

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

GENERALIZAÇÃO DO DECODIFICADOR MULTISTÁGIO:
PROPOSIÇÃO E ANÁLISE DE DESEMPENHO

BARTOLOMEU FERREIRA UCHÔA FILHO

Orientador : PROF. DR. REGINALDO PALAZZO JR.†

Este exemplar corresponde à redação final da tese
defendida por Bartolomeu Ferreira

Uchôa Filho e aprovada pela Comissão

Julgadora em 6 / 7 / 92.

Reginaldo Palazzo Jr.
Orientador

Tese de Mestrado apresentada
à Faculdade de Engenharia Elé-
trica da Universidade Estadual
de Campinas, como parte dos re-
quisitos exigidos para a ob-
tenção do título de *MESTRE EM*
ENGENHARIA ELÉTRICA

Julho/1992

UNIGAMP
BIBLIOTECA CENTRAL

Dedico esta tese aos meus pais,
BARTOLOMEU e YEDA,
e à minha noiva,
FAL.

Quando Hilbert nos diz: "Consideremos três classes de entes que chamaremos pontos, retas e planos", dotados de propriedades tais como: "por dois pontos pode-se fazer passar uma reta e uma só", sabemos muito bem que Hilbert não teria pensado em tais "entes" se Euclides não tivesse vivido antes dele.

E. Borel

RESUMO

Nesta tese propomos uma estratégia de decodificação multiestágio da qual o esquema de Imai e Hirakawa é um caso particular. O decodificador multiestágio ótimo nesta classe é obtido otimizando os custos dos estimadores associados a cada estágio no processo de decodificação. Também fornecemos um procedimento geral para se obter um limitante superior para a probabilidade de erro de bit, que leva em consideração os efeitos da informação lateral errônea, para as constelações de sinais 2^{τ} -PSK e 4^{τ} -QAM. Finalmente, uma comparação com os decodificadores multiestágio de Imai-Hirakawa e Kofman *et. al.* é apresentada onde mostramos que a estratégia de decodificação proposta apresenta um melhor desempenho que as anteriores.

ABSTRACT

In this thesis we propose a multistage decoding strategy from which the Imai-Hirakawa scheme is a particular case. The optimum multistage decoder in this class is obtained by the optimization of the estimator costs associated with each stage in the decoding process. We also provide a general procedure to derive an upper-bound on the bit error probability, which takes into account wrong side information effects, both for the 2^{τ} -PSK and for the 4^{τ} -QAM signal constellations. Finally, a comparison with both Imai-Hirakawa and Kofman *et. al.* multistage decoders is presented where we show that the proposed decoding strategy outperforms the previous ones.

AGRADECIMENTOS

Ao Prof. Dr. Reginaldo Palazzo Jr., meu orientador, pelo invulgar apoio, incentivo, dedicação e disponibilidade com que contei durante todo o curso de Mestrado; por tudo que representa e fez pela minha realização profissional, onde seus atos foram muito além de suas obrigações como orientador; pessoa que considero um verdadeiro exemplo de profissional e de ser humano;

Aos Professores: Borelli, João Bosco, Akebo, Fábio, Amauri, Rege, Celso e Dalton da FEE/UNICAMP, pelos ensinamentos que muito contribuíram para a realização deste trabalho;

Aos Professores Ricardo e Márcia Campello, e Waldemar C. da Rocha Jr., da UFPE, pelos ensinamentos, apoio e incentivo que me fizeram derivar para o fascinante mundo das Comunicações;

À Elza Aoki, pelo apoio, dedicação e sugestões na confecção da tese e ao exímio desenhista Sr. Airton Ramos;

Aos responsáveis pelo LCA (Laboratório de Computação e Automação), pela disponibilidade dos recursos computacionais;

Aos meus amigos e colegas da FEE, que proporcionaram um ambiente de estudo alegre e descontraído, fundamental para o bom andamento de qualquer trabalho;

Ao CNPq, pelo apoio financeiro;

Aos meus familiares, especialmente meus pais e minhas irmãs, pelo apoio e incentivo à minha carreira acadêmica;

À minha noiva, pelo apoio, incentivo e prova de amor ao suportar minha ausência física no período em que estive na UNICAMP.

CONTEÚDO

CAPÍTULO 1

INTRODUÇÃO	1
1.1. INTRODUÇÃO	2
REFERÊNCIAS BIBLIOGRÁFICAS	7

CAPÍTULO 2

OS CÓDIGOS CONVOLUCIONAIS	8
2.1. INTRODUÇÃO	9
2.2. OS CÓDIGOS DE ÁRVORE	10
2.3. OS CÓDIGOS DE TRELIÇA	11
2.4. OS CÓDIGOS CONVOLUCIONAIS	15
2.4.1. O Processo de Codificação	17
2.4.2. A Representação Matricial	18
2.4.3. A Representação por Meio de Registradores de Deslocamento	20
2.4.4. O Diagrama de Estados	24
2.4.5. O Diagrama de Estados Particionado	24
2.4.6. Algumas Subclasses de Códigos Convolucionais	30
2.4.6.1. Os códigos convolucionais catastróficos	30
2.4.6.2. Os códigos convolucionais de memória unitária	32
2.4.6.3. Os códigos convolucionais puncionados	35
REFERÊNCIAS BIBLIOGRÁFICAS	40

CAPÍTULO 3

LIMITANTES DE DESEMPENHO VIA AS FUNÇÕES DE TRANSFERÊNCIA GENERALIZADAS	41
3.1. INTRODUÇÃO	42
3.2. DESCRIÇÃO DO SISTEMA GENÉRICO	43
3.3. O ALGORITMO DE VITERBI	47
3.4. OS EVENTOS DE ERRO	53
3.5. A DISTRIBUIÇÃO MÉDIA	58
3.6. OBTENÇÃO DA FUNÇÃO DE TRANSFERÊNCIA GENERALIZADA	64
3.7. CASO ESPECIAL: OS CÓDIGOS CONVOLUCIONAIS	73
3.7.1. Códigos Convolutivos sobre um Canal "BSC"	79
3.7.2. Códigos Convolutivos sobre um Canal "AWGN"	85
REFERÊNCIAS BIBLIOGRÁFICAS	89

CAPÍTULO 4

A MODULAÇÃO CODIFICADA	90
4.1. INTRODUÇÃO	91
4.2. MODULAÇÃO CODIFICADA POR TRELIÇA	93
4.2.1. Conceitos Básicos	93
4.2.2. Fator de Expansão da Constelação Via Capacidade de Canal	95
4.2.3. Mapeamento por Partição de Conjunto	96
4.2.4. Exemplo e Decodificação de Sistemas TCM	102
4.3. MODULAÇÃO CODIFICADA MULTINÍVEL	106
4.3.1. Conceitos Básicos	106
4.3.2. Decodificação Multiestágio	110
4.4. ALGUMAS CONSIDERAÇÕES SOBRE CONSTELAÇÕES DE SINAIS	113
REFERÊNCIAS BIBLIOGRÁFICAS	115

CAPÍTULO 5

A DECODIFICAÇÃO MULTISTÁGIO GENERALIZADA	118
5.1. INTRODUÇÃO	119
5.2. A ESTRATÉGIA DE DECODIFICAÇÃO PROPOSTA	120
5.3. ANÁLISE DE DESEMPENHO DO DECODIFICADOR MULTISTÁGIO GENERALIZADO	123
5.3.1. Procedimento Geral	123
5.3.2. O Caso da Constelação 8-PSK	130
5.3.3. O Caso da Constelação 2^T -PSK (Generalização)	136
5.3.4. O Caso da Constelação 4^T -QAM (Generalização)	138
5.4. EXEMPLOS E RESULTADOS OBTIDOS	150
REFERÊNCIAS BIBLIOGRÁFICAS	160

CAPÍTULO 6

CONCLUSÕES	162
6.1. COMENTÁRIOS E CONCLUSÕES	163

APÊNDICES

APÊNDICE A : O LIMITANTE DE CHERNOFF	165
REFERÊNCIAS BIBLIOGRÁFICAS	168
APÊNDICE B : PARTIÇÃO DE CONJUNTO	169

GLOSSÁRIO

A	: árvore
\underline{A}	: matriz transição de estado
\underline{A}'	: derivada de \underline{A} com relação a I
AWGN	: Additive White Gaussian Noise
\underline{a}	: rótulo de subconjunto na partição
a_{ij}	: ganho do ramo do super-estado ϵ_i ao super-estado ϵ_j
b	: número de bits de informação do código
\underline{b}_i	: vetor transição de estado partindo de ϵ_i
\underline{b}'_i	: derivada de \underline{b}_i com relação a I
BSC	: Binary Symmetric Channel
b_i	: unidade de retardo
C	: capacidade de canal (Cap. 1); subconjunto de S (Cap. 5)
C_A	: código de árvore
C_T	: código de treliça
\underline{c}_i	: vetor transição de estado convergindo para ϵ_i
\underline{c}'_i	: derivada de \underline{c}_i com relação a I
$C(\cdot)$: função característica
C_{UNG}	: código de Ungerboeck
\underline{C}_i	: bloco codificado de i-ésimo estágio
C_i	: coordenada do bloco codificado do i-ésimo estágio
$C_i^{(t)}$: coordenada do bloco codificado no instante t do i-ésimo estágio
\hat{C}_i	: coordenada do bloco codificado estimada do i-ésimo estágio
$c(z)$: custo associado ao estado de canal z

