de Chase q-ário

Nesta seção, o método proposto de decodificação q-ária suave, baseado no algoritmo de Chase e apresentado no Capítulo 3 para o canal gaussiano, é aplicado ao sistema FFH-CDMA.

Conforme a seção 3.3.1, para gerar os padrões de teste q-ários é necessário o conhecimento da palavra recebida $\mathbf{Y}$ e da seqüência $\mathbf{Y}^a$ apresentada naquela seção. Para o sistema FFH-CDMA, o símbolo escolhido $Y_i = m$ obedece à desigualdade abaixo:

$$p(\mathbf{R}|Y_i = m) > p(\mathbf{R}|Y_i = j), \quad \textit{para todo} \;\; j \neq m. \tag{4.15}$$

Observe que, aplicando o teorema de Bayes e considerando $P(Y_i) = cte$, a equação (4.15) pode ser reescrita da seguinte forma:

$$P(Y_i = m|\mathbf{R}) > P(Y_i = j|\mathbf{R}), \quad \textit{para todo} \;\; j \neq m. \tag{4.16}$$

Observe que a equação (4.16) é semelhante à equação (3.20). Analogamente, o símbolo $Y_i^a = k$ da seqüência $\mathbf{Y}^a$ é obtido através da desigualdade abaixo:

$$p(\mathbf{R}|Y_i^a = m) > p(\mathbf{R}|Y_i^a = k) > p(\mathbf{R}|Y_i^a = j), \quad \textit{para todo} \;\; j \neq m \neq k. \tag{4.17}$$

De posse da palavra recebida $\mathbf{Y}$ e da seqüência $\mathbf{Y}^a$ para o canal FFH-CDMA, podemos gerar os padrões de teste q-ários conforme o procedimento descrito na seção 3.3.1, obedecendo à equação (3.22) e fazendo uso da equação (4.9).

Um outro passo importante para a aplicação do método proposto de decodificação q-ária suave é a obtenção da confiabilidade dos símbolos da palavra recebida $\mathbf{Y}$. Para o sistema FFH-CDMA, a confiabilidade é gerada de forma análoga ao procedimento descrito na seção 3.3.2 para o canal gaussiano.

No canal FFH-CDMA, o Logaritmo da Razão de Verossimilhança, LLR, pode ser obtido a partir da probabilidade $P(Y_i|\mathbf{R})$, ou seja, a probabilidade do símbolo Y_i ser recebido corretamente dado que $\mathbf{R}$ foi observado na saída dos detectores de energia. Obtendo estes valores de probabilidades para todos os símbolos do alfabeto utilizado, podemos aplicar a equação (3.23) para gerar o LLR. Porém, podemos usufruir das mesmas condições utilizadas na seção 3.3.2 para resultar na equação (3.25) a fim de utilizar a probabilidade $p(\mathbf{R}|Y_i)$ da equação (4.9) já conhecida. Desta maneira, o valor do logaritmo da razão de verossimilhança para o canal FFH-CDMA pode ser encontrado através da seguinte equação:

$$L(Y_i|\mathbf{R}) = \ln\left(\frac{p(\mathbf{R}|Y_i^{(1)})}{\sum_{j=2}^{q} p(\mathbf{R}|Y_i^{(j)})}\right). \tag{4.18}$$

Observe que, igualmente ao canal gaussiano, o valor $L(Y_i|\mathbf{R})$ pode ser utilizado como uma medida de confiabilidade do símbolo recebido Y_i, pois assumirá maiores valores quanto maior for o valor do numerador em relação ao denominador, bem como assumirá menores valores quanto mais próximo, ou maior, for o valor do denominador em relação ao numerador.

4.3.2 Resultados

Nesta seção, são apresentados os resultados das simulações realizadas para avaliar o desempenho do método proposto de decodificação q-ária suave baseado no algoritmo de Chase. Os procedimentos descritos na seção 4.3.1 foram implementados para a obtenção dos resultados. Como a melhoria de desempenho do método de decodificação q-ária proposto, em relação ao método original do algoritmo de Chase II, foi demonstrada na seção 3.4, apenas serão enfatizados os resultados do método proposto.

Basicamente, as simulações foram baseadas no diagrama em blocos da figura 4.4. A ordem de grandeza da quantidade de bits gerados pela fonte de informação foi de 10^7, a quantidade exata dependeu dos parâmetros do código corretor de erro utilizado. Desta maneira, taxas de erro de bit com ordem de grandeza de 10^{-6} puderam ser obtidas. Foram utilizados os códigos RS(15,9) e RS(28,20).

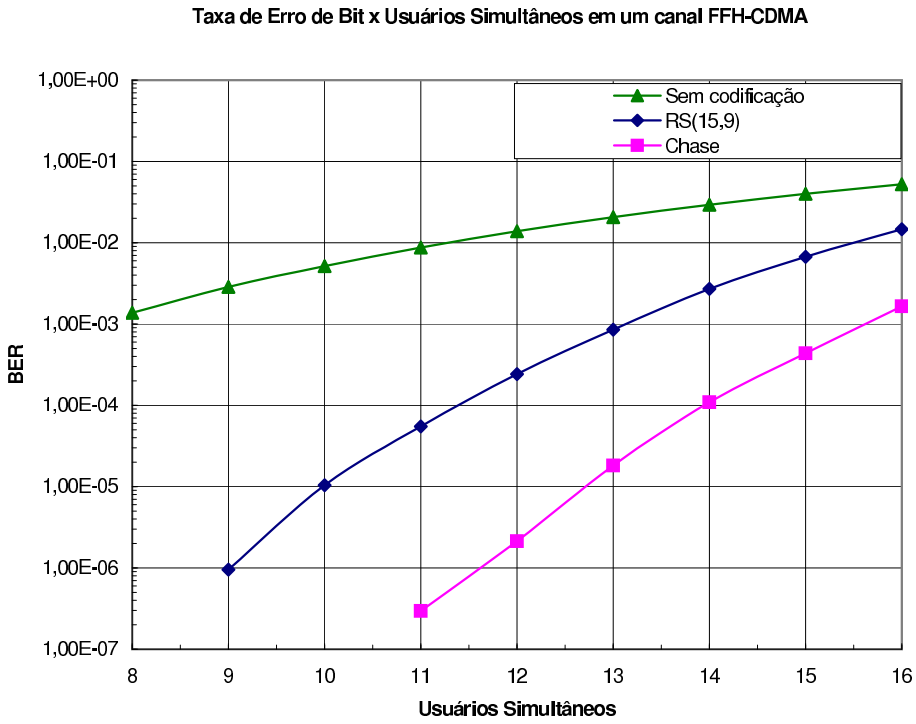

Fig. 4.7: Desempenho do código RS(15,9) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$.

Com a aplicação do método proposto de decodificação q-ária suave, o bloco de detectores de energia acumula a função de fornecer ao decodificador o logaritmo da razão de verossimilhança $L(Y_i|\mathbf{R})$, ou seja, a confiabilidade dos símbolos detectados, bem como os valores de $p(\mathbf{R}|Y_i)$, $\forall\ Y_i \in GF(q)$. De posse destes valores, o decodificador executa os procedimentos para a geração dos padrões de teste $\mathbf{T}$ e decodifica a palavra recebida $\mathbf{Y}$, conforme o método proposto descrito anteriormente.

Na figura 4.7, são apresentados os resultados da taxa de erro de bit versus o número de usuários simultâneos em um sistema FFH-CDMA com a aplicação da decodificação q-ária suave proposta. Na simulação, foi utilizado o código q-ário RS(15,9) e uma relação $E_b/N_o = 25dB$. Podemos observar claramente a melhoria de desempenho do sistema FFH-CDMA com a aplicação do método proposto de decodificação q-ária. Para $U = 11$ usuários simultâneos utilizando o canal, podemos observar uma taxa de erro de bit de, aproximadamente, 5×10^{-5} para o código RS(15,9) com decodificação algébrica. Enquanto que $U = 11$ usuários simultâneos

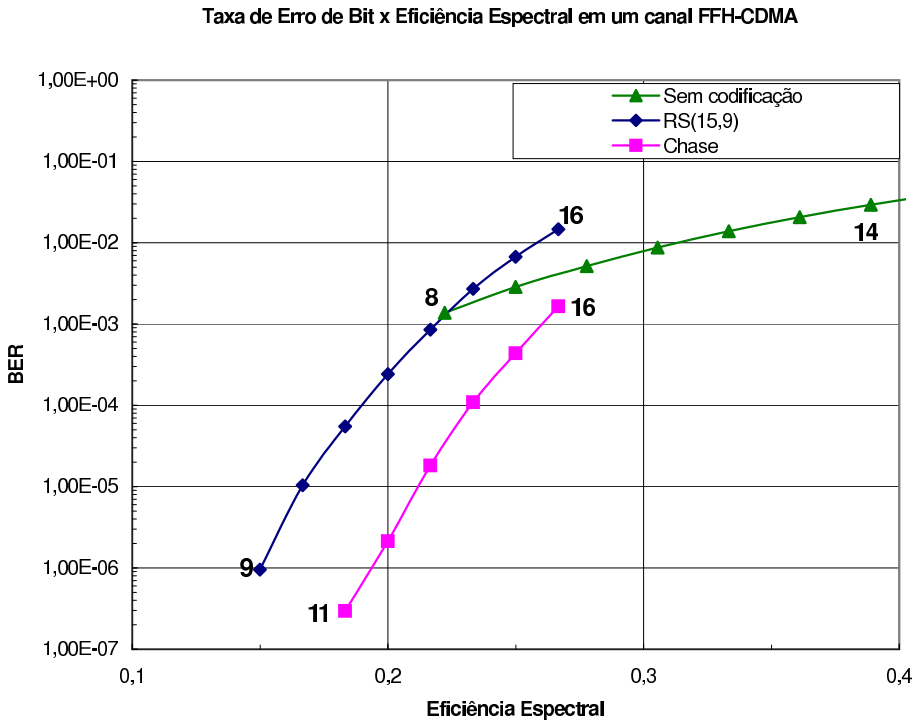

Fig. 4.8: Desempenho do código RS(15,9) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$.

em um sistema FFH-CDMA que utiliza o método proposto de decodificação q-ária suave, gera uma taxa de erro de bit de, aproximadamente, 3×10^{-7}. Observe que, conforme esperado, a medida que o número de usuários simultâneos é incrementada, menor será a melhoria de desempenho do método proposto de decodificação q-ária suave, pois maior será a interferência entre usuários. Uma outra maneira de avaliar a melhoria de desempenho do sistema FFH-CDMA é apresentada na figura 4.8, onde a taxa de erro de bit é informada de acordo com a eficiência espectral do sistema FFH-CDMA, os números apresentados nas curvas representam a quantidade de usuários simultâneos do sistema.

Na figura 4.9, é apresentado o desempenho do código RS(28,20) com a decodificação q-ária suave em um sistema FFH-CDMA para uma $E_b/N_o = 25dB$. Semelhantemente aos resultados mostrados para o código RS(15,9), a melhoria de desempenho obtida pela utilização do método proposto é claramente observada. Para $U = 14$ usuários simultâneos utilizando o canal, podemos observar uma taxa

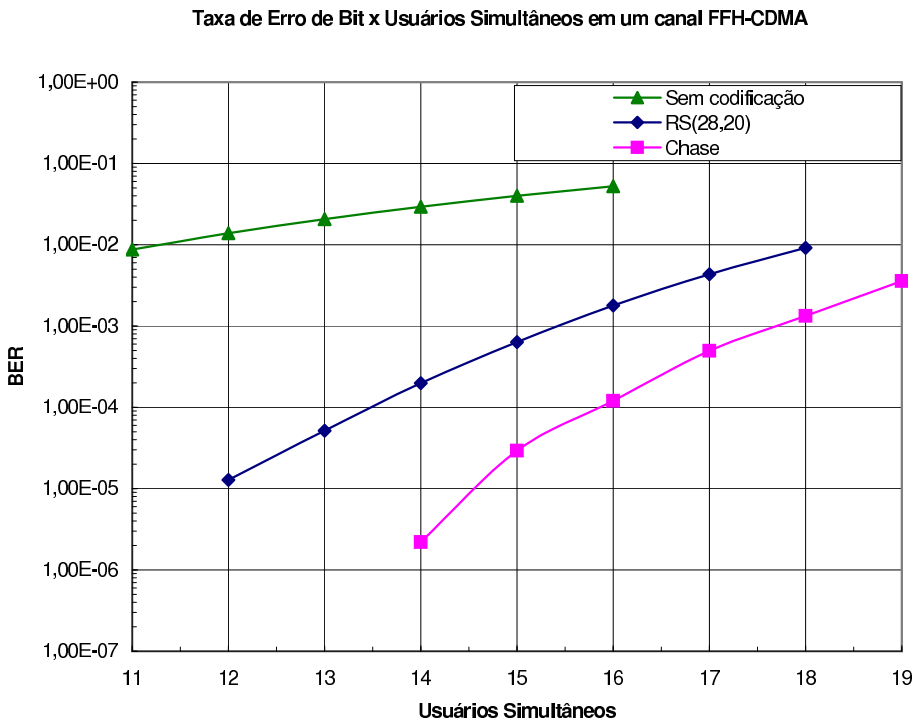

Fig. 4.9: Desempenho do código RS(28,20) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$.