D	: profundidade da árvore
d_{free}	: distância livre (de Hamming ou Euclidiana)
$d_H(\cdot, \cdot)$: distância de Hamming
$d(\cdot, \cdot)$: medida de distorção
$D(\cdot, \cdot, \cdot)$: esperança do limitante de Chernoff
\bar{d}	: distorção média
$d_E(\cdot, \cdot)$: distância Euclidiana
$d(S)$: distância mínima de S
D_i	: decodificador do i -ésimo estágio
$D^{(i)}$: função $D(\cdot, \cdot, \cdot)$ para o i -ésimo estágio
\mathbb{D}^2	: reticulado \mathbb{D}^2
$dec(\cdot, \cdot, \cdot, \dots, \cdot)$: função de representação em base decimal
d_M^2	: distância Euclidiana quadrática mínima da matriz código
d_{min}^2	: distância Euclidiana quadrática mínima do código total
$E(R)$: função de confiabilidade
E_o	: número de estados do código originador do código puncionado
E_p	: número de estados do código puncionado
$E(\cdot)$: esperança matemática
$E(\cdot \{s_k\})$: esperança matemática condicionada a $\{s_k\}$
E	: alfabeto de entrada do canal
E_i	: codificador do i -ésimo estágio
$erfc$: função erro complementar
$f(\cdot, \cdot)$: função de saída
$GF(q)$: Galois Field
G	: matriz geradora do código convolucional
$G_i(\cdot)$: submatriz de G

$g_{i,j}^{(k)}(\cdot)$: elementos de $G_i(\cdot)$
$g(\cdot, \cdot)$: relação de estado
$G(\cdot, \cdot)$: relação de super-estado
G_a	: ganho assintótico
$h(\cdot, \cdot)$: função do canal
$H(a)$: subconjunto da partição
i.i.d	: independentes e identicamente distribuído
I	: informação binária
I_{-i}	: bloco componente do i-ésimo estágio
I_i	: coordenada do bloco componente do i-ésimo estágio
\underline{I}	: matriz identidade (Cap. 3); bloco de informação binária (Cap. 4)
\hat{I}_i	: coordenada do bloco componente estimada do i-ésimo estágio
K	: comprimento total de restrição
k_i	: número de bits de informação do i-ésimo código componente
ℓ	: comprimento do bloco do código puncionado
M	: número de sinais no espaço Euclidiano
m	: ordem de memória do codificador convolucional
m_i	: número de células no registrador de deslocamento da i-ésima entrada de codificador convolucional
$m(\cdot, \cdot)$: métrica
$m(\cdot, \cdot, \cdot, \cdot)$: métrica generalizada
MLD	: maximum likelihood decoding

M-PSK	: M-ary Phase Shift Keying
M-QAM	: M-ary Quadrature Amplitude Modulation
N	: dimensão do espaço de sinais
n	: comprimento da palavra código
n_k	: ruído no instante k
$N(d_{free})$: número de caminhos com distância d_{free}
n_i	: comprimento da palavra código do i -ésimo estágio
$N_1(\cdot), N_2(\cdot)$: número médio de vizinhos
n_I	: número de pontos no interior da constelação
n_v	: número de pontos no vértice da constelação
n_L	: número de pontos na lateral da constelação
n_p	: componente em fase do ruído
n_q	: componente em quadratura do ruído
P_E	: probabilidade de erro
$P(s_k)$: probabilidade do estado s_k em regime estacionário
$P((s_k) \rightarrow \{\tilde{s}_k\})$: probabilidade de evento de erro
$P(S_i \rightarrow S_j)$: probabilidade de erro no par de sinais
P_b	: probabilidade de erro de bit total
P_c	: probabilidade de erro de bit codificado total
$P_r(e)$: probabilidade de evento de erro assintótico
P_{b_i}	: probabilidade de erro de bit do i -ésimo estágio
P_{c_i}	: probabilidade de erro de bit codificado do i -ésimo estágio
P_{cs_i}	: probabilidade de erro na estimação do estado do canal
$P_\Theta(\Theta)$: função densidade de probabilidade da fase
$P_1(\cdot), P_2(\cdot)$: número médio de PW-vizinhos

q	: ordem do corpo
$q(u)$: probabilidade de ocorrer u
$Q(\cdot)$: função de probabilidade de erro Gaussiana
R	: taxa do código
R_p	: taxa do código punccionado
R_i	: taxa do código do i -ésimo estágio
\mathbb{R}^n	: espaço n -dimensional
s_k	: estado no instante k
S	: alfabeto de estados
$ S $: número de estados
$S^2 = S \times S$: alfabeto de super-estados
$S^2_{(x)}$: alfabeto dos super-estados com componentes iguais
$S^2_{(D)}$: alfabeto dos super-estados com componentes desiguais
S_k	: super-estado no instante k
S	: conjunto de sinais
$ S $: ordem de S
S_i	: sinal em S
S/H	: partição de S
$ S/H $: ordem da partição S/H
$S \equiv S^{(0)}/S^{(1)}/\dots/S^{(\Psi)}$: cadeia de partições com Ψ níveis
S_A	: constelação com os sinais rotulados com "A"
S_{AB}	: constelação com os sinais rotulados com "A" ou "B"
SNR	: Signal-to-Noise Ratio
T	: treliça
$T(\cdot, \cdot)$: função de transferência
$\underline{t}_i(\cdot)$: função de transferência para o "super-estado" intermediário ϵ_i
$\underline{t}(I)$: vetor função de transferência de estados intermediários

$(\cdot)^T$: transposição matricial
$T_1(\cdot, \cdot)$: função de transferência do 1-ésimo codificador
TCM	: Trellis Coded Modulation
u_1	: informação
u_k	: sinal de entrada no instante k
\mathcal{U}	: alfabeto de entrada
$ \mathcal{U} $: tamanho do alfabeto \mathcal{U}
$\mathcal{U}^2 = \mathcal{U} \times \mathcal{U}$: alfabeto de super-entradas
U_k	: super-entrada no instante k
UCM	: unidade de cômputo de métrica
UEC	: unidade de estado de canal
v_i	: coordenada da palavra codificada
y_k	: observação ruidosa
\mathcal{Y}	: alfabeto da observação ruidosa
Y	: sinal recebido
y_p	: componente em fase do sinal recebido
y_Q	: componente em quadratura do sinal recebido
$w_H(\cdot)$: peso de Hamming
\mathcal{W}	: alfabeto de saída do canal
x_k	: sinal de saída
\mathcal{X}	: alfabeto de saída
$ \mathcal{X} $: tamanho do alfabeto de saída
X	: sinal transmitido
x_p	: componente em fase do sinal transmitido
x_p^*	: componente em fase do sinal referência

\hat{x}_p	: componente em fase do sinal errôneo
x_q	: componente em quadratura do sinal transmitido
x'_q	: componente em quadratura do sinal referência
\hat{x}_q	: componente em quadratura do sinal errôneo
z	: estado do canal
\hat{z}	: estado do canal estimado
Z^2	: reticulado quadrático
γ	: comprimento de restrição da árvore
$\Gamma(\cdot)$: função do codificador
δ	: distância intrasubconjunto
δ_i	: distância intrasubconjunto do i-ésimo nível de partição
Δ_i	: distância intrasubconjunto da i-ésima estrutura
Λ_i	: região de decisão associada ao sinal S_i
μ	: número de ramos que divergem de cada vértice de uma árvore até uma profundidade D
ν	: número de dígitos de informação dependentes no rotula- mento
$\nu^{(t)}$: vetor binário fornecido pelas coordenadas das palavras código do i-ésimo estágio no instante t.

CAPÍTULO 1

INTRODUÇÃO

1.1. INTRODUÇÃO

A Comunicação Digital, que desde a segunda metade deste século vem recebendo uma contínua e crescente atenção, é vista indubitavelmente como a forma mais conveniente de se transmitir informação de um ponto (transmissor) para outro (receptor), segundo um critério de confiabilidade, quando o meio de transmissão é ruidoso.

Embora a maior parte dos sinais de interesse em sistemas de comunicação seja de natureza analógica (onde amplitude e tempo variam continuamente), a conversão destes para uma forma digital, segundo um critério de fidelidade, é sempre possível.

Um diagrama geral de um sistema de comunicação digital é mostrado na Figura 1.1, onde os blocos encerrados pela linha tracejada constituem o canal discreto. A partir deste modelo dois problemas, formulados e resolvidos por Shannon [1] e que encerram a tão conhecida Teoria da Informação, surgem no sentido de determinar:

- 1) a máxima taxa na qual uma comunicação confiável sobre um canal ruidoso seja possível;
- 2) a mínima taxa na qual uma seqüência da fonte digital transmitida, sobre um canal sem ruído, possa ser fielmente reconstruída.

Neste trabalho nos limitamos ao primeiro problema, onde nos interessa que símbolos aleatoriamente selecionados de um conjunto finito sejam reproduzidos a partir de observações ruidosas destes, segundo um critério de confiabilidade. O fato dos símbolos transmitidos transportarem, ou não, o significado desejado; ou o fato do significado recebido afetar, ou não, uma possível conduta da forma desejada; não será aqui levado em consideração.