de erro de bit de, aproximadamente, 2×10^{-4} com o uso da decodificação algébrica. Enquanto que $U = 14$ usuários simultâneos em um sistema FFH-CDMA que utiliza o método proposto de decodificação q-ária suave, gera uma taxa de erro de bit de, aproximadamente, 2×10^{-6}. Comparando com os resultados da decodificação por apagamento e erro apresentados em [20], temos o incremento de 3 usuários na quantidade de usuários simultâneos, quando utilizamos a decodificação q-ária. Observe que a melhoria de desempenho com a utilização do método proposto, em relação ao método algébrico, é maior para a codificação RS(28,20) do que para a codificação RS(15,9). Uma outra maneira de avaliar a melhoria de desempenho do código RS(28,20) em um sistema FFH-CDMA é apresentada na figura 4.10, onde a taxa de erro de bit é informada de acordo com a eficiência espectral do sistema FFH-CDMA, os números apresentados nas curvas representam a quantidade de usuários simultâneos do sistema. Comparando novamente com os resultados de [20], na decodificação por apagamento e erro temos uma eficiência espectral $\eta \approx 0,22$, para uma

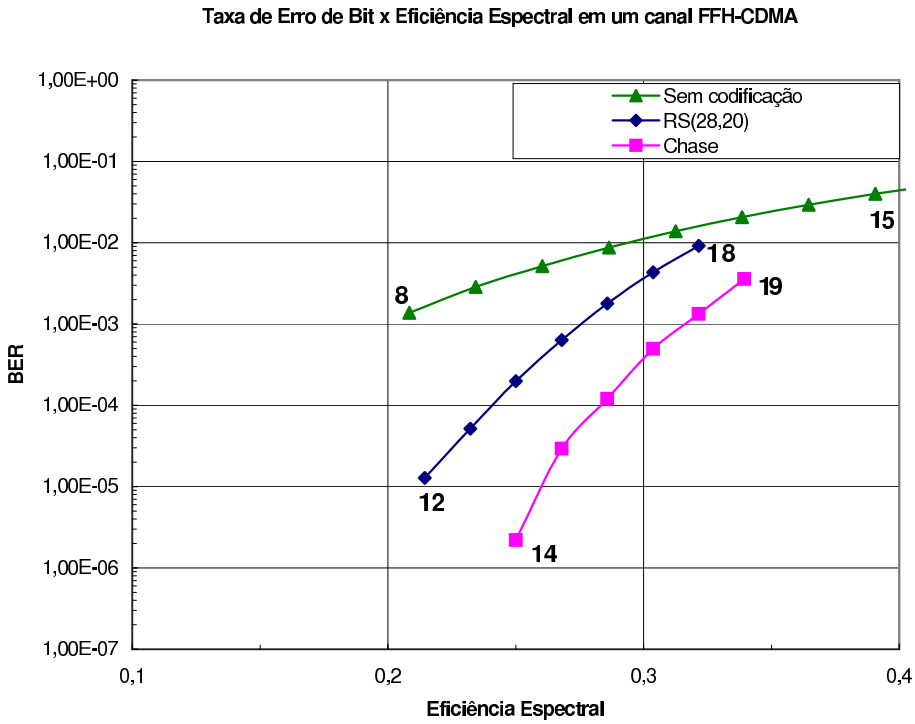

Fig. 4.10: Desempenho do código RS(28,20) utilizando decodificação q-ária suave em um sistema FFH-CDMA com $E_b/N_o = 25dB$.

$BER = 10^{-4}$, já o decodificador q-ário resulta em uma eficiência espectral $\eta \approx 0,29$.

Conforme observado para o canal gaussiano na seção 3.4, através dos resultados apresentados nesta seção, podemos afirmar que a utilização do método proposto de decodificação q-ária suave gera uma considerável melhoria de desempenho do sistema FFH-CDMA. Também podemos mencionar que o ganho de desempenho, em relação ao método algébrico, é mais significante para o código com maior alfabeto q-ário, neste caso, para o código do GF(32).

Capítulo 5

Decodificação Turbo q-ária

5.1 Introdução

No capítulo 3, foi apresentado o método proposto de decodificação q-ária suave e seu desempenho em um canal AWGN. No capítulo 4, foram apresentadas as adaptações necessárias para a aplicação do método proposto de decodificação q-ária suave em um canal FFH-CDMA, bem como seu desempenho. Neste capítulo, serão detalhadas as propostas de adaptação do método de decodificação q-ária suave para suportar os conceitos de decodificação turbo, bem como sua aplicação e desempenho em um sistema FFH-CDMA.

Primeiramente, serão apresentados os conceitos de códigos turbo, que compreende tanto a aplicação de códigos produtos quanto a utilização de decodificação iterativa para recuperar a palavra enviada.

Sabemos que um sistema de comunicação, entre outros parâmetros, torna-se mais robusto quanto mais poderosas forem as técnicas de codificação e decodificação empregadas. Todavia, a utilização de algoritmos de decodificação mais poderosos implica no aumento da complexidade do sistema utilizado, tornando-o economicamente menos viável ou até inviável. Dado estas observações, o projetista deve desenvolver seu sistema de comunicação levando em consideração o compromisso entre complexidade e desempenho.

Anteriormente, a complexidade dos códigos mais poderosos estava relacionada com o tamanho do código (códigos longos) e o algoritmo de decodificação necessário para recuperá-lo. Assim, para mitigar a complexidade dos poderosos algoritmos de codificação e decodificação, surgiram os códigos concatenados. Neste esquema de codificação, utiliza-se uma combinação de códigos mais curtos de maneira que cada um deles possa ser decodificado através de decodificadores mais simples, o que acarreta na diminuição da complexidade do sistema. Através do uso de decodificadores de decisão suave SISO (do inglês, *Soft-In-Soft-Out*), a informação pode ser passada de um decodificador a outro, caracterizando um processo iterativo. Desta maneira, o conceito de código turbo pode ser definido como um estágio de codificação concatenada e um algoritmo de decodificação iterativa casado com os códigos utilizados.

A primeira experiência realizada com o intuito de se alcançar a alta capacidade de correção de erros de códigos longos, sem sua complexidade inerente, foi através de códigos produto. Porém, a decodificação dos códigos componentes não resultou em melhorias de desempenho, como se esperava, devido ao uso de algoritmos de decodificação de decisão abrupta HIHO (do inglês, *Hard-In-Hard-Out*), conforme mencionado em [2]. Em 1993, Berrou *et al.* [1] mostraram que era possível transmitir dados com uma taxa acima da taxa de corte do canal. Ele obteve uma taxa de erro de bit extremamente baixa com uma relação sinal-ruído por bit de informação (E_b/N_o) próxima do limite de Shannon em um canal gaussiano, utilizando códigos turbo convolucionais - CTC (do inglês, *Convolutional Turbo Codes*) [1] [21]. A partir de então, os pesquisadores passaram a se dedicar mais aos códigos turbo convolucionais, deixando os códigos turbo de bloco de lado.

Em 1994, Pyndiah *et al.* propuseram um novo decodificador SISO [22] para todos os códigos de bloco lineares e mostraram que os códigos turbo de bloco (BTC, do inglês, *Block Turbo Codes*) possuem desempenho comparável com os códigos turbo convolucionais. Este novo código turbo de bloco oferece um bom compromisso entre desempenho e complexidade, sendo portanto, muito atrativo para implementação. Neste capítulo, os BTCs são discutidos e aplicados no algoritmo proposto de decodificação q-ária suave em um canal FFH-CDMA.

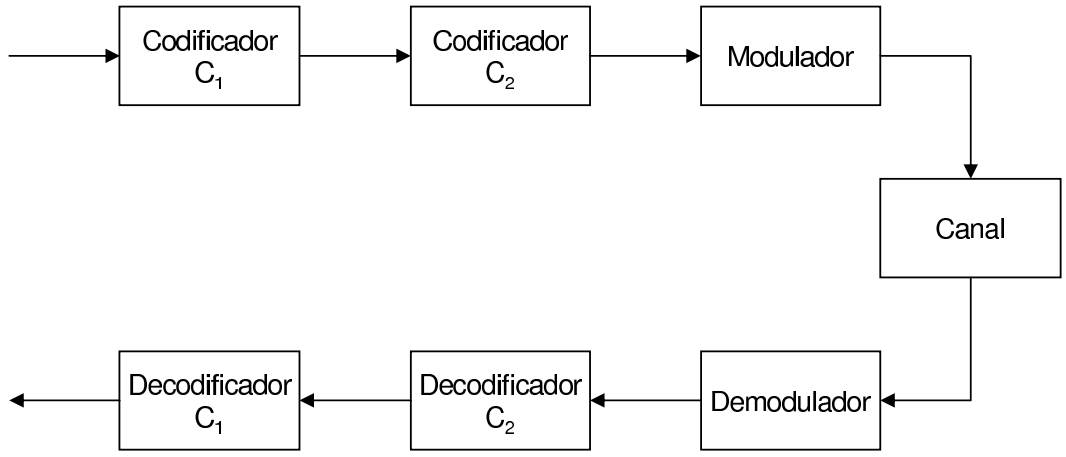

Fig. 5.1: Esquema geral da concatenação serial.

5.2 Códigos Concatenados

A técnica de codificação concatenada [23] é caracterizada pela utilização de dois ou mais códigos componentes, com o intuito de se obter um aumento da capacidade de correção de erros sem incrementar, na mesma proporção, a complexidade do sistema. Nesta aplicação, o sistema de codificação é composto pela combinação de dois ou mais codificadores, que utilizam códigos curtos, para gerar um código longo. O decodificador é formado pelos respectivos decodificadores dos códigos mais simples, fornecendo uma estrutura de decodificação multi-estágio, que é menos complexa do que uma de estágio simples. Pode-se classificar a concatenação em dois tipos: serial e paralela, que serão descritas a seguir.

5.2.1 Concatenação Serial

Na figura 5.1 é ilustrado um esquema de concatenação serial, onde o par codificador/decodificador 1 é comumente chamado codificador/decodificador externo, enquanto o par 2 é chamado interno. Um bloco de k bits é inicialmente codificado pelo código externo $C_1(n_1, k)$. Em seguida, o resultado desta codificação, um vetor de tamanho n_1, é codificado pelo código interno $C_2(n_2, n_1)$. A decodificação completa ocorre então em duas etapas, com a decodificação de C_2 sendo executada primeiramente, seguida da decodificação de C_1.

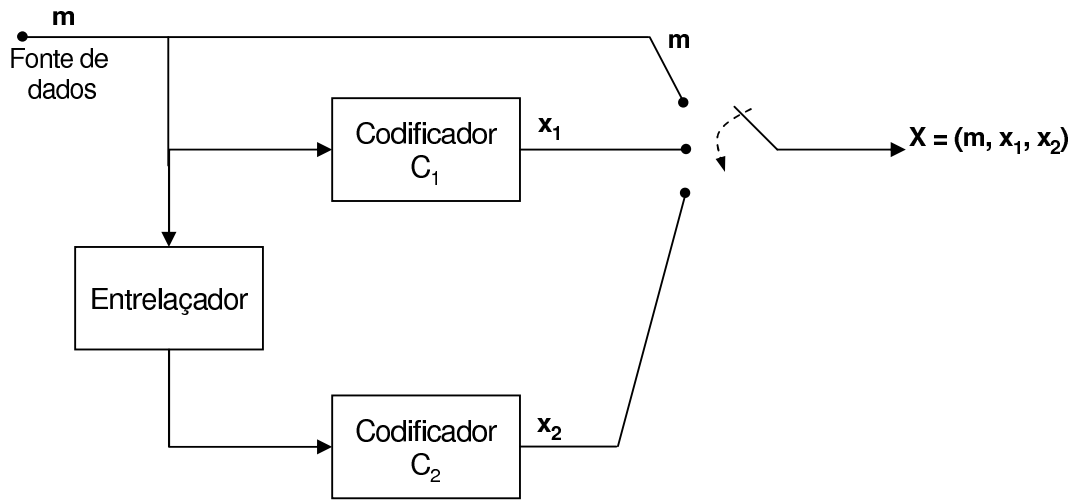

Fig. 5.2: Esquema geral da concatenação paralela.

5.2.2 Concatenação Paralela

A concatenação paralela foi introduzida originalmente por Berrou, Glavieux e Thitimajshima [1]. Da mesma forma que os seriais, os codificadores paralelamente concatenados utilizam dois ou mais codificadores componentes para implementar códigos longos. Na figura 5.2 é ilustrado um esquema de concatenação paralela, onde um bloco de m bits de mensagem é codificado pelo código C_1 para produzir a seqüência de paridade $\mathbf{x_1}$. Paralelamente, os bits de mensagens são entrelaçados e codificados por C_2, originando a seqüência de paridade $\mathbf{x_2}$. Os m bits de mensagem e as duas seqüências $\mathbf{x_1}$ e $\mathbf{x_2}$ são multiplexadas para gerar a seqüência codificada $\mathbf{X}$. Seqüências codificadas tanto por concatenação paralela quanto por serial podem fornecer desempenho muito próximo ao limite de Shannon, por razões ligadas à distribuição de pesos de cada código componente, bem como do código concatenado.

5.2.3 Entrelaçamento

O entrelaçamento é uma técnica de embaralhamento de símbolos, sejam eles binários ou q-ários, utilizada em algumas aplicações de sistemas de comunicação. O entrelaçador tem a função de permutar os símbolos das seqüências de entrada, ou seja, ele reordena os símbolos entrantes de uma forma apenas temporal, não havendo,

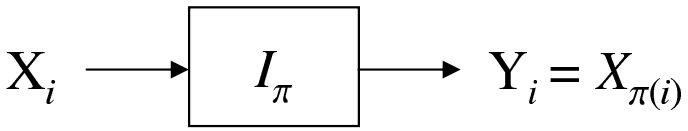

Fig. 5.3: Representação matemática do entrelaçador.

assim, nenhuma alteração na quantidade de cada símbolo de um alfabeto em uma determinada seqüência de entrada. Na figura 5.3 é mostrada uma representação matemática do entrelaçador.

Conforme observado na figura 5.2, o entrelaçador é utilizado na concatenação paralela com o objetivo de aumentar o peso das palavras-código de saída do codificador. Com isso, a probabilidade do decodificador gerar um erro de decodificação devido a uma interpretação errônea da palavra código transmitida é reduzida [24].

Observe novamente a figura 5.2, a seqüência de entrada $\mathbf{m}$ gera uma seqüência codificada $\mathbf{x_1}$ de baixo peso através do codificador convolucional C_1. Para que outra seqüência codificada de baixo peso não seja produzida em $\mathbf{x_2}$, o entrelaçador permuta os símbolos da seqüência $\mathbf{m}$ com o propósito de obter uma seqüência diferente, com maior peso. Vamos ilustrar estes conceitos através de um exemplo. Na figura 5.4, a seqüência de entrada $\mathbf{m_{i0}}$ produz as seqüência de saída $\mathbf{x_{1i}}$ e $\mathbf{x_{2i}}$. Observe que, $\mathbf{m_{10}}$ e $\mathbf{m_{20}}$ são permutações distintas da seqüência $\mathbf{m_{00}}$. Na permutação π_1, o peso da seqüência de paridade permanece o mesmo da seqüência de informação $\mathbf{m_{10}}$. Entretanto, na permutação π_2, o peso da seqüência $\mathbf{x_{22}}$ aumenta em uma unidade quando comparada à $\mathbf{m_{20}}$, mostrando que o peso da palavra-código pode ser aumentado com o uso de entrelaçadores. Na tabela 5.1 é ilustrado o efeito do entrelaçamento nas seqüências de paridade do codificador, em que π_0 representa a permutação identidade, ou seja, não há troca de posição dos bits que compõem a seqüência. Diante dos resultados podemos afirmar que o entrelaçador atua diretamente nas propriedades de distância do código, contribuindo assim para o melhor desempenho dos códigos turbo que utilizam esquemas de codificação concatenada.

Os entrelaçadores comumente utilizados em esquemas de codificação turbo são: entrelaçador de bloco e entrelaçador pseudo-aleatório. O entrelaçador de bloco é o tipo de entrelaçador mais usado nos sistemas de comunicação. Ele é caracterizado por um processo onde os dados são escritos em linhas e lidos ao longo das colunas de

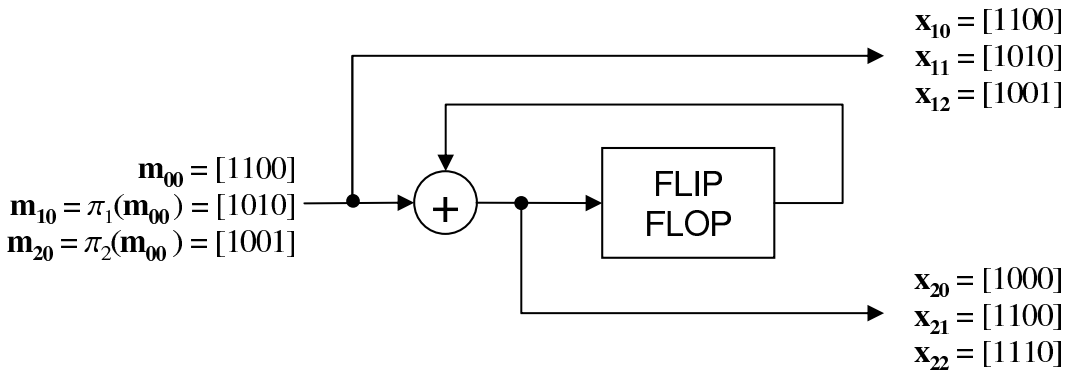

Fig. 5.4: Ilustração da capacidade de aumento de peso do entrelaçador.

	Seqüência de info. $\mathbf{m}_{i0}$	Seqüência de paridade $\mathbf{x}_{2i}$	Peso da palavra-código $(\mathbf{m}_{i0}, \mathbf{x}_{2i})$
π_0	1100	1000	3
π_1	1010	1100	4
π_2	1001	1110	5

Tab. 5.1: Efeito das permutações nas seqüências do codificador da figura 5.4.

matrizes ou vice-versa. Existem quatro variações básicas deste esquema, que variam de acordo com a ordem em que as colunas são lidas (LR: Left-to-Right ou RL: Right-to-Left) e a ordem em que as linhas são lidas (TB:Top-to-Bottom ou BT:Bottom-to-Top). O entrelaçador pseudo-aleatório utiliza uma ordem de permutação definida aleatoriamente para mapear a seqüência de entrada. Porém, uma vez definida, esta ordem se mantém para toda seqüência em sua entrada.

5.3 Códigos Turbo

Para que um código seja caracterizado como turbo, ele deve satisfazer a duas condições: a utilização de pelo menos duas camadas de codificação concatenadas, aplicando-se ou códigos de bloco ou convolucionais, e a implementação de um decodificador iterativo, que para alcançar um desempenho desejável, é recomendado o uso de algoritmos de decodificação com critérios suaves de decisão.

Conforme mencionado na seção 5.2, pode-se utilizar tanto a concatenação serial, quanto a concatenação paralela, entre os códigos componentes do código turbo, de bloco ou convolucionais. Considerando a aplicação da concatenação paralela para códigos turbo de bloco (CTB), temos duas opções de entrelaçamento para os símbolos de informação: o aleatório e o entrelaçamento linha/coluna simples. Ambas as técnicas de entrelaçamento apresentam desempenhos semelhantes, quando aplicados aos códigos turbo de bloco [25]. Isto se deve ao fato de que as linhas e as colunas dos códigos de bloco concatenados são independentes, assim, um entrelaçamento aleatório não acrescenta nenhum desempenho adicional significativo.

Admitindo que o entrelaçamento usado para os códigos turbo de bloco é o linha/coluna, podemos mostrar que a concatenação serial apresenta maiores vantagens em relação à concatenação paralela.

Considere a concatenação de dois códigos de bloco lineares sistemáticos $C_1(n_1, k_1, d_1)$ e $C_2(n_2, k_2, d_2)$. A concatenação paralela de C_1 e C_2 resulta em uma taxa de código R_p levemente superior à taxa de código produzida pela concatenação serial R_s, tal que:

$$\left.\begin{array}{l} R_p = \frac{R_1 R_2}{R_1 + R_2 - R_1 R_2} \\ R_s = R_1 R_2 \end{array}\right\} \Rightarrow R_p > R_s. \tag{5.1}$$

Onde R_1 é a taxa do código C_1 e R_2 é a taxa do código C_2. Entretanto, em relação à distância mínima, a concatenação serial gera um valor d_s maior que a paralela d_p:

$$\left.\begin{array}{l} d_p = d_1 + d_2 - 1 \\ d_s = d_1 \times d_2 \end{array}\right\} \Rightarrow d_s > d_p. \tag{5.2}$$

Para definir qual dos resultados apresentados nas equações (5.1) e (5.2) é o mais determinante na avaliação de desempenho dos métodos de concatenação, podemos utilizar o ganho assintótico de codificação, que é dado por:

$$(Ga)_i = 10 \log(R_i d_i). \tag{5.3}$$

Substituindo os valores da taxa de código e da distância mínima, para as concatenações serial e paralela de códigos de bloco linerares, obtidos através das

equações (5.1) e (5.2), respectivamente, podemos concluir que a concatenação serial apresenta um maior ganho assintótico [25]. Para ilustrar esta conclusão, na tabela 5.2 são mostrados os dados para as duas formas de concatenação utilizando códigos componentes BCH(63,57,3) e (64,57,4).

	$(\mathbf{63,57,3})^{\mathbf{2}}$	$(\mathbf{64,57,4})^{\mathbf{2}}$
$\mathbf{R_s}$	0,818	0,793
$\mathbf{d_s}$	9	16
$(\mathbf{Ga})_{\mathbf{s}}[dB]$	8,67	11,03
$\mathbf{R_p}$	0,826	0,802
$\mathbf{d_p}$	5	7
$(\mathbf{Ga})_{\mathbf{p}}[dB]$	6,16	7,49
$\frac{\mathbf{R_s}}{\mathbf{R_p}}$	0,990	0,989
$\Delta(\mathbf{Ga})[dB]$	2,51	3,54

Tab. 5.2: Tabela comparativa das formas de concatenação de códigos de bloco.

Vale ressaltar que os códigos de bloco concatenados serialmente possuem uma propriedade atrativa em que todas as linhas e colunas do código são palavras-código dos correspondentes códigos componentes.

5.3.1 Códigos Produto

O princípio de construção dos códigos produtos, ou iterados, foi introduzido por Elias [26] em 1954 e se baseia nas características dos códigos serialmente concatenados. Com a aplicação dos conceitos dos códigos produto, pode-se construir códigos de bloco longos a partir de códigos de bloco curtos de uma maneira simples e eficiente. Vamos exemplificar a construção de códigos produto de bloco. Considere dois códigos de bloco lineares $C_1(n_1, k_1, d_1)$ e $C_2(n_2, k_2, d_2)$ de tamanho curto. Conforme [26], podemos construir o código produto $P = C_1 \otimes C_2$ montando a matriz de palavras-código, ilustrada na figura 5.5, de acordo com o seguinte procedimento:

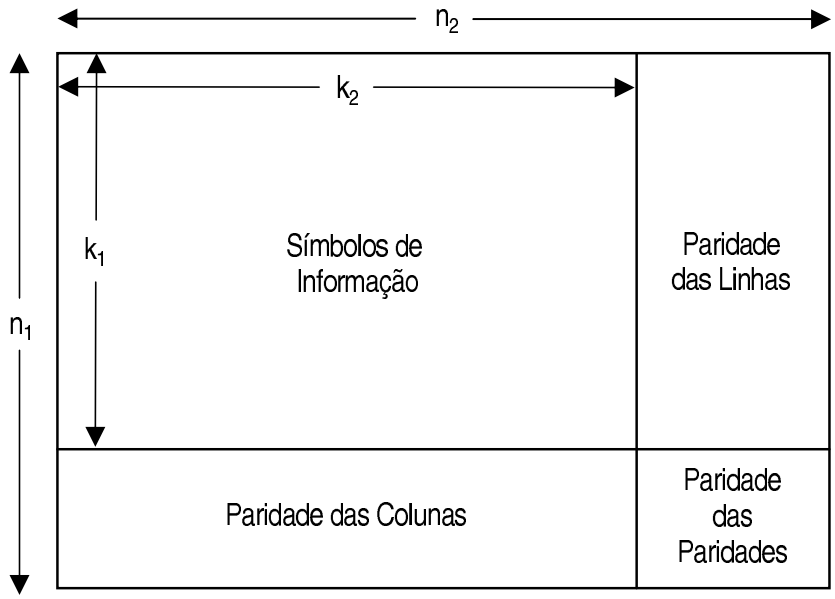

Fig. 5.5: Construção do código $P = C_1 \otimes C_2$.