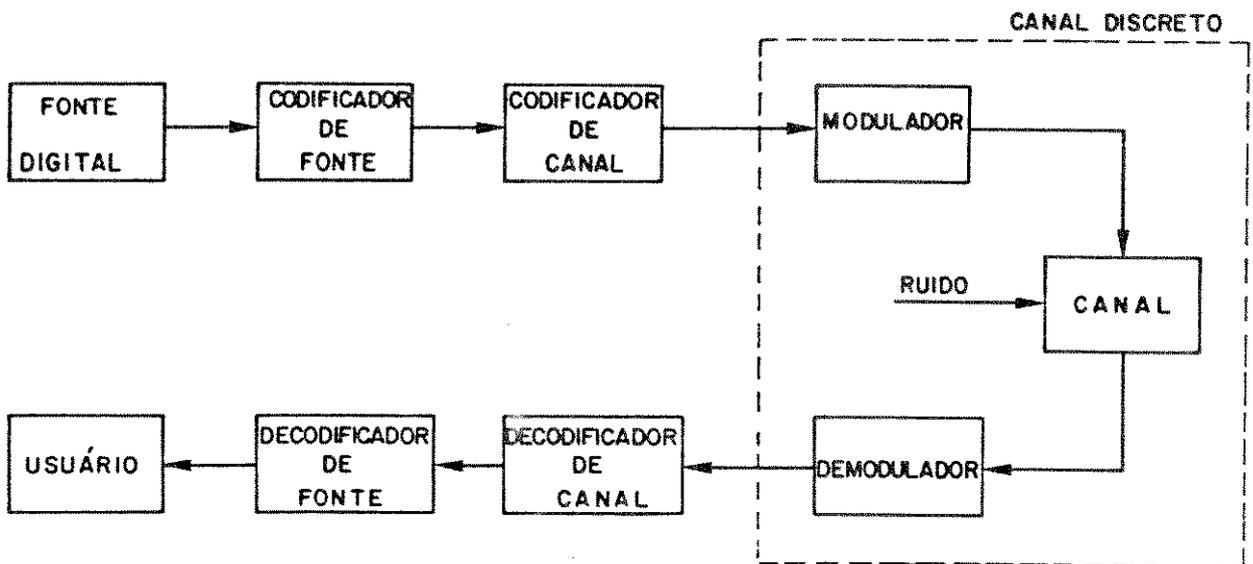


Figura 1.1 - Diagrama Geral de um Sistema de Comunicação Digital.

Nestes termos, os blocos da Figura 1.1 correspondendo ao codificador e ao decodificador de canal serão estudados. O canal discreto é considerado sem memória, que é definido como sendo um canal onde o símbolo de saída em um certo instante depende apenas do símbolo de entrada naquele instante e de nenhum outro.

Da Teoria da Informação, demonstra-se que para qualquer canal com alfabeto de entrada discreto existe um código (conjunto de sinais) com M palavras código (sinais), de dimensão N , cuja taxa é dada por $R = \frac{\log_2 M}{N}$ bits/dimensão, para o qual a probabilidade de erro com decodificação por máxima verossimilhança é limitada superiormente por

$$P_E \leq 2^{-N \cdot E(R)}, \quad (1.1)$$

onde $E(R)$ é uma função de confiabilidade que depende das características do canal e que satisfaz

$$E(R) \begin{cases} > 0 & ; \text{ se } 0 \leq R < C \\ = 0 & ; \text{ se } R \geq C \end{cases}, \quad (1.2)$$

onde C é a capacidade de canal.

Com base neste teorema, a Teoria da Codificação de Canal estuda a busca de esquemas de codificação e decodificação cada vez mais sofisticados, e em dimensões elevadas, para tornar a probabilidade de erro da Equação (1.1) tão pequena quanto se queira, condicionado a $R < C$.

Por outro lado o modulador e o demodulador, que transformam um canal contínuo em um canal discreto, também contribuem para este fim.

Tratando os processos de codificação e modulação como uma entidade única, Ungerboeck [4] conseguiu margens de ganho extremamente relevantes. A partir desta postura unificada de codificação e modulação: a Modulação Codificada, várias outras classes de códigos têm sido propostas. Entre elas, e de grande importância, figura a Modulação Codificada Multinível.

Neste trabalho, propomos uma estratégia de decodificação para Sistemas de Modulação Codificada Multinível, e apresentamos um procedimento geral para a análise de desempenho desta proposta. Sob a forma de exemplo mostramos que esta estratégia apresenta um desempenho superior ao da decodificação multi-estágio convencional [5], a qual é vista como um caso particular do esquema proposto.

Esta tese está organizada em seis capítulos, que abordam os seguintes assuntos:

CAPÍTULO 2 - Apresenta os conceitos básicos dos códigos convolucionais, uma vez que estes são utilizados no sistema considerado nesta tese;

CAPÍTULO 3 - Mostra um procedimento adotado na determinação de um limitante para o desempenho de um sistema discreto no tempo e de estado finito genérico;

CAPÍTULO 4 - Introduz os conceitos básicos e apresenta os Sistemas de Modulação Codificada;

CAPÍTULO 5 - Apresenta a estratégia de decodificação proposta com um procedimento geral para a análise de desempenho desta. Dois exemplos comparando o esquema proposto com outros existentes são também fornecidos;

CAPÍTULO 6 - Neste último capítulo, apresentamos as conclusões e alguns comentários concernentes à estratégia de decodificação proposta.

APÊNDICE A - Fornece o limitante de Chernoff utilizado no Capítulo 3.

APÊNDICE B - Mostra a Partição de conjunto de Ungerboeck aplicada aos esquemas de modulação 8-PSK e 16-QAM.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication* (University of Illinois Press, 1949).
- [2] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding*, (McGraw-Hill, 1979).
- [3] S. Haykin, *Digital Communications* (Wiley, 1988).
- [4] G. Ungerboeck, "Channel Coding with Multilevel-Phase Signals", *IEEE Trans. Information Theory*, Vol. IT-28, pp. 55-67, Jan. 1982.
- [5] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error Correcting Codes", *IEEE Trans. Information Theory*, Vol. IT-23, pp. 371-377, May 1977.

CAPÍTULO 2

OS CÓDIGOS CONVOLUCIONAIS

2.1. INTRODUÇÃO

A utilização da codificação de símbolos, através da introdução de uma redundância controlada, em sistemas de comunicação, com o propósito de detectar/corrigir erros durante a transmissão é bem conhecida. Ressaltamos da Teoria de Codificação duas classes de códigos importantes: a dos códigos de bloco e a dos códigos convolucionais, que apesar das suas dissemelhanças guardam entre si algumas intrigantes equivalências. Os códigos de bloco, por excelência, apresentam uma estrutura matemática bem definida, o que facilita enormemente os processos de geração e decodificação dos mesmos. Apesar dos esforços dispendidos para este fim, tal estrutura ainda inexistente nos códigos convolucionais. Entretanto, esta última classe se abastece de um esquema de decodificação por máxima verossimilhança: o Algoritmo de Viterbi, que sem dúvida é o responsável pela crescente adoção dos códigos convolucionais nos sistemas de comunicação.

Neste capítulo, apresentamos os códigos convolucionais uma vez que estes são utilizados neste trabalho. Entretanto, é bom salientar, não temos como fim cobrir este tema em sua totalidade, mas sim, ressaltar alguns conceitos básicos para que os conceitos desenvolvidos posteriormente fiquem plenamente entendidos.

Iniciamos com a apresentação dos Códigos de Árvore, que formam a classe mais geral de códigos, da qual os Códigos de Treliza, seguidamente apresentados são um caso especial. Finalmente, com a imposição da linearidade recaímos nos Códigos Convolucionais, sendo estes uma subclasse das classes supracitadas. Considerando os Códigos Convolucionais como o tema central deste capítulo, descrevemos o seu processo de codificação, apresentamos formas distintas, porém, equivalentes de representá-los tais como: a representação por matriz geradora, a representação por meio de registradores de deslocamento, o

diagrama de estados e o diagrama de estados particionado. Finalmente, algumas classes de códigos convolucionais, bem como algumas de suas propriedades são apresentadas. A decodificação por máxima verossimilhança, através do algoritmo de Viterbi, e os limitantes superiores das probabilidades de erro de evento, bit e bit codificado dos Códigos Convolucionais, são tratados como caso particular de um sistema de estados finito genérico no Capítulo 3.

2.2. OS CÓDIGOS DE ÁRVORE

Esta classe bem geral de códigos é fundamentada numa estrutura de árvore, daí o seu nome. As árvores formam uma classe especial da classe dos grafos, e didaticamente representam um papel fundamental para a caracterização dos Códigos Convolucionais, bem como da dependência que existe nestes de símbolos da fonte em instantes anteriores e da linearidade presente nestes como um elemento viabilizador da avaliação de desempenho.

Definição 2.2.1 [1] : Uma árvore $A = (D, m)$ μ -ária é um grafo conectado sem ciclos onde, para nossa conveniência, μ ramos divergem de cada vértice até uma profundidade D e somente um ramo diverge de cada vértice com profundidade maior ou igual a D , porém, menor ou igual a $D + m$, com D , m e μ inteiros tal que $D \geq 1$, $m \geq 0$ e $\mu \geq 2$.

É importante observar que o número de vértices terminais, aqueles

com profundidade $D + m$, numa árvore $A = (D, m)$ μ -ária é igual a μ^D .

Na Figura 2.1 é mostrada uma árvore $A = (4, 2)$ binária. (Esta figura é reutilizada em todo o capítulo e, portanto, qualquer informação nela contida ainda não definida deve ser ignorada no momento da consideração).

Definição 2.2.2 [1] : Um código de árvore $C_A = (n, R, D, m)$ μ -ário é obtido associando-se à cada ramo de uma árvore $A = (D, m)$ μ -ária uma "palavra de ramo" com n símbolos do alfabeto de entrada de um canal discreto.