1. Arranjar $k_1 \times k_2$ bits de informação em uma matriz de k_1 linhas e k_2 colunas;

2. Codificar as k_1 linhas utilizando o código C_2;

3. Codificar as n_2 colunas utilizando o código C_1.

Desta maneira, o código produto resultante terá os seguintes parâmetros $n = n_1 n_2$, $k = k_1 k_2$, $d = d_1 d_2$ e sua taxa será $R = R_1 R_2$. Observe que o código produto resulta em uma distância mínima alta, porém, a taxa do código é impactada negativamente, evidenciando que o compromisso entre estes dois parâmetros deve ser considerado. Vale ressaltar que, os códigos produto não se resumem a duas dimensões, conforme ilustrado no exemplo, eles podem atingir dimensões maiores através de um procedimento de construção análogo ao apresentado.

A complexidade dos códigos produtos pode ser reduzida com a utilização da decodificação sequencial das linhas e colunas da matriz resultante. Todavia, para atingir um desempenho ótimo, devemos usar a decodificação por decisão suave nos códigos componentes. Desta maneira, podemos iterar a decodificação sequencial do código produto e reduzir a taxa de erro de bit após cada iteração.

5.4 Decodificação Turbo

Nesta seção, serão descritos os procedimentos necessários para a aplicação da decodificação turbo. A explicação teórica será abordada para símbolos binários, como já é bem difundida na literatura. Na seção 5.5, serão apresentadas as etapas da decodificação turbo que necessitam de uma adaptação e/ou complementação para a aplicação com símbolos q-ários.

5.4.1 Logaritmo da Razão de Verossimilhança

O Logaritmo da Razão de Verossimilhança, apresentado no capítulo 3, também é utilizado na decodificação turbo. Vejamos uma descrição mais minuciosa de seu conceito, bem como as particularidades para a aplicação na decodificação turbo. Considere a variável aleatória binária u em $GF(2)$, tal que o elemento lógico 0 seja mapeado como -1 e o elemento lógico 1 como $+1$. A definição de Logaritmo da Razão de Verossimilhança (LLR, do inglês *Log-Likelihood Ratio*) $L(u)$ para a variável u é dada por:

$$L(u) = ln\left(\frac{P(u=+1)}{P(u=-1)}\right). \tag{5.4}$$

A grandeza $L(u)$ é freqüentemente referenciada como um valor "suave"da variável aleatória u. Considerando o caso de símbolos binários, o sinal de $L(u)$ é a decisão abrupta de u, enquanto seu valor é a confiabilidade desta decisão, pois é fácil ver por (5.4) que, à medida que $L(u)$ cresce, a probabilidade de $u = +1$ também cresce. Similarmente, à medida que $L(u)$ decresce, a probabilidade de $u = -1$ aumenta. Desta forma, $L(u)$ fornece uma espécie de confiabilidade para u.

Considere a variável u descrita anteriormente como um sinal binário a ser transmitido em um canal AWGN. Podemos calcular $L(u|x)$ por sua definição:

$$L(u|x) = ln\left(\frac{P(u=+1|x)}{P(u=-1|x)}\right). \tag{5.5}$$

Aplicando a Regra de Bayes à equação (5.5), temos:

$$\begin{aligned} L(u|x) &= \ln\left(\frac{P(x|u=+1)}{P(x|u=-1)}\frac{P(u=+1)}{P(u=-1)}\right) \\ &= \ln\left(\frac{P(x|u=+1)}{P(x|u=-1)}\right) + \ln\left(\frac{P(u=+1)}{P(u=-1)}\right) \\ &= L(x|u) + L(u), \end{aligned} \tag{5.6}$$

onde $L(x|u)$, também chamada de $L_C(x)$, representa o LLR das probabilidades de transição de canal e $L(u)$, o LLR a priori de u. A equação (5.6) pode ser escrita da seguinte forma:

$$L'(\hat{u}) = L_C(x) + L(u). \tag{5.7}$$

Considerando os códigos sistemáticos, pode ser mostrado que o LLR $L(\hat{u})$ referente à saída do decodificador, ou seja, a confiabilidade da decodificação, é igual a:

$$L(\hat{u}) = L'(\hat{u}) + L_e(\hat{u}), \tag{5.8}$$

onde $L'(\hat{u})$ representa o LLR do bit de entrada do decodificador, e $L_e(\hat{u})$, chamada de LLR extrínseca, representa a informação adicional que é obtida do processo de decodificação [1].

Aplicando (5.7) em (5.8), temos:

$$L(\hat{u}) = L_C(x) + L(u) + L_e(\hat{u}). \tag{5.9}$$

5.4.2 Decodificação Iterativa

Nesta seção, os princípios da decodificação iterativa serão descritos, ou seja, de que forma a informação é realimentada de uma iteração para outra usando um algoritmo MAP. Considere o decodificador SISO (do inglês, *Soft-Input Soft-Output*) ilustrado na figura 5.6. Assume-se que os símbolos de entrada u são equiprováveis,

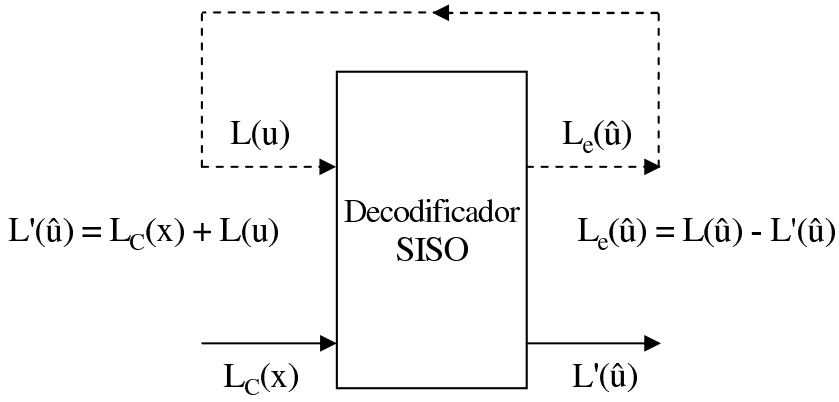

Fig. 5.6: Decodificação SISO pata códigos sistemáticos.

fazendo com que na primeira iteração do decodificador SISO, o LLR a priori, L(u), seja igual a zero. O LLR das probabilidades de transição do canal, $L_C(x)$, é calculado através do logaritmo da razão entre os valores obtidos das densidades de probabilidade $P(x|u=+1)$ e $P(x|u=-1)$. Desta forma, $L'(\hat{u}) = L_C(x)$. A saída do decodificador $L(\hat{u})$ é calculada através da soma de $L'(\hat{u})$ com a LLR extrínseca $L_e(\hat{u})$. Esta informação extrínseca é então reenviada à entrada do decodificador como o novo valor $L(u)$ da próxima iteração.

Vamos exemplificar a explanação do parágrafo anterior através do código produto ilustrado na figura 5.7. Neste arranjo, são representados: k_1k_2 símbolos de informação, $(n_2 - k_2)$ colunas com a informação da paridade horizontal (p_h), $(n_1 - k_1)$ linhas com a informação da paridade vertical (p_v) e os blocos L_{eh} e L_{ev}, constituindo as informações extrínsecas relativas às decodificações horizontal e vertical, respectivamente. O algoritmo de decodificação deste código produto obedece ao seguinte procedimento [27]:

1. Inicialmente, fazer $L(u) = 0$;
2. Decodificar horizontalmente e, através de (5.9), obter a informação extrínseca horizontal: $L_{eh}(\hat{u}) = L(\hat{u}) - L_C(x) - L(u)$;
3. Fazer $L(u) = L_{eh}(\hat{u})$;
4. Decodificar verticalmente e, através de (5.9), obter a informação extrínseca vertical: $L_{ev}(\hat{u}) = L(\hat{u}) - L_C(x) - L(u)$;

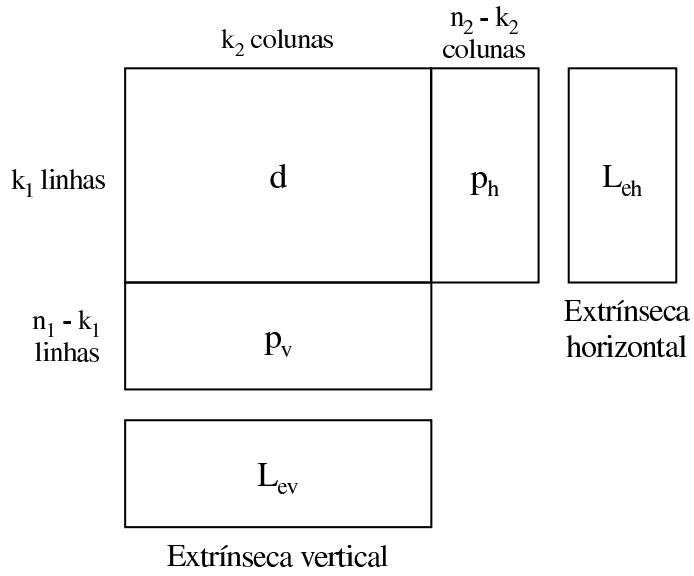

Fig. 5.7: Código produto bidimensional.

5. Fazer $L(u) = L_{ev}(\hat{u})$;

6. Se houve iterações suficientes para que a confiabilidade atinja um valor mínimo pré-estabelecido, ir para o passo 7, caso contrário, ir para o passo 2;

7. A saída será: $L(\hat{u}) = L_C(x) + L_{eh}(\hat{u}) + L_{ev}(\hat{u})$.

Para símbolos binários, no final, o decodificador fará uma decisão abrupta baseada na saída calculada. Se $L(\hat{u}) \geq 0$, o bit decodificado será 1, entretanto, se $L(\hat{u}) < 0$, o bit decodificado será 0.

5.4.3 Confiabilidade da Decisão

O algoritmo de decodificação turbo, utilizado para o código produto aplicado nas simulações da seção 5.5.2, diferencia-se do algoritmo SISO descrito na seção anterior por não atualizar os LLRs a priori. Porém, o algoritmo de decodificação turbo usufrui do conceito de LLR para estabelecer uma confiabilidade para cada decisão $\mathbf{D} = (d_1, d_2, ..., d_j, ..., d_N)$ do algoritmo de Chase, para símbolos binários, ou do algoritmo adaptado proposto no capítulo 3, para símbolos q-ários. Para o caso de símbolos binários, esta confiabilidade é dada por:

$$\Lambda(d_j) = \ln\left(\frac{p\{e_j = +1|\mathbf{R}\}}{p\{e_j = -1|\mathbf{R}\}}\right), \tag{5.10}$$

onde:

$$\begin{aligned} p\{e_j = +1|\mathbf{R}\} &= \sum_{\mathbf{C}^i \in S_j^{+1}} p\{\mathbf{E} = \mathbf{C}^i|\mathbf{R}\} \qquad e \\ p\{e_j = -1|\mathbf{R}\} &= \sum_{\mathbf{C}^i \in S_j^{-1}} p\{\mathbf{E} = \mathbf{C}^i|\mathbf{R}\}, \end{aligned}$$

com $p\{\}$ representando probabilidade, $\mathbf{E}$ a palavra transmitida e $\mathbf{R}$ a palavra recebida. S_j^{+1} e S_j^{-1} são subconjuntos de palavras-código, $\mathbf{C}^i = (c_1^i, c_2^i, ..., c_j^i, ..., c_N^i)$, tal que $c_j^i = +1$ e -1, respectivamente. Aplicando-se a Regra de Bayes e assumindo palavras-código igualmente prováveis, obtemos que:

$$\Lambda(d_j) = \ln\left(\frac{\sum_{\mathbf{C}^i \in S_j^{+1}} p\{\mathbf{C}^i|\mathbf{R} = \mathbf{E}\}}{\sum_{\mathbf{C}^i \in S_j^{-1}} p\{\mathbf{C}^i|\mathbf{R} = \mathbf{E}\}}\right). \tag{5.11}$$

A fim de citar mais adiante a proposta apresentada em [3], para o cálculo da confiabilidade de decisão em canais não gaussianos, inicialmente abordaremos uma aproximação da confiabilidade de decisão para o canal AWGN. Considere que os bits da palavra-código transmitida são independentes entre si. De acordo com [2], podemos manipular (5.11) e obter a seguinte equação:

$$\Lambda'(d_j) = \frac{1}{4}\left(|\mathbf{R} - \mathbf{C}^{-1(j)}|^2 - |\mathbf{R} - \mathbf{C}^{+1(j)}|^2\right), \tag{5.12}$$

onde $\mathbf{C}^{+1(j)}$ e $\mathbf{C}^{-1(j)}$ são palavras-código em, respectivamente, S_j^{+1} e S_j^{-1}; e ambas com as menores distâncias mínimas de $\mathbf{R}$. $\Lambda'(d_j)$ tem o mesmo sinal de d_j e seu valor absoluto representa a confiabilidade da decisão.

Para calcularmos a confiabilidade da decisão d_j, precisamos identificar duas palavras-código $\mathbf{C}^{+1(j)}$ e $\mathbf{C}^{-1(j)}$, sendo que uma delas é $\mathbf{D}$, e a outra, considerada concorrente de $\mathbf{D}$, é definida como $\mathbf{C}$, a palavra-código mais próxima de $\mathbf{R}$ com $c_j \neq d_j$. Dadas as palavras-código $\mathbf{C}$ e $\mathbf{D}$, podemos reescrever (5.12) da seguinte

forma:

$$\Lambda'(d_j) = \frac{1}{4}\left(|\mathbf{R}-\mathbf{C}|^2 - |\mathbf{R}-\mathbf{D}|^2\right) d_j. \tag{5.13}$$

Como $\mathbf{D}$ está mais próximo de $\mathbf{R}$ do que $\mathbf{C}$, conclui-se que o termo entre parênteses de (5.13) será sempre positivo. O fator multiplicativo $\frac{1}{4}$ é usado para normalizar o valor absoluto da confiabilidade entre 0 e 1.

A equação (5.13) considera que a palavra-código concorrente $\mathbf{C}$, que sempre difere de $\mathbf{D}$ pelo bit da posição j ($c_j = -d_j$), possa ser encontrada. A probabilidade de se encontrar as palavras $\mathbf{C}$ e $\mathbf{D}$ aumenta com o número de bits menos confiáveis da palavra recebida, que depende de qual algoritmo de Chase é utilizado. A complexidade do decodificador pode aumentar exponencialmente com este número de bits, e portanto fazer com que procuremos um compromisso entre desempenho e complexidade.

Diante desta necessidade, $\mathbf{C}$ só é pesquisada em um subconjunto de palavras-código geradas pelo algoritmo de Chase e não em todo código. Desta maneira, em alguns casos, a palavra $\mathbf{C}$ pode não ser encontrada para a decisão d_j, isto é, todas as palavras propostas tem o mesmo sinal na posição j. Neste caso, uma solução foi proposta em [2] para computar a saída do algoritmo, dada pela seguinte equação:

$$\Lambda'(d_j) = \beta d_j \quad , \qquad \beta \geq 0. \tag{5.14}$$

Esta aproximação foi usada para canais AWGN e se justifica pelas seguintes razões: a) o sinal de $\Lambda'(d_j)$ é igual a d_j; b) se $\mathbf{C}$ não for encontrada no subconjunto gerado pelo algoritmo de Chase, isto significa que $\mathbf{C}$ está bem longe de $\mathbf{R}$ em termos de distância Euclidiana, então, a probabilidade da decisão d_j estar correta é alta e a confiabilidade de d_j também.

Baseado na equação (5.14) proposta em [2], foi desenvolvida em [3] uma maneira de calcular a confiabilidade da decisão d_j, com a possibilidade de aplicação em outros tipos de canais, dada por:

$$\Lambda'(d_j) = \beta \Phi(d_j), \tag{5.15}$$

onde $\Phi(d_j)$ é o logaritmo da razão de verossimilhança da decisão d_j. A fim de demonstrarmos a aplicabilidade de (5.15), vamos compará-la à equação (5.14) considerando um sistema em um canal AWGN e que utiliza uma modulação BPSK, assim:

$$\Phi(d_j) = \ln\left(\frac{p(d_j|e_j = +\sqrt{E_b})}{p(d_j|e_j = -\sqrt{E_b})}\right), \tag{5.16}$$

onde:

$$p(d_j|e_j = y) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-(d_j - y)^2}{2\sigma^2}\right) \tag{5.17}$$

e E_b é a energia média por bit de informação. A equação (5.16) nos diz que $\Phi(d_j)$ tem o mesmo sinal de d_j e conseqüentemente $\Lambda'(d_j)$ também. Assim como em (5.14), o valor absoluto de $\Phi(d_j)$ indica o quanto a decisão d_j é confiável. Também é possível mostrar que (5.15) pode ser reduzida a (5.14) por um fator multiplicativo $2\frac{\sqrt{E_b}}{\sigma^2}$, quando aplicada a uma canal AWGN. Vale ressaltar que, pelo fato de (5.15) depender de E_b, a proposta apresentada por [3] requer que o receptor tenha a informação da RSR do canal utilizado.

A grandeza β, chamada fator de confiabilidade, representa o valor médio da confiabilidade das decisões d_j quando não há palavra-código concorrente $\mathbf{C}$ no subconjunto de palavras candidatas geradas pelo algoritmo de Chase. Nas simulações realizadas em [3], foi mostrado que os valores a serem atribuídos a β, para que o sistema alcance um melhor desempenho, devem estar contidos no intervalo $[0,2; 1,0]$ e seguir uma regra linear simples que contemple a quantidade de iterações definida para o decodificador.

5.4.4 Decodificação Turbo de Códigos Produto

Nesta seção, será apresentada a metodologia para a decodificação turbo de códigos produtos binários. As adaptações para códigos q-ários serão apresentadas na seção 5.5.

Considerando o código produto P, descrito na seção 5.3.1, abordaremos a sua decodificação iterativa, isto é, das linhas e colunas que o compõem em mais de

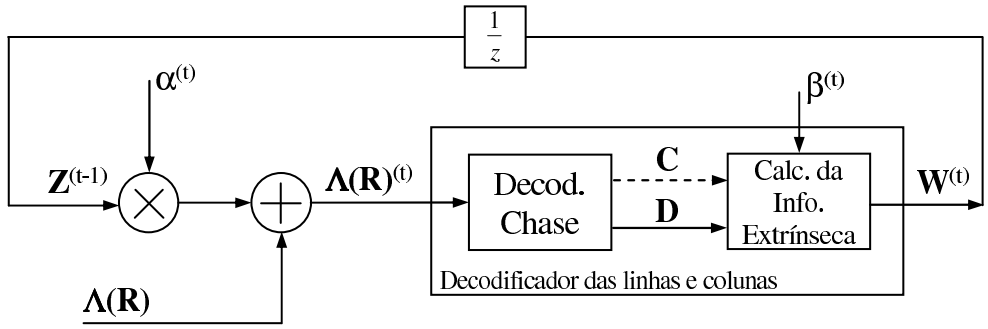

Fig. 5.8: Diagrama esquemático do decodificador turbo de bloco.

um ciclo de decodificação. Seja $\mathbf{\Lambda(R)}$ a matriz que representa o sinal recebido na entrada do decodificador turbo, composta por valores de confiabilidade sobre os símbolos recebidos, parâmetro fundamental na execução do algoritmo SISO de cada decodificação. Cada elemento de $\mathbf{\Lambda(R)}$ é calculado da seguinte forma:

$$\Lambda(r_j) = \ln\left(\frac{p(r_j|b_j = 0)}{p(r_j|b_j = 1)}\right), \tag{5.18}$$

com $j \in [0, N^2 - 1]$.

O primeiro decodificador componente atua sobre as linhas (ou colunas) de P usando $\mathbf{\Lambda(R)}$ como entrada. A decodificação suave é executada através do algoritmo de Chase e sua saída é calculada segundo (5.13) ou (5.15). Na figura 5.8, é ilustrado o procedimento da decodificação de um código produto baseado em códigos de bloco lineares.

A entrada da t-ésima iteração do decodificador de linhas e colunas é dada por:

$$\Lambda(\mathbf{R})^{(t)} = \Lambda(\mathbf{R}) + \alpha^{(t)}\mathbf{Z}^{(t-1)}. \tag{5.19}$$

Vale ressaltar que, como cada iteração compreende a atuação dos dois decodificadores componentes (linha e coluna), o total de passos será sempre o dobro do total de iterações.

As matrizes $\mathbf{Z}^{(t-1)}$ e $\mathbf{W}^{(t)}$ representam a informação extrínseca calculada pelo decodificador. A diferença entre elas, apesar de numericamente iguais, é que a primeira é a informação extrínseca calculada pelo decodificador anterior e é usada para atualizar a entrada do decodificador atual, enquanto a segunda é a saída do

decodificador atual e será usada no cálculo da entrada do próximo decodificador. Sendo assim, a informação extrínseca obedece a uma estrutura recursiva e depende da confiabilidade gerada em cada decodificador SISO componente.

De acordo com o método heurístico discutido em [28], a confiabilidade da decisão é descrita da seguinte forma:

$$L(a_j) = Fz_j + w_j, \tag{5.20}$$

onde F é um fator multiplicativo. Trazendo (5.20) para o nosso contexto, temos que:

$$\Lambda(d_j) = \alpha z_j + w_j, \tag{5.21}$$

ou matricialmente,

$$\mathbf{\Lambda}(d_j)^{(t)} = \alpha^{(t)}\mathbf{Z}^{(t-1)} + \mathbf{W}^{(t)}. \tag{5.22}$$

Substituindo (5.19) em (5.22) e isolando o termo $\mathbf{W}^{(t)}$, temos:

$$\mathbf{W}^{(t)} = \mathbf{\Lambda}(d_j)^{(t)} - (\mathbf{\Lambda}(\mathbf{R})^{(t)} - \mathbf{\Lambda}(\mathbf{R})). \tag{5.23}$$

Podemos simplificar (5.23), considerando que o valor constante de $\mathbf{\Lambda}(\mathbf{R})$ não altera as decisões. Então,

$$\mathbf{W}^{(t)} = \mathbf{\Lambda}(d_j)^{(t)} - \mathbf{\Lambda}(\mathbf{R})^{(t)}. \tag{5.24}$$

A equação (5.24) expressa que a informação extrínseca gerada em cada decodificador componente é dada pela diferença entre $\mathbf{\Lambda}(d_j)^{(t)}$, confiabilidade da decisão dada pelo algoritmo de Chase, e $\mathbf{\Lambda}(\mathbf{R})^{(t)}$, confiabilidade da nova entrada deste decodificador. Baseado na equação (5.15), podemos aproximar $\mathbf{\Lambda}(d_j)^{(t)}$ para:

$$\mathbf{\Lambda}(d_j)^{(t)} = \beta^{(t)}\mathbf{\Phi}(d_j)^{(t)}. \tag{5.25}$$

A grandeza α, chamada fator de escala, é usada para reduzir o efeito da informação extrínseca no decodificador nos primeiros passos de decodificação. Ela varia

no intervalo $[0,0;1,0]$ e assim como β, pode ter seus valores pré-determinados de maneira intuitiva ou seguir um critério específico, como por exemplo, variar segundo a função logarítmica no intervalo mencionado, que irá gerar um melhor desempenho para a decodificação [3].

5.5 Decodificação Turbo q-ária em Canais FFH-CDMA

Na seção 5.4, foram apresentados os procedimentos gerais para a decodificação turbo de códigos binários. Nesta seção, será apresentada uma proposta para a adequação da decodificação turbo para códigos q-ários. Será dada ênfase à aplicação da decodificação turbo q-ária em canais FFH-CDMA, visto que uma comparação com os resultados obtidos por [3], para a decodificação turbo binária em um canal FFH-CDMA, é desejável.