Em função dos novos parâmetros temos que o número de ramos que divergem de cada vértice até uma profundidade D é igual a $\mu = 2^{nR}$.

Definição 2.2.3 [1] : O comprimento de restrição γ da árvore da Definição 2.2.2 é o número total de dígitos desde o vértice $D - 1$ até $D + m$, a saber

$$\gamma = (m + 1) \cdot n.$$

Um código de árvore $(2, 0.5, 4, 2)$ binário é mostrado na Figura 2.1.

2.3. OS CÓDIGOS DE TRELIÇA

Até o presente momento, os Códigos de Árvore foram caracterizados

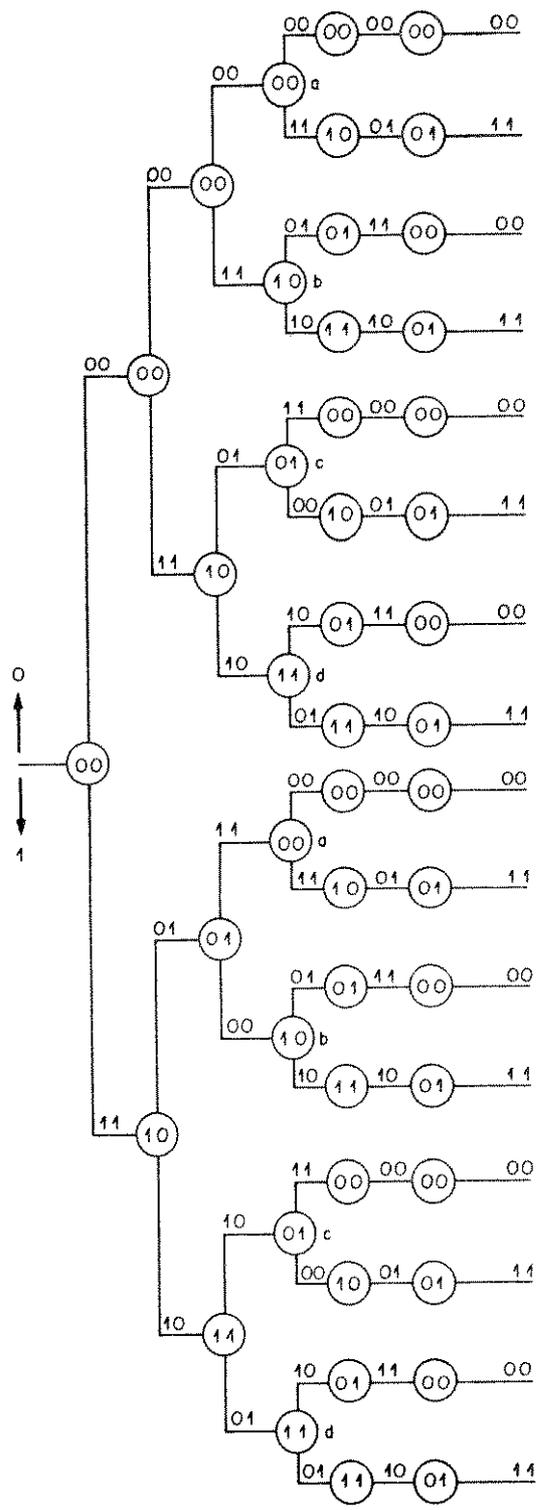


Figura 2.1 - Código de Árvore (2, 0.5, 4, 2) binário.

pela associação de uma palavra de ramo a cada ramo da árvore. O procedimento de codificação, neste caso, é completamente independente dos símbolos gerados pela fonte em instantes anteriores.

Este panorama pode ser modificado com a introdução de uma dependência entre os símbolos da fonte no processo de codificação. Para tanto, vamos associar à cada caminho na árvore uma seqüência $\left\{u_i\right\}_{i=0}^{D+m-1}$ de dígitos de informação μ -ária, onde a subseqüência $\left\{u_i\right\}_{i=0}^{D-1}$ é proveniente da fonte e $\left\{u_i\right\}_{i=D}^{D+m-1}$ é feita igual $\{0\}$, para nossa conveniência.

Com o propósito de estabelecer a referida dependência, vamos rotular cada vértice da árvore $A(D, m)$ μ -ária com os últimos ν dígitos de informação da fonte, onde ν é um número inteiro tal que $m \leq \nu \leq D + m$. Na Figura 2.1 um exemplo deste procedimento é mostrado para $\nu = m = 2$.

Um fato notável nesta nova configuração é que a árvore se torna repetitiva a partir do terceiro vértice. De fato, a partir do terceiro ramo, as seqüências de informação $(100u_3u_4\dots)$ e $(000u_3u_4\dots)$ geram a mesma subseqüência codificada. Com isto, os vértices rotulados com "a" podem ser aglutinados sem perda de informação. O mesmo ocorre com os vértices rotulados com "b", "c" e "d". A estrutura mais compacta gerada é conhecida como treliça, e é mostrada na Figura 2.2.

Definição 2.3.1 [1] : Uma treliça $T = (D, m, \nu)$ μ -ária é uma árvore $A = (D, m)$ μ -ária com os vértices enumerados pelos ν últimos dígitos de informação, quando subseqüências codificadas iguais são aglutinadas.

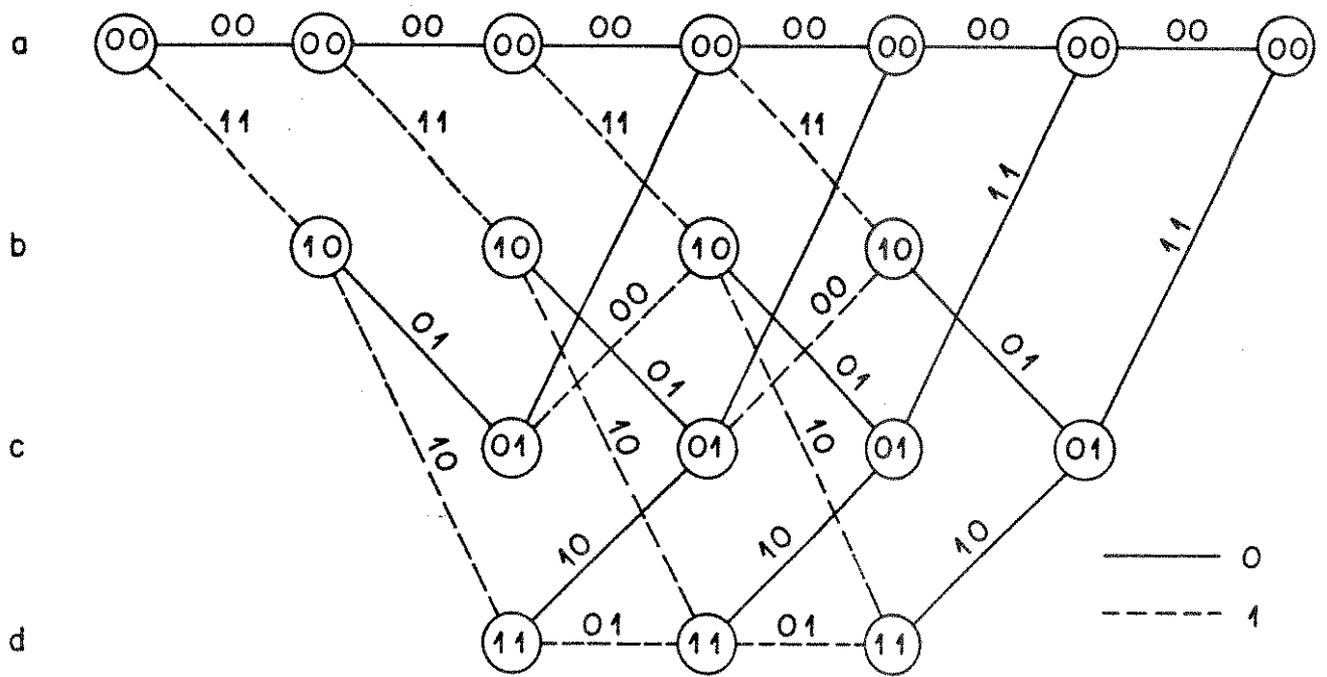


Figura 2.2 - Código de Treliça (2, 0.5, 4, 2, 2).

Naturalmente passamos a tornar dependente dos ν últimos dígitos de informação as palavras de ramo codificadas, fazendo existir uma função representando o papel do codificador. Assim sendo, a seqüência codificada $\left\{ \begin{matrix} v \\ -i \end{matrix} \right\}_{i=0}^{D+m-1}$, para uma seqüência de dígitos de informação $\left\{ u_i \right\}_{i=0}^{D-1}$ e uma seqüência toda nula $\left\{ u_i \right\}_{i=D}^{D+m-1}$, pode ser representada matematicamente por:

$$\underline{v}_i = \Gamma(\underline{u}_i, \underline{u}_{i-1}, \dots, \underline{u}_{i-\nu}, i), \quad (2.1)$$

onde i representa o tempo e o sublinhado denota um vetor.

Definição 2.3.2 [1] : Um código de treliça $C_T(n, R, D, m, \nu)$ μ -ário é obtido associando-se à cada ramo de uma treliça $T = (D, m, \nu)$ μ -ária uma "palavra de ramo" com n símbolos do alfabeto de entrada de um canal discreto, segundo uma função do tipo mostrado em (2.1).

Um código de treliça é mostrado na Figura 2.2.

2.4. OS CÓDIGOS CONVOLUCIONAIS

Da maneira que é apresentada, a função $\Gamma(\cdot)$, dada em (2.1), é bastante arbitrária. Algumas formas particulares desta função caracterizam, a partir de um código de treliça, subclasses especialmente importantes.

Quando a função $\Gamma(\cdot)$ independe da variável i , dizemos ter um código de treliça invariante no tempo. Caso contrário ele é dito ser variante no tempo. E ainda, se $\Gamma(\cdot, i) = \Gamma(\cdot, i+T)$ o código é dito ser periodicamente variante no tempo com período T . Na ocorrência da função $\Gamma(\cdot)$ ser linear, os códigos de treliça são especialmente chamados de códigos convolucionais. Deste modo, referir-se a esta última classe por códigos convolucionais lineares é cometer um pleonasma.

Definição 2.4.1 [1] : De uma maneira mais geral, o codificador convolucional (n, R, D, m, ν) com

$$\underline{v}_{-i} = \Gamma(\underline{u}_{-i}, \underline{u}_{-i-1}, \dots, \underline{u}_{-i-\nu}, i)$$

é tal que

$$\underline{u}_{-i} = (u_{1,i}, u_{2,i}, \dots, u_{b,i}) \in [GF(q)]^b$$

e

$$\underline{v}_{-i} = \left(v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(n)} \right) \in [GF(q)]^n,$$

onde $\Gamma(\cdot, i)$ são funções lineares dadas por

$$\Gamma(\cdot, i) : [GF(q)]^{(\nu+1)b} \longrightarrow [GF(q)]^n,$$

e a taxa do código é dada por $R = \frac{b}{n} \log_2 q$.

Na definição acima, cada um dos $\mu = q^b$ ramos saindo de cada vértice é representado por uma b -upla \underline{u}_{-i} , onde cada coordenada pertence ao (corpo de Galois) $GF(q)$ (do inglês: Galois field), sendo q um primo ou potência de um primo.

2.4.1. O Processo de Codificação

O codificador convolucional, da forma que é apresentado, pode ser considerado como um sistema linear discreto onde a sequência de saída $\{v_i\}$ é obtida pela convolução discreta entre a sequência de entrada $\{u_i\}$ e sua resposta impulsiva. Nestes termos, a função $\Gamma(\cdot)$ pode ser representada por:

$$v_{-i} = \sum_{j=0}^{\nu} u_{-i-j} G_j(i) = u_{-i} G_0(i) + u_{-i-1} G_1(i) + \dots + u_{-i-\nu} G_{\nu}(i), \quad (2.2)$$

onde $G_j(i)$ são matrizes $b \times n$ com elementos em $GF(q)$. Por razões óbvias, este codificador é dito ser convolucional.

Por outro lado, na estrutura de treliça podemos associar um estado a cada vértice enumerado. Como nesta estrutura o número de estados é finito, devido à aglutinação dos vértices (Definição 2.3.1), deveremos ter uma outra definição para o codificador convolucional.

Definição 2.4.1.1 [2] : Um codificador convolucional (n, R, D, m, ν) com b entradas e n saídas, sobre um corpo finito $GF(q)$, é uma máquina seqüencial de estados finitos, linear, inversível e realizável, a qual estava no estado zero num passado longínquo.

Sobre os parâmetros do codificador, estamos interessados em fazer a profundidade D tão grande quanto possível. Isto torna a decodificação de Viterbi mais eficiente, como veremos no Capítulo 3. O parâmetro ν é feito igual a m , a ordem de memória do codificador, que será definido mais adiante. Portan-

to, ficaremos com apenas um destes parâmetros.

Para o propósito deste trabalho, apenas os codificadores convolucionais binários, $q = 2$, e invariantes no tempo são considerados. Desta forma a taxa do codificador é $R = b/n$.

2.4.2. A Representação Matricial

A equação (2.2) sugere uma representação matricial para os codificadores convolucionais, tal qual ocorre com os codificadores de bloco.

A partir da equação (2.2) identificamos:

$$\begin{Bmatrix} \mathbf{v} \\ -\mathbf{i} \end{Bmatrix}_{i=0}^{D+m-1} = \begin{Bmatrix} \mathbf{u} \\ -\mathbf{i} \end{Bmatrix}_{i=0}^{D-1} \cdot \begin{bmatrix} G_0(0) & G_1(1) & \dots & G_m(m) & & & & & & & \\ & G_0(1) & \dots & G_{m-1}(m) & G_m(m+1) & & & & & & \\ & & & \vdots & \vdots & & & & & & \\ & & & \vdots & \vdots & & & & & & \\ & & & \vdots & \vdots & & & & & & \\ & & & G_0(m) & G_1(m+1) & \dots & & & & & \\ & & & & \vdots & & & & & & \\ & & & & \vdots & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & G_0(D-1) & \dots & G_m(D+m-1) \end{bmatrix} \quad (2.3).$$

A matriz em (2.3) é a matriz geradora para o codificador convolucional (n, R, D, m) , onde cada submatriz $G_\ell(i)$ tem ordem $b \times n$ e é dada por:

$$G_{\ell}^{(1)} = \begin{bmatrix} g_{1,\ell}^{(1)} & g_{1,\ell}^{(2)} & \dots & g_{1,\ell}^{(n)} \\ g_{2,\ell}^{(1)} & g_{2,\ell}^{(2)} & \dots & g_{2,\ell}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{b,\ell}^{(1)} & g_{b,\ell}^{(2)} & \dots & g_{b,\ell}^{(n)} \end{bmatrix}, \quad (2.4)$$

onde cada elemento pertence ao $GF(q)$, e os espaços em branco denotam zeros.

As equações (2.3) e (2.4) especificam plenamente o codificador convolucional. Uma vez que este é estabelecido, a definição de código convolucional segue naturalmente.

Definição 2.4.2.1 [3] : O código convolucional C_c , ou simplesmente C , gerado por um codificador convolucional (n, R, D, m) é o conjunto de todas as seqüências de saída $\left\{ \begin{matrix} v \\ -i \end{matrix} \right\}_{i=0}^{D+m-1}$, produzidas pelo conjunto de todas as possíveis seqüências de entrada $\left\{ \begin{matrix} u \\ -i \end{matrix} \right\}_{i=0}^{D-1}$.

Definição 2.4.2.2 [3] : Dois codificadores são ditos equivalentes se eles geram o mesmo código.

Exemplo 2.1 : Seja o codificador convolucional $(2, 0.5, 4, 2)$ invariante no tempo tal que: $G_0 = [1 \ 1]$, $G_1 = [0 \ 1]$ e $G_2 = [1 \ 1]$. Assim a matriz geradora semi-infinita é dada por:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ & 1 & 1 & 0 & 1 & 1 & 1 \\ & & 1 & 1 & 0 & 1 & 1 & 1 \\ & & & 1 & 1 & 0 & 1 & 1 & 1 \\ & & & & & & \ddots & & \\ & & & & & & & & \ddots & & \end{bmatrix},$$

e todas as seqüências de saída, ou seja, todas as palavras-código do código convolucional são representadas pelos caminhos na treliça da Figura 2.2.

2.4.3. A Representação por Meio de Registradores de Deslocamento

Esta representação se mostra bastante útil sob o aspecto da implementação do codificador convolucional através de circuitos digitais.

Como afirmado anteriormente, o codificador convolucional fica plenamente especificado através das submatrizes G_ℓ da matriz geradora. Deste modo, o circuito digital para este pode ser obtido por meio da sua matriz geradora. Nesta direção, observamos que a submatriz G_0 está relacionando as entradas atuais do codificador às suas saídas. A submatriz G_1 desempenha o mesmo papel considerando as entradas em um instante de tempo anterior. De uma forma mais geral, a submatriz G_ℓ relaciona as entradas em ℓ instantes de tempo anteriores às saídas do codificador. Sendo mais específico, o elemento $g_{i,\ell}^{(j)}$ da submatriz G_ℓ , relaciona a i -ésima entrada do codificador em ℓ instantes de tempo anteriores à j -ésima saída. Nestes termos, a configuração geral de um codificador convolucional é mostrada na Figura 2.3, onde as caixas denotam células dos registradores de deslocamento (do tipo FLIP-FLOP) e os somadores \oplus realizam adição módulo 2 (ou módulo q , num caso mais geral). Na Figura 2.4 é mostrado o codifi-