5.5.1 Proposta para a Decodificação Turbo q-ária

Na figura 5.9, é apresentado um sistema de comunicação FFH-CDMA com as adaptações para suportar a decodificação turbo q-ária proposta. Seguindo a mesma estrutura dos conceitos de decodificação turbo introduzidos na seção 5.4, porém, alterando algumas denotações de variáveis devido às necessidades impostas pelas adaptações, iniciaremos a apresentação da proposta pelo Logaritmo da Razão de Verossimilhança de símbolos q-ários em um canal FFH-CDMA. Considerando o código produto, seja $\mathbf{\Lambda}(\mathbf{Y})$ a matriz que representa a confiabilidade dos símbolos q-ários recebidos do FFH-CDMA, onde $\mathbf{Y}$ é a matriz cujo elemento Y_{ij} é o símbolo definido pelo receptor, com $i = 0, 1, ..., (N-1)$ e $j = 0, 1, ..., (N-1)$. No caso de símbolos binários, vimos que a partir do valor de um elemento de $\mathbf{\Lambda}(\mathbf{R})$ da equação (5.18) sabemos, através de seu sinal (positivo ou negativo), o valor do bit por decisão brusca. Porém, para o caso de símbolos q-ários, não podemos fazer tal dedução, sendo necessário, assim, o uso de um receptor para definir o símbolo recebido. Utilizamos um receptor de máxima verossimilhança para realizar esta decisão, assim, o símbolo escolhido $Y_{ij} = m$ obedece a seguinte desigualdade:

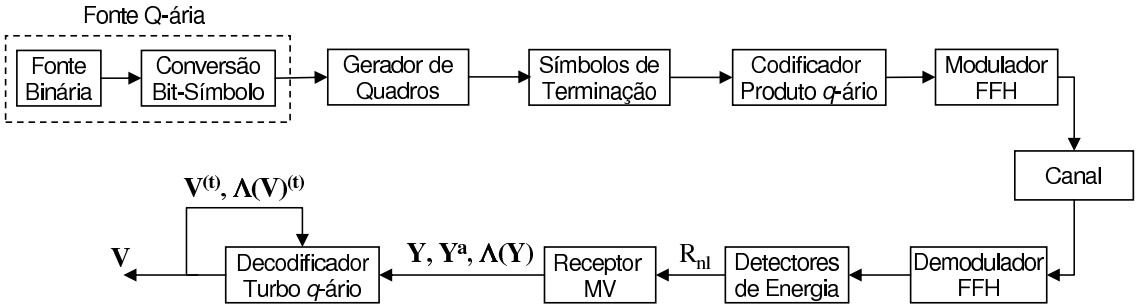

Fig. 5.9: Sistema de comunicação FFH-CDMA com decodificação turbo q-ária.

$$P(Y_{ij} = m|\mathbf{R}_{ij}) > P(Y_{ij} = n|\mathbf{R}_{ij}), \qquad \forall \;\; n \neq m. \tag{5.26}$$

Observe que $\mathbf{R}_{ij}$ também é uma matriz cujos elementos são os valores de R_{nl} apresentados na seção 4.2. Desta maneira, aplicando o teorema de Bayes à equação (5.26), podemos calcular a desigualdade fazendo uso da equação (4.9). Como o algoritmo de Chase q-ário, proposto na seção 3.3, será utilizado no decodificador turbo q-ário, faz-se necessária a obtenção da matriz $\mathbf{Y}^a$, com $Y^a_{ij} = k$, tal que:

$$P(Y_{ij} = m|\mathbf{R}_{ij}) > P(Y_{ij} = k|\mathbf{R}_{ij}) > P(Y_{ij} = n|\mathbf{R}_{ij}), \qquad \forall \;\; n \neq m \neq k. \tag{5.27}$$

Uma vez obtidas as matrizes $\mathbf{Y}$ e $\mathbf{Y}^a$, o próximo passo é o cálculo da matriz de confiabilidade $\mathbf{\Lambda}(\mathbf{Y})$. Para isso, utilizamos o mesmo procedimento descrito na seção 4.3. Assim, o valor da confiabilidade de um símbolo recebido através de um canal FFH-CDMA é:

$$\Lambda(Y_{ij}) = \ln\left(\frac{p(\mathbf{R}_{ij}|Y^{(1)}_{ij})}{\sum_{k=2}^{q} p(\mathbf{R}_{ij}|Y^{(k)}_{ij})}\right). \tag{5.28}$$

Dado que os valores das confiabilidades do canal foram calculados, devemos agora realizar a decodificação das linhas e colunas da matriz produto recebida $\mathbf{Y}$. O decodificador utilizado implementa a proposta do algoritmo de Chase adaptado para símbolos q-ários. Os detalhes desta proposta, inclusive da geração dos padrões de teste, encontram-se na seção 3.3.

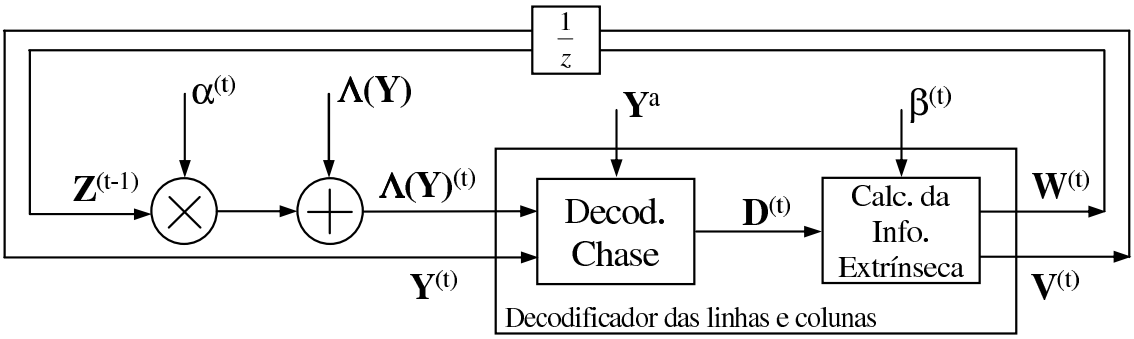

Fig. 5.10: Decodificador turbo q-ário para o caso $Y_{ij}^{(t)} = D_{ij}^{(t)}$.

Conforme mencionado anteriormente para os símbolos binários, podemos saber o valor do bit através do sinal (positivo ou negativo) da sua confiabilidade. Porém, para o caso de símbolos q-ários, a informação do valor do símbolo não está implícita no valor da confiabilidade. Desta maneira, faz-se necessário informar ao decodificador turbo q-ário a matriz das decisões $\mathbf{Y}$ realizadas pelo receptor, suas confiabilidades $\mathbf{\Lambda}(\mathbf{Y})$ e a matriz $\mathbf{Y}^a$ para o decodificador Chase q-ário interno.

Observe que, para o caso binário, o valor recalculado das confiabilidades para cada iteração, através de realimentação no decodificador turbo, já informa diretamente se houve alteração ou não no bit de entrada para a próxima iteração. No caso q-ário, não encontramos esta informação implícita.

Para cada iteração do decodificador turbo q-ário, devemos comparar o valor do símbolo $Y_{ij}^{(t)}$ da palavra de entrada da t-ésima iteração com o valor do símbolo $D_{ij}^{(t)}$ da palavra de saída do decodificador de Chase da t-ésima iteração. Quando $Y_{ij}^{(t)} = D_{ij}^{(t)}$, ou seja, o decodificador de Chase q-ário não corrigir este símbolo da palavra-código, devemos proceder o cálculo da confiabilidade da próxima iteração conforme a figura 5.10. Observe que o procedimento ilustrado na figura 5.10 é similar ao apresentado para o caso binário na seção 5.4.4, exceto pela inclusão da informação da palavra a ser decodificada na t-ésima iteração $Y^{(t)}$ e da seqüência Y^a referente à $Y^{(t)}$.

A decodificação turbo q-ária apresenta uma particularidade quando temos $Y_{ij}^{(t)} \neq D_{ij}^{(t)}$. Neste caso, não podemos utilizar o procedimento apresentado na figura 5.10 porque não necessariamente o símbolo de entrada na próxima iteração

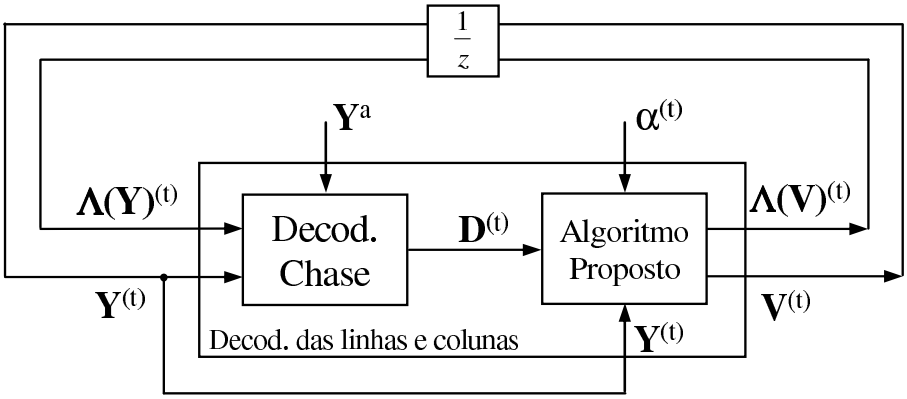

Fig. 5.11: Decodificador turbo q-ário para o caso $Y_{ij}^{(t)} \neq D_{ij}^{(t)}$.

do decodificador turbo será o mesmo da saída do decodificador de Chase q-ário, assim, o método para calcular o valor da confiabilidade para a próxima iteração deve ser diferente. Tendo em vista esta particularidade, propomos um procedimento para a decodificação turbo q-ária cujo diagrama em blocos é ilustrado na figura 5.11. Pelo fato de podermos conhecer o valor do bit pelo sinal da confiabilidade, no caso binário podemos considerar que o valor da confiabilidade $\Lambda(R)$ varia dentro de um segmento com extremidades -1 e $+1$, conforme a figura 5.12a. Não podemos fazer esta mesma consideração para o caso q-ário devido a diversidade de símbolos. Porém, para os casos onde $Y_{ij}^{(t)} \neq D_{ij}^{(t)}$, podemos nos basear na figura 5.12b para calcular o valor da confiabilidade da próxima iteração, ou seja, $\Lambda(V_{ij})^{(t)}$. Observe a figura 5.12b, considerando a t-ésima iteração, o ponto **A**, no eixo das abscissas, representa o valor da confiabilidade do símbolo Y_{ij} da palavra-código de entrada, $\Lambda(Y_{ij})^{(t)}$, e o ponto **B**, no eixo das ordenadas, representa o valor da confiabilidade do símbolo d_j da palavra-código decodificada pelo algoritmo de Chase q-ário, $\Lambda(d_j)^{(t)}$, cujo valor pode ser encontrado conforme a equação (5.25). Devemos encontrar o ponto **C** traçando uma reta perpendicular ao segmento $\overline{\mathbf{AB}}$ de tal modo que o segmento $\overline{\mathbf{AC}}$ seja igual a $\alpha^{(t)} \times \overline{\mathbf{AB}}$, onde $\alpha^{(t)}$ é o mesmo fator de escala utilizado no caso binário. Através de alguns cálculos trigonométricos básicos, podemos encontrar o segmento $\overline{\mathbf{OD}}$, cujo valor representará o valor da confiabilidade a ser utilizado na próxima iteração, ou seja, $\Lambda(V_{ij})^{(t)}$. Observe que, dependendo dos valores de **A**, **B** e $\alpha^{(t)}$, o ponto **D** poderá pertencer ao eixo das abscissas $\Lambda(Y_{ij})^{(t)}$ ou ao eixo das

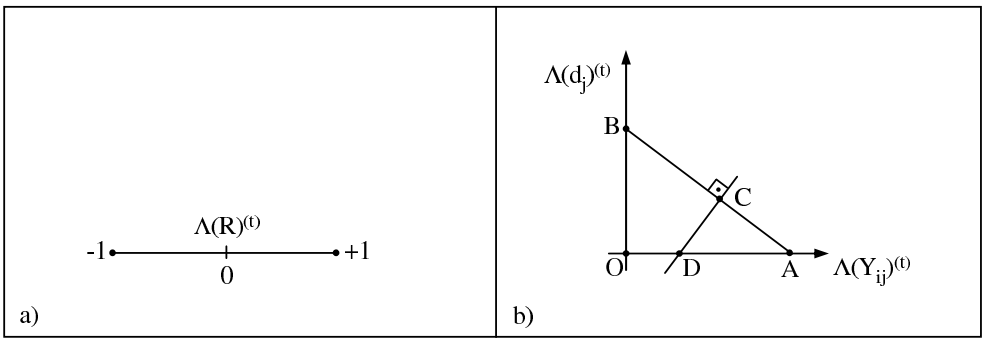

Fig. 5.12: Projeções cartesianas para a decodificação turbo a) binária e b) q-ária.

ordenadas $\Lambda(d_j)^{(t)}$, isto definirá se o símbolo da próxima iteração V_{ij} será igual a, respectivamente, Y_{ij} ou d_j.

Analisando a proposta descrita acima para o cálculo da confiabilidade e símbolo da próxima iteração, podemos observar que, quando $\alpha^{(t)} = 0$, ou seja, o valor calculado de $\Lambda(d_j)^{(t)}$ não é confiável, teremos $\Lambda(V_{ij})^{(t)} = \Lambda(Y_{ij})^{(t)}$ e $V_{ij} = Y_{ij}$. Porém, quando $\alpha^{(t)} = 1$, ou seja, o valor calculado de $\Lambda(d_j)^{(t)}$ é muito confiável, teremos $\Lambda(V_{ij})^{(t)} = \Lambda(d_j)^{(t)}$ e $V_{ij} = d_j$. Com isto, podemos concluir que a proposta apresentada é coerente com seu propósito.