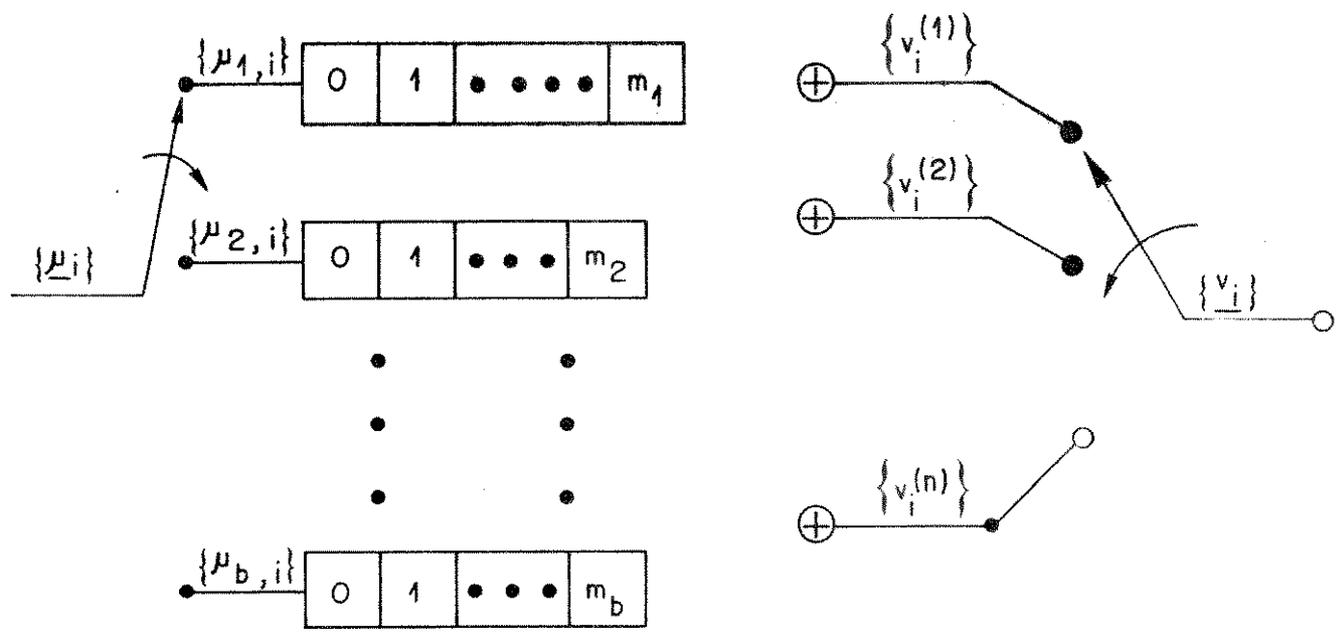


Figura 2.3 - Representação Geral de um Codificador Convolutacional.

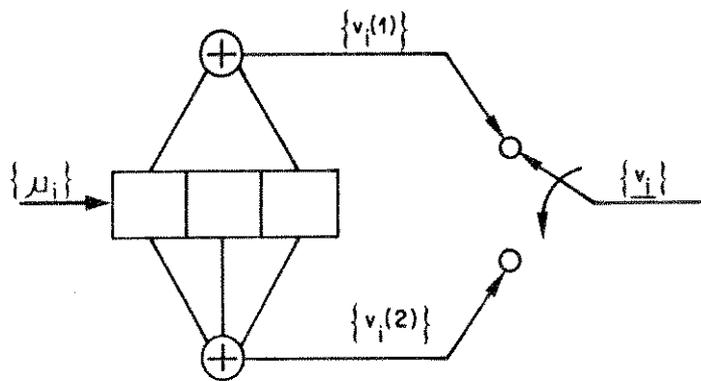


Figura 2.4 - Codificador Convolutacional (2, 0.5, 4, 2) com $G_0=[1 \ 1]$,
 $G_1=[0 \ 1]$ e $G_2[1 \ 1]$.

cador do Exemplo 2.1. A conexão da ℓ -ésima célula do registrador de deslocamento da i -ésima entrada à j -ésima saída é estabelecida ou não, quando $g_{i,\ell}^{(j)}$ é "1" ou "0", respectivamente. No caso mais geral a conexão é estabelecida com um fator multiplicativo $g_{i,\ell}^{(j)}$ em $GF(q)$. As entradas e saídas devem ser demultiplexadas e multiplexadas, respectivamente, na forma

$$\underline{u}_i = (u_{1,i}, u_{2,i}, \dots, u_{b,i})$$

e

$$\underline{v}_i = \left(v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(n)} \right).$$

Definição 2.4.3.1 [5] : A ordem de memória m do codificador convolucional (n, R, D, m) é dada por:

$$m = \max_i(m_i), \quad \text{com } 1 \leq i \leq b \quad (2.5)$$

onde m_i é o número de células do registrador de deslocamento na i -ésima entrada do codificador.

Definição 2.4.3.2 [5] : O comprimento total de memória (CTM) K do codificador convolucional (n, R, D, m) é dado por:

$$K = \sum_{i=1}^b m_i. \quad (2.6)$$

O codificador do exemplo 2.1 apresenta os seguintes valores para os parâmetros supracitados: $m = 2$ e $K = 2$.

2.4.4. O Diagrama de Estados

Na seção 2.4.1 definimos um codificador convolucional como uma máquina seqüencial de estados finito. Por conseguinte, uma representação deste por meio de um diagrama de estados é perfeitamente possível. Já é notório que os rótulos que nomeiam os estados são uma função dos m últimos símbolos de entrada. A partir de Figura 2.3 deveremos nomear os estados com o conteúdo dos registradores de deslocamento. Em conformidade com a definição 2.4.3.2, o número de estados resultante é então fornecido por 2^K , para o caso binário (ou q^K para o caso mais geral).

Segundo o que foi estabelecido, o diagrama de estados do codificador convolucional do exemplo 2.1 é mostrado na Figura 2.5. Cada ramo recebe um rótulo da forma $u_i/v_i^{(1)}v_i^{(2)}$, que são variáveis já definidas. Devemos observar que esta representação é semelhante à treliça quando transições entre estados entre dois instantes de tempo consecutivos é nesta última considerada. Correspondentemente, as palavras códigos são obtidas seguindo caminhos no diagrama, a partir de um estado inicial (que assumimos ser S_0 desde o princípio), determinados pelas seqüências de informação e pela seqüência de K elementos nulos para que o codificador retorne ao estado S_0 .

Considerando a Figura 2.5, teremos que a seqüência de informação $\left\{u_i\right\}_{i=0}^5 = \{1,0,1,1,0,0\}$ produz a palavra código $\left\{v_i\right\}_{i=0}^5 = \{11,01,00,10,10,11\}$, através do caminho $S_0 S_2 S_1 S_2 S_3 S_1 S_0$.

2.4.5. O Diagrama de Estados Particionado

Neste trabalho estamos particularmente interessados em medir o desem-

Estados: $S_0 = 00$
 $S_1 = 01$
 $S_2 = 10$
 $S_3 = 11$

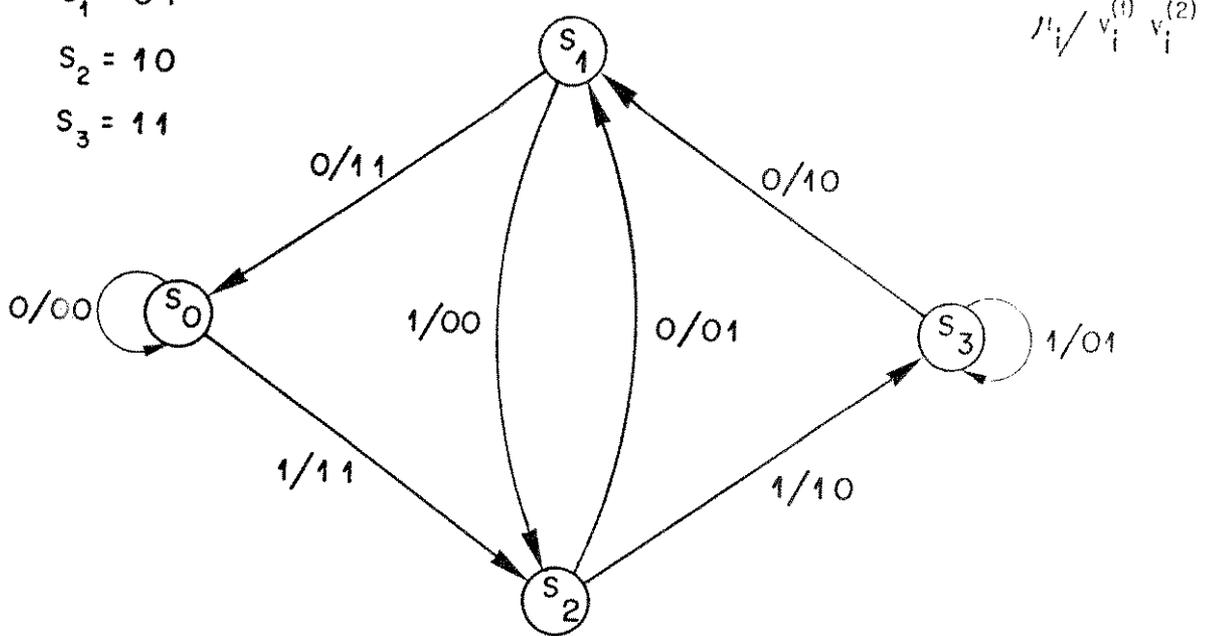


Figura 2.5 - Diagrama de Estados do Codificador Convolutacional do Exemplo 2.1.

penho de sistemas de modulação codificada multinível, que fazem uso de códigos convolucionais. Para tanto o desempenho dos códigos convolucionais utilizados devem ser individualmente avaliados.

Nesta direção, apresentaremos a seguir a definição do parâmetro d_{free} e uma representação modificada do diagrama de estado que são fundamentalmente importantes na avaliação de desempenho abordada no Capítulo 3.

Definição 2.4.4.1 : A distância livre d_{free} do código convolucional (n, R, D, m) é dada por:

$$d_{free} \triangleq \min \left\{ d_H(\underline{v}', \underline{v}'') : \underline{v}' \neq \underline{v}'' \right\}, \quad (2.7)$$

onde \underline{v}' e \underline{v}'' são as palavras código correspondendo às seqüências de informação \underline{u}' e \underline{u}'' , respectivamente; e $d_H(\underline{v}, \underline{\omega})$ é definido como o número de coordenadas em que \underline{v} e $\underline{\omega}$ diferem, o que é chamado de distância de Hamming entre \underline{v} e $\underline{\omega}$.

Como o código convolucional é linear (a soma de duas palavras código é uma palavra código) deveremos ter para $q = 2$:

$$\begin{aligned} d_{free} &= \min \left\{ \omega_H(\underline{v}' + \underline{v}'') : \underline{u}' \neq \underline{u}'' \right\} \\ &= \min \left\{ \omega_H(\underline{v}) : \underline{u} \neq \underline{0} \right\} \\ &= \min \left\{ \omega_H(\underline{u} G) : \underline{u} \neq \underline{0} \right\}, \end{aligned} \quad (2.8)$$

onde $\omega_H(\underline{v})$ é o peso de Hamming definido como o número de coordenadas não nu-

las em $\{v\}$. Como consequência da linearidade, a distância livre de um código convolucional é fornecida pelo menor peso de Hamming encontrado dentre todas as palavras código não nulas.

Como reportado na Seção 2.4.4, os caminhos percorridos no diagrama de estados fornecem as palavras código do código convolucional. Com o propósito de se obter uma estrutura representativa para os pesos de Hamming de tais palavras, podemos modificar o diagrama de estados de forma que, seguindo-se o mesmo caminho do diagrama original, para uma dada palavra código, o peso de Hamming desta é fornecido. O novo diagrama é obtido particionando-se o estado S_0 em dois, um inicial e outro final, uma vez que a transição $S_0 S_0$ não contribui para o peso de Hamming. Devemos ainda rotular cada ramo com D^i , onde i é o peso de Hamming deste ramo no diagrama original. Desta maneira, o diagrama de estados particionado para o codificador convolucional do exemplo 2.1 é mostrado na Figura 2.6.

Enquanto esta forma representativa é proveitosa para a avaliação da probabilidade de erro de bit codificado, uma vez que os pesos de Hamming fornecidos são relativos às palavras código, não possibilita a mesma análise com relação aos bits de informação. Substituir o rótulo D^i , no diagrama particionado, por $D^i I^j$, onde j é o peso de Hamming da palavra de informação referente a este ramo, propicia tal análise. O diagrama particionado aumentado para o codificador convolucional do exemplo 2.1 é mostrado na Figura 2.7.

Em se seguindo todos os caminhos do último diagrama, e multiplicando os ganhos de todos os ramos percorridos, obtemos a distribuição de pesos de todas as palavras código. Somando cada uma destas parcelas obtemos a Função de Transferência do código convolucional. Para o caso do exemplo 2.1 esta é dada por:

$$T(D,I) = D^5 I + 2D^6 I^2 + 4D^7 I^3 + \dots \quad (2.9)$$

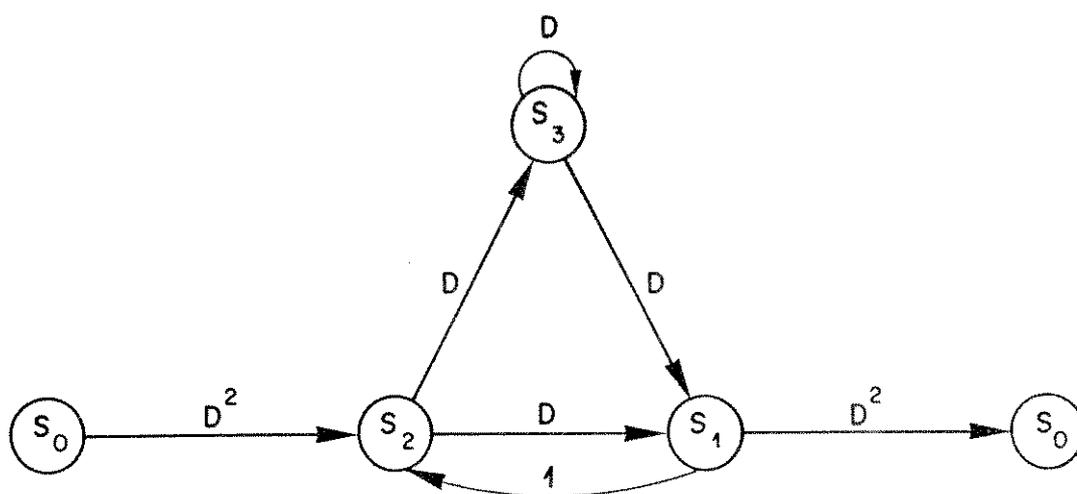


Figura 2.6 - Diagrama de Estados Particionado do Codificador Convolutcional do Exemplo 2.1.

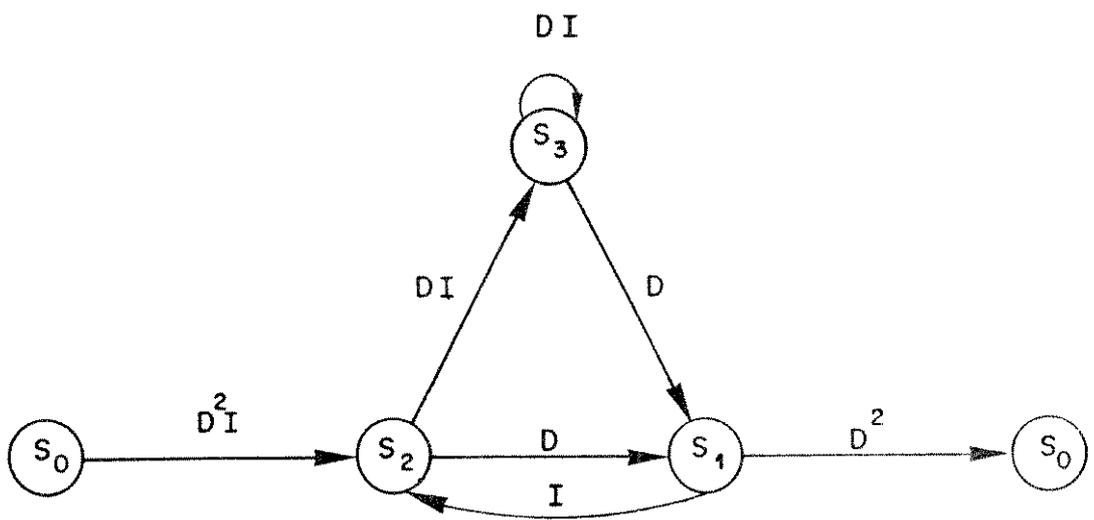


Figura 2.7 - Diagrama de Estado Particionado e Aumentado do Codificador Convolutivo do Exemplo 2.1.

Devemos interpretar a função acima como: existe um caminho com peso de Hamming igual a 5 e um bit de informação não nulo, 2 caminhos com peso de Hamming igual a 6 e 2 bits de informação não nulos, e assim por diante.

A Função de Transferência é essencial para a avaliação de desempenho de códigos convolucionais. Esta análise bem como uma maneira sistemática de obtenção da Função de Transferência são tratadas no Capítulo 3.

2.4.6. Algumas Subclasses de Códigos Convolucionais

Nesta seção, chamamos a atenção para algumas subclasses de códigos convolucionais que devem ser usadas, ou evitadas, em codificação de canal.

2.4.6.1. Os códigos convolucionais catastróficos

De acordo com a definição 2.4.1.1, um codificador convolucional é uma máquina inversível no sentido de que, existe uma máquina, denominada máquina inversa, tal que para qualquer par entrada-saída, $\{\underline{u}\} - \{\underline{v}\}$, na máquina original corresponde o par entrada-saída, $\{\underline{v}\} - \{\underline{u}\}$, na máquina inversa. A inversão do codificador é proveitosa no processo de decodificação, uma vez que leva à recuperação da seqüência de informação original (ou a melhor estimativa desta). Entretanto, divergindo da definição 2.4.1.1, nem sempre é possível conseguir o circuito inverso.

Definição 2.4.6.1.1 : Um codificador convolucional é dito ser catastrófico quando um número finito de erros introduzidos pelo canal causa um número infinito de erros de decodificação. Caso contrário, ele é dito ser não catastrófico.

Esta situação de catastroficidade é completamente indesejada para o fim de correção de erros aleatórios. Portanto, devemos evitar tal classe de codificadores convolucionais.

Teorema 2.4.6.1.1 : Um codificador é não catastrófico se e somente se ele é inversível [5].

Massey e Sain [8] estabeleceram condições de necessidade e suficiência para um codificador convolucional ser inversível.

Definição 2.4.6.1.2 : Um codificador convolucional é dito ser sistemático se, para todo instante de tempo considerado, as b entradas aparecem inalteradas entre as n saídas do codificador.

Teorema 2.4.6.1.2 : Todo codificador convolucional sistemático é não catastrófico.

Prova : Do Teorema 2.4.6.1.1, sobre a não catastroficidade, basta mostrar que um codificador convolucional sistemático é inversível. Mas isto é verdade. Basta que o circuito inverso seleccione os b símbolos de informação da entrada \underline{v} e os apresente como saída. ■

A partir do diagrama de estados, identificamos que um codificador convolucional é catastrófico se e somente se existe uma malha de peso zero além da auto-malha ao redor do estado zero (S_0). O codificador da Figura 2.6 é não catastrófico uma vez que a única malha (além da referente a S_0), aquela em torno do estado S_3 , apresenta peso 1.

2.4.6.2. Os códigos convolucionais de memória unitária

Esta classe de codificadores muito importante é possuidora de propriedades que estimulam, e ao mesmo tempo facilitam a busca de códigos convolucionais ótimos, isto é, códigos que apresentam o melhor desempenho quando comparados com outros de mesma taxa e mesmo comprimento total de memória.