5.5.2 Resultados

Os procedimentos propostos para a decodificação q-ária suave, seção 4.3, e para a decodificação turbo q-ária, seção 5.5.1, são aplicados a um sistema FFH-CDMA para que possamos mensurar o seu desempenho com as novas adaptações. Os resultados obtidos nas simulações serão comparados com os resultados apresentados em [3] para o mesmo sistema FFH-CDMA, porém com decodificação turbo binária, como também com os resultados obtidos em [20].

O primeiro sistema FFH-CDMA analisado utiliza uma modulação 16-FSK, $L = 9$ *chips* por padrão de salto em freqüência, $RSR = 25dB$, um código produto composto de dois códigos de Reed-Solomon RS(15,9,7) em $GF(16)$ e 8 iterações no decodificador turbo. O código produto resultante $P_a(225, 81, 49)$ possui em taxa $r = 0,36$. Vale lembrar que, apesar dos códigos produtos analisados possuírem uma alta capacidade de correção, sua aplicação deve ser ponderada de acordo com a

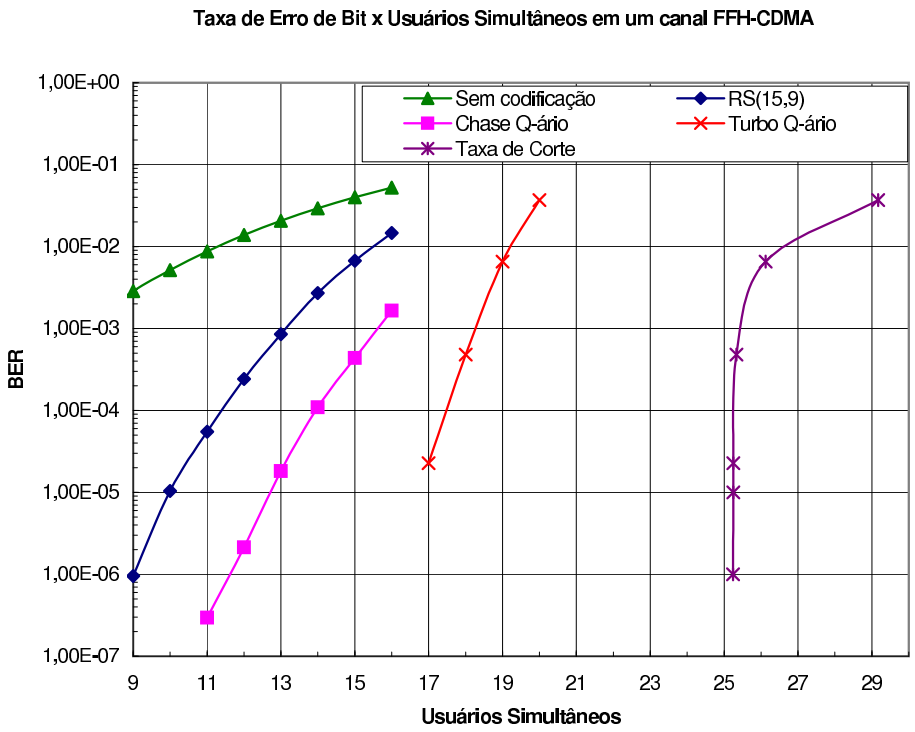

Fig. 5.13: Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando o número de usuários simultâneos, com $M = 16$, $L = 9$ e $P_a = (225, 81, 49)$.

eficiência espectral requerida, pois esta é dependente da taxa do código r.

Na figura 5.13 é ilustrado o desempenho da proposta de decodificação turbo q-ária em um sistema FFH-CDMA em função do número de usuários simultâneos U. Observe que, devido à alta capacidade de correção da decodificação turbo, a curva de desempenho se aproxima ainda mais da taxa de corte do canal FFH-CDMA. Para $U < 17$, temos uma transmissão quase livre de erros. Portanto, para aplicações que requerem uma alta quantidade de usuários simultâneos, a proposta de decodificação turbo q-ária é indicada.

Na figura 5.14 é ilustrado o desempenho da proposta de decodificação turbo q-ária em um sistema FFH-CDMA em função da eficiência espectral η. Observe que a inclinação da curva da decodificação turbo q-ária, é a que mais se assemelha à curva da taxa de corte do canal. Porém, a decodificação turbo q-ária não apresenta uma boa eficiência espectral devido a baixa taxa do código produto utilizado, quando

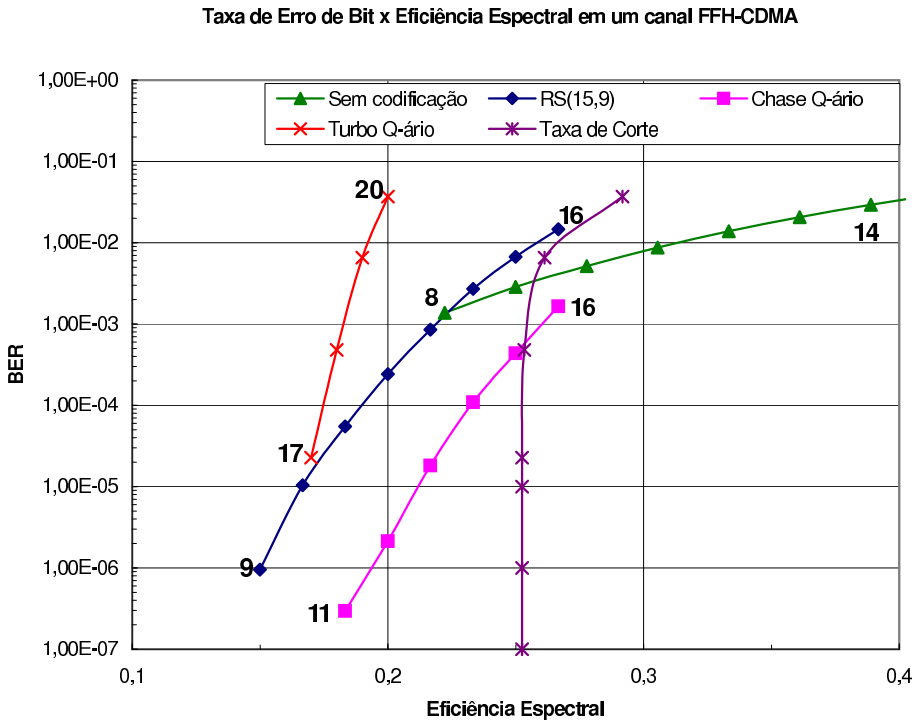

Fig. 5.14: Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando a eficiência espectral, com $M = 16$, $L = 9$ e $P_a = (225, 81, 49)$.

comparada às outras curvas. Vamos fazer uma comparação com os resultados de [3]. Apesar de o código $P_1(576, 144, 64)$, utilizado em [3], possuir uma taxa $r = 0,25$, ou seja, $\approx 30\%$ menor que a de P_a, considerando uma taxa de erro de 10^{-3}, a curva da decodificação turbo q-ária se distancia de sua taxa de corte de canal em $\Delta\eta \approx 0,07$, que é $\approx 77\%$ menor que a obtida em [3] ($\Delta\eta \approx 0,3$).

O segundo sistema FFH-CDMA analisado utiliza uma modulação 32-FSK, $L = 6$ *chips* por padrão de salto em freqüência, $RSR = 25dB$, um código produto composto de dois códigos de Reed-Solomon RS(28,20,9) em $GF(32)$ e 8 iterações no decodificador turbo. O código produto resultante $P_b(784, 400, 81)$ possui em taxa $r \approx 0,5102$. Na figura 5.15, é ilustrado o desempenho da proposta de decodificação turbo q-ária em um sistema FFH-CDMA em função do número de usuários simultâneos U. Observe que, devido a alta capacidade de correção da decodificação turbo, a curva de desempenho se aproxima mais da taxa de corte do canal FFH-CDMA.

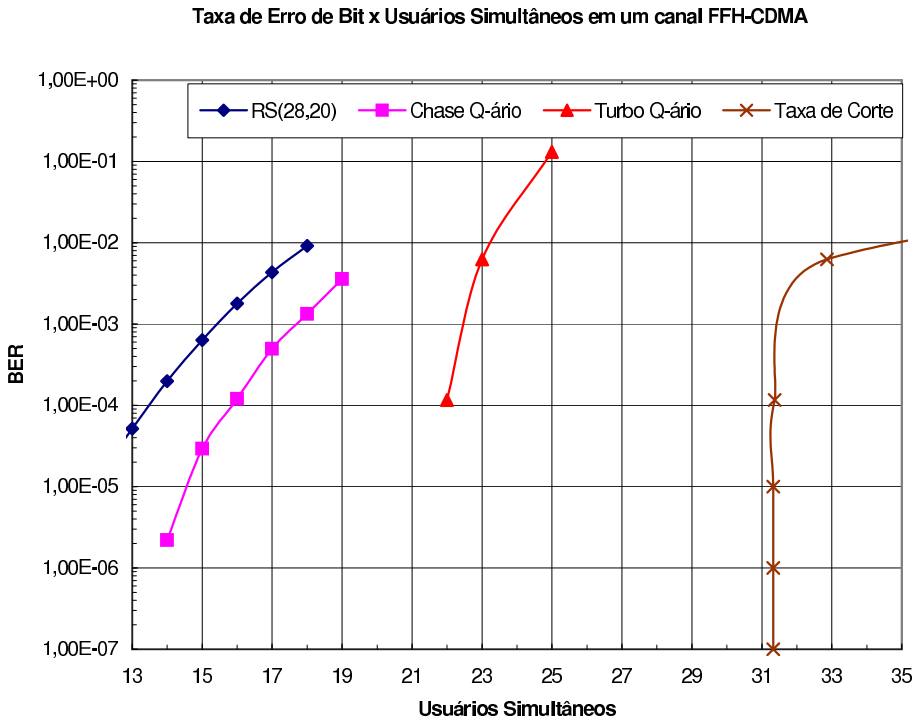

Fig. 5.15: Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando o número de usuários simultâneos, com $M = 32$, $L = 6$ e $P_b = (784, 400, 81)$.

Considerando a curva da decodificação de Chase q-ária, obtemos uma transmissão quase livre de erros para $U < 14$. No caso da decodificação turbo q-ária, atingimos esta condição já em $U < 22$, ou seja, com 8 usuários simultâneos a mais no sistema. Podemos observar que esta melhoria de desempenho, quando observada para o sistema da figura 5.13, é de 6 usuários simultâneos, ou seja, menor que o anterior. Comparando com os resultados apresentados em [20], observamos que, para uma $BER = 10^{-4}$ o decodificador turbo q-ário suporta $U = 22$ usuários simultâneos, superando a quantidade obtida em [20] que é $U = 15$.

Na figura 5.16 é ilustrado o desempenho da proposta de decodificação turbo q-ária em um sistema FFH-CDMA, com $M = 32$, $L = 6$ e $P_b = (784, 400, 81)$, em função da eficiência espectral η. Diferentemente do desempenho apresentado para o sistema da figura 5.14, a decodificação turbo q-ária apresenta uma eficiência espectral melhor que da decodificação de Chase q-ária. Esta melhoria ocorre devido às alterações

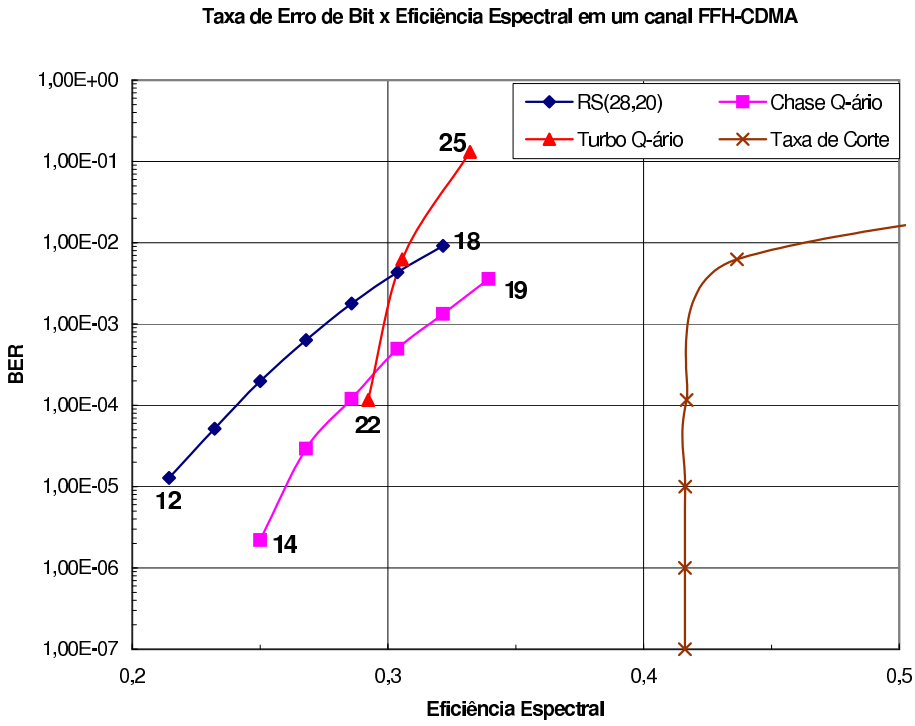

Fig. 5.16: Desempenho da proposta de decodificação turbo q-ária em um canal FFH-CDMA, considerando a eficiência espectral, com $M = 32$, $L = 6$ e $P_b = (784, 400, 81)$.

favoráveis dos parâmetros M, L e K do sistema FFH-CDMA, bem como à menor taxa do código $P_b = (784, 400, 81)$ utilizado. Vamos fazer uma comparação com os resultados de [3]. Observe que o código $P_2(400, 196, 16)$, utilizado em [3], possui uma taxa $r = 0,49$, ou seja, muito próxima da taxa de P_b ($r = 0,51$). Porém, o decodificador turbo q-ário com o código P_b fornece uma transmissão quase livre de erros a uma eficiência espectral $\eta \approx 0,3$, já o decodificador turbo binário de [3], alcança esta condição apenas em $\eta \approx 0,2$. Comparando agora com os resultados de [20], vemos que, para uma $BER = 10^{-4}$, o decodificador turbo q-ário gera uma eficiência espectral $\eta \approx 0,3$, superando os resultados de [20] que apresenta uma eficiência espectral $\eta \approx 0,25$ para a mesma taxa de erro de bit. Estes resultados mostram que a decodificação turbo q-ária apresenta um desempenho mais próximo da taxa de corte do sistema FFH-CDMA.