Definição 2.4.6.2.1 : Um codificador convolucional de taxa $R = b/n$, é dito de memória unitária quando $m_i = 1, i=1,2,\dots,b$. O mesmo nome é dado ao código por ele gerado.

Algumas propriedades seguem diretamente da definição acima.

Propriedade 2.4.6.2.1 : Um codificador convolucional de memória unitária de taxa b/n apresenta um número de estados igual a q^b .

Prova : Segue naturalmente da equação (2.6) e da Seção 2.4.4.

Propriedade 2.4.6.2.2 : Em um codificador convolucional de memória unitária de taxa b/n , todos os q^b estados são atingíveis, a partir de qualquer estado, em qualquer instante de tempo.

Prova : Como cada entrada sofre uma e apenas uma unidade de atraso, todo conteúdo da memória é preenchido pelos b bits de entrada a cada instante de tempo. Cada um dos possíveis q^b valores de entrada conduzem o codificador para um dos possíveis q^b estados. ■

Os códigos convolucionais de memória unitária apresentam, em geral, uma distância livre maior (no pior caso igual) que a de qualquer código convolucional de mesma taxa e mesmo comprimento total de memória. Este fato sem dúvida estimula a busca de algoritmos para obtenção de códigos convolucionais de memória unitária ótimos.

Um outro fato importante que intensifica a tendência acima é o da equivalência entre um código convolucional e um código de memória unitária, que é apresentada formalmente a seguir.

Propriedade 2.4.6.2.3 : Para todo codificador convolucional de taxa b/n e ordem de memória m , existe um codificador de memória unitária equivalente de taxa mb/mn .

Prova : Devemos mostrar que é sempre possível passar a matriz geradora da forma:

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_m \\ & G_0 & G_1 & \dots & G_{m-1} & G_m \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot \end{bmatrix},$$

para a forma:

$$G = \begin{bmatrix} G'_0 & G'_1 & & & \\ & G'_0 & G'_1 & & \\ & & G'_0 & G'_1 & \\ & & & \cdot & \cdot \\ & & & & \cdot & \cdot \end{bmatrix},$$

onde G'_0 e G'_1 são submatrizes de ordens $m_b \times m_n$.

Isto é conseguido pelo seguinte agrupamento:

$$G'_0 = \begin{bmatrix} G_0 & G_1 & \dots & G_{m-1} \\ & G_0 & \dots & G_{m-2} \\ & & \cdot & \cdot \\ & & & G_0 \end{bmatrix} \quad \text{e} \quad G'_1 = \begin{bmatrix} G_m \\ G_{m-1} & G_m \\ \vdots & \vdots \\ G_1 & G_2 & \dots & G_m \end{bmatrix},$$

que são as submatrizes referentes ao codificador de memória unitária equivalente. ■

Comentário : É bom ressaltar que, se para alguma linha de entrada i do codificador convolucional original ocorrer que $m_i < m$, a submatriz G_m apresentará a i -ésima linha nula. Conseqüentemente o mesmo ocorre com a submatriz G'_1 . Isto provoca um valor $m_i = 0$ para a i -ésima linha de entrada do codificador equivalente, o que não caracteriza um codificador de memória unitária, segundo a definição 2.4.6.2.1. Quando esta condição ocorre temos um código de memória unitária parcial.

Propriedade 2.4.6.2.4 : Um codificador convolucional de taxa b/n , onde todos os estados são atingíveis, a partir de qualquer estado, e em qualquer instante de tempo, pertence à classe dos codificadores de memória unitária.

Prova : Vamos supor que para alguma linha de entrada do codificador, digamos a i -ésima, ocorra $m_i \geq 2$. Nesta situação, os b símbolos de entrada não preenchem o conteúdo da memória completamente, em um único instante de tempo. A parte não preenchida da memória recebe símbolos em instantes anteriores.

Sendo assim, apenas alguns estados são atingíveis, em um único instante de tempo, os quais são determinados também pelos símbolos em instantes anteriores. Portanto, devemos ter $m_i < 2$ para todo $i=1, \dots, b$. Como por hipótese $m_i > 0$ (a menos da situação do comentário acima) ficamos com $m_i = 1; i=1, \dots, b$. ■

2.4.6.3. Os códigos convolucionais puncionados

Na seção anterior, vimos que é possível construir um codificador de memória unitária, a partir de um codificador convolucional qualquer, com a propriedade de equivalência, onde a taxa R se mantém inalterada. Todavia, é de interesse prático obter-se, a partir de um codificador convolucional, uma família de codificadores cuja taxa varie sistematicamente. Isto é conseguido com os códigos convolucionais puncionados, que são obtidos quando removemos periodicamente uma parte dos símbolos codificados, a partir de um código convolucional. Tal procedimento é melhor explicado a seguir.

Vamos considerar um codificador convolucional de taxa $R = 1/2$, como mostrado na Figura 2.8(a). A seguir agrupamos os símbolos de saída a cada ℓ

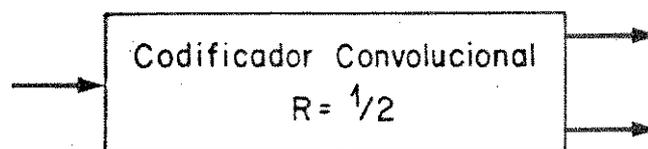
instantes de tempo, como indicado na Figura 2.8(b). O bloco engloba 2ℓ símbolos e foi gerado por ℓ símbolos de informação. Se agora removemos (puncionamos) $\ell-1$ símbolos (os marcados com X na Figura 2.8(c)), sempre nas mesmas posições em cada bloco, e deixamos de transmiti-los, geramos um código de taxa $R_P = \frac{\ell}{\ell+1}$. Este último passo é mostrado na Figura 2.8(c) e, por razões óbvias, caracteriza o código puncionado.

Podemos generalizar o codificador original para uma taxa $R = \frac{b}{b+1}$ e com ordem de memória m . Seguindo o mesmo procedimento de puncionar $\ell-1$ símbolos a cada ℓ instantes de tempo obtemos um código puncionado de taxa $R_P = \frac{\ell b}{\ell b+1}$. Cain et al. [6] e Yasuda et al. [7] derivaram alguns códigos puncionados a partir de um codificador original de taxa $R = 1/2$ ($b=1$).

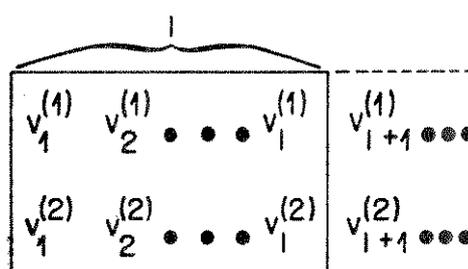
Sobre o código puncionado de taxa $R_P = \frac{\ell b}{\ell b+1}$ temos o seguinte resultado.

Teorema 2.4.6.3.1 : O código puncionado de taxa $R_P = \frac{\ell b}{\ell b+1}$, obtido a partir de um codificador convolucional de taxa $R = \frac{b}{b+1}$ e ordem de memória m , pelo procedimento acima estabelecido, pertence à classe dos códigos convolucionais de memória unitária se $\ell \geq m$.

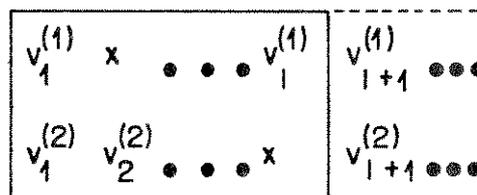
Prova : Cada ℓ entradas (ℓ instantes de tempo) de b símbolos no codificador original corresponde a uma única entrada (um único instante de tempo) de ℓb símbolos no codificador puncionado. Se fizermos $\ell \geq m$ todo o conteúdo de memória é preenchido pelo bloco, que representa um instante de tempo no codificador puncionado. Sendo assim, podemos afirmar que no codificador puncionado todos os estados são atingíveis, a partir de qualquer estado, e em qualquer instante de tempo. O resto da prova segue diretamente da propriedade 2.4.6.2.4. ■



(a)



(b)



(c)

Figura 2.8 - Obtenção do Código Puncionado a Partir de um Código Convolutacional de Taxa 1/2.

A seguir um exemplo é mostrado para ilustrar a consequência deste teorema.

Exemplo 2.2 : Vamos tomar o código convolucional de taxa $R = 1/2$ e ordem de memória $m = 3$ cujas submatrizes geradoras são dadas por [7]: $G_0 = [1 \ 1]$, $G_1 = [1 \ 1]$, $G_2 = [0 \ 1]$ e $G_3 = [1 \ 1]$. A matriz geradora é então dada por:

$$G = \begin{bmatrix} 1 \ 1 & 1 \ 1 & 0 \ 1 & 1 \ 1 & & & \\ & 1 \ 1 & 1 \ 1 & 0 \ 1 & 1 \ 1 & & \\ & & 1 \ 1 & 1 \ 1 & 0 \ 1 & 1 \ 1 & \\ & & & & \ddots & & \end{bmatrix}.$$

◦ Para $\ell < m$: Tomemos $\ell = 2 < 3$ para obtermos o código puncionado de taxa $R_p = \frac{2}{3}$. Formando blocos a cada $\ell = 2$ instantes de tempo teremos:

$$G = \begin{bmatrix} \begin{bmatrix} 1 \ 1 & 1 \ 1 \\ 0 \ 0 & 1 \ 1 \end{bmatrix} & \begin{bmatrix} 0 \ 1 & 1 \ 1 \\ 1 \ 1 & 0 \ 1 \end{bmatrix} & \begin{bmatrix} 0 \ 0 & 0 \ 0 \\ 