Capítulo 6

Conclusões

Nesta dissertação, foram apresentados estudos sobre o desenvolvimento e aplicação de métodos para a decodificação suave de códigos q-ários, bem como para a decodificação turbo q-ária com a implementação do critério SISO. As propostas apresentadas abordaram sistemas de comunicações que utilizam o modelo gaussiano como canal de transmissão para avaliação da sua aplicabilidade, porém o principal objetivo das propostas foi sua implementação em canais CDMA de salto rápido em freqüência (FFH-CDMA).

A primeira contribuição da dissertação foi realizar uma generalização dos algoritmos de decodificação suave de Chase [12], utilizados em códigos binários, a fim de suportar símbolos q-ários com elementos pertencentes a $GF(q > 2)$. Para alcançar este resultado, foram realizadas adaptações nos seguintes tópicos do método original de Chase: a) no cálculo do Logaritmo da Razão de Verossimilhança (da sigla inglesa, LLR) que representa uma medida de confiabilidade dos símbolos recebidos; e b) na geração dos padrões de testes necessários para a busca da palavra-código com menor distância analógica da palavra-recebida. O último tópico necessitou maiores mudanças para a adequação aos símbolos q-ários, bem como a consideração de um compromisso entre complexidade e capacidade de correção.

A segunda contribuição apresentada neste trabalho é considerada a mais importante, porém é dependente da mencionada no parágrafo anterior. A proposta generaliza o estudo realizado por [3] sobre a decodificação turbo binária em canais FFH-CDMA, viabilizando a aplicação deste tipo de decodificação para códigos q-

ários. Os principais tópicos tratados foram: a) o cálculo da confiabilidade da decisão do decodificador para suportar símbolos q-ários; b) os parâmetros adicionais de entrada necessários para realizar a decodificação turbo q-ária; e c) o desenvolvimento de um novo procedimento iterativo para o decodificador turbo, que agora possui um novo comportamento, onde um símbolo de entrada gera mais que duas possibilidades para o símbolo de saída, o que requer um tratamento diferenciado para definir o símbolo e sua confiabilidade para a próxima iteração.

Uma das motivações para a realização deste estudo foi a eficiência dos códigos produto para aplicações que requerem altas taxas de correções. De acordo com [2], sistemas de transmissão digital que utilizam este tipo de codificação conseguem se aproximar bastante da capacidade do canal gaussiano. Uma outra motivação foi o estudo realizado por [3], onde códigos produto de bloco foram aplicados em canais não-gaussianos, especialmente o FFH-CDMA, em conjunto com um método iterativo de decodificação que faz uso do critério SISO, porém é aplicável apenas a símbolos binários. Esta limitação requer a realização de conversões bit-símbolo para adequar os bits de informação ao alfabeto do modulador do sistema FFH-CDMA. Na verdade, a confiabilidade do símbolo recebido no demodulador FFH-CDMA é que deve ser medida, porém o decodificador suave só processa símbolos binários. Desta maneira, em [18] foi proposto um método para encontrar a confiabilidade do bit recebido, que foi aplicado em [3]. Devido a esta adaptação realizada para gerar a confiabilidade do bit, levantamos a possibilidade de haver alguma perda de informação neste processo, bem como uma melhoria de desempenho que poderia ser atribuída ao sistema FFH-CDMA, caso uma decodificação turbo q-ária fosse realizada.

No capítulo 3, foi explorada a decodificação suave através dos algoritmos de Chase aplicados a um canal gaussiano. A proposta para adaptação do algoritmo de Chase original a códigos q-ários foi apresentada. De acordo com os resultados ilustrados nas figuras 3.9, 3.10 e 3.11, observamos que o método de decodificação adaptado para símbolos q-ários supera em desempenho o método original de Chase para os três códigos Reed-Solomon utilizados, sem incrementar a complexidade da decodificação. Também pudemos observar que a melhoria de desempenho se torna mais significativa quanto maior for o alfabeto q-ário utilizado. Esta observação é

justificada pela queda de desempenho que o método original de decodificação de Chase sofre ao trabalhar com símbolos q-ários, pois conversões símbolo-bit e bit-símbolo são utilizadas para adequar o método original à decodificação de palavras-código q-árias.

No capítulo 4, utilizamos a proposta de decodificação suave q-ária em um sistema FFH-CDMA. Foram realizadas algumas adaptações na proposta apresentada no capítulo 3, a fim de adequá-la ao canal FFH-CDMA. De acordo com os resultados apresentados no capítulo, podemos afirmar que a utilização do método proposto de decodificação q-ária suave também gera uma considerável melhoria de desempenho para o sistema FFH-CDMA. Além disso, podemos mencionar que o ganho de desempenho, em relação ao método algébrico, é mais significativa para o código com maior alfabeto q-ário.

Desenvolvidos os métodos para a decodificação suave q-ária e suas adaptações ao canal FFH-CDMA, no capítulo 5, foram propostos alguns procedimentos para tornar possível a decodificação turbo q-ária. A principal contribuição do capítulo foi o novo método de realimentação do decodificador iterativo para suportar o processamento de símbolos q-ários. De acordo com os resultados apresentados, observamos que a aplicação da decodificação q-ária aumenta consideravelmente o número máximo de usuários simultâneos no sistema FFH-CDMA. Porém, a taxa do código produto utilizado pode comprometer a eficiência espectral do sistema. A utilização de códigos q-ários em $GF(q^n \geq 32)$ é recomendada para que a eficiência espectral não seja comprometida.

Como perspectiva de trabalhos futuros, sugerimos um estudo no desenvolvimento de algoritmos de busca de códigos q-ários ótimos para a utilização como códigos produtos de bloco, juntamente com a decodificação turbo q-ária, em sistemas FFH-CDMA. Estes códigos q-ários ótimos devem possuir características que considerem um compromisso entre a capacidade de correção e a eficiência espectral do sistema FFH-CDMA. Com isso, poderemos utilizar códigos mais eficientes que possibilitem uma maior aproximação da taxa de corte do canal FFH-CDMA.

Referências Bibliográficas

[1] C. Berrou, A. Glavieux e P. Thitimajshima. Near Shannon Limit Error-Correcting Coding and Decoding:Turbo-Codes(1). *IEEE Int. Conf. Communications ICC-93*, vol. 2/3:pp. 1064–1071, May 1993.

[2] R. Pyndiah. Near Optimum Decoding of Product Codes: Block Turbo Codes. *IEEE Trans. on Communications*, vol. 2(no. 8):pp. 1003–1010, August 1998.

[3] D. C. da Cunha. *Decodificação Iterativa (Turbo) de Códigos Produto em Canais Não-Gaussianos*. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e Computação, UNICAMP, Campinas-SP, Brasil, 2003.

[4] M. D. Yacoub. *Foundations of Mobile Radio Engineering*. CRC Press, 1993.

[5] R. Ash. *Information Theory*. J. Wiley, 1965.

[6] S. Haykin. *Digital Communications*. J. Wiley, 1988.

[7] I.S. Reed e G. Solomon. Polynomial Codes over Certain Finite Fields. *SIAM J. Appli. Math*, vol. 8:pp. 300–304, 1960.

[8] R.C. Singleton. Maximum distance Q-nary codes. *IEEE Trans. Inform. Theory*, vol. IT-10(no. 2):pp. 116–118, Abril 1964.

[9] R.A. Silverman e M. Balser. Coding for Constant-Data-Rate Systems - Part I. A New Error-Correcting Code. *Proc. IRE*, vol. 42:pp. 1428–1435, September 1954.

[10] M. Balser e R.A. Silverman. Coding for Constant-Data-Rate Systems - Part II. Multiple-Error-Correcting Codes. *Proc. IRE*, vol. 43:pp.728–733, June 1955.

[11] G. D. Forney, Jr. Generalized Minimum Distance Decoding. *IEEE Trans. Inform. Theory*, IT-12(no. 2):pp. 125–131, April 1966.

[12] D. Chase. A Class of Algorithms for Decoding Block Codes with Channel Measurement Information. *IEEE Trans. Inform. Theory*, IT-18(no. 1):pp. 170–182, January 1972.

[13] E. R. Berlekamp. *Algebraic Coding Theory*. Mc-Graw Hill, 1968.

[14] R. L. Pickholtz, D. L. Schilling e L. B. Milstein. Theory of Spread-Spectrum Communications - A Tutorial. *IEEE Trans. Commun. Technol.*, COM-30(no. 5):pp. 855–884, May 1982.

[15] G. A. de Deus Jr. e J. Portugheis. Sistemas FFH-CDMA Codificados - Parte II : Critério de Projeto. *XIX Simpósio Brasileiro de Telecomunicações*, Gramado-RS, Brasil, Setembro 1999.

[16] O. Yue. Maximum Likelihood Combining for Noncoherente and Differentially Coherent Frequency-Hopping Multiple Access Systems. *IEEE Trans. Inform. Theory*, IT-28(no. 4):pp. 631–639, July 1982.

[17] J. G. Proakis. *Digital Communications*, volume 3th Edition. McGraw-Hill, 1995.

[18] G. A. de Deus Jr. *Sistemas FFH-CDMA Codificados*. Tese de Doutorado, Faculdade de Engenharia Elétrica e Computação, UNICAMP, Campinas-SP, Brasil, 2002.

[19] B. Frey. *Graphical Models for Machine Learning and Digital Communication*. The MIT Press, 1998.

[20] U-C. Fiebig e P. Robertson. Soft-Decision and Erasure Decoding in Fast Frequency-Hopping Systems with Convolutional, Turbo, and Reed-Solomon Codes. *IEEE Trans. on Communications*, vol. 47(no. 11):pp. 1646–1654, November 1999.

[21] C. Berrou e A. Glavieux. Near Optimum Error Correcting Coding and Decoding: Turbo-Codes. *IEEE Trans. on Communications*, vol. 44:pp. 1261–1271, October 1996.

[22] R. Pyndiah, A. Glavieux, A. Picart e S. Jacq. Near Optimum Decoding of Product Codes. *IEEE GLOBECOM 94 Conf.*, vol. 1/3:pp. 339–343, Nov-Dec 1994.

[23] G. D. Forney. *Concatenated Codes*. Ph.D. Thesis, Cambridge: Massachusetts Institute of Technology, 1966.

[24] F. H. Huang. *Evaluation of Soft Output Decoding for Turbo Codes*. Tese de Mestrado, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, 1997.

[25] R. Pyndiah. Iterative Decoding of Product Codes: Block Turbo Codes. *IEEE Int. Symposium on Turbo Codes and Related Topics*, vol. 1/1:pp. 71–79, September 1997.

[26] P. Elias. Error-free coding. *IRE Trans. Inform. Theory*, vol. IT-4:pp. 29–37, September 1954.

[27] B. Sklar. *Digital Communications: Fundamentals and Applications*. Prentice Hall PTR, 2nd Edition, 2000.

[28] G. Colavolpe, G. Ferrari e R. Raheli. Extrinsic Information in Iterative Decoding: A Unified View. *IEEE Trans. on Communications*, vol. 49(no. 12):pp. 2088–2094, December 2001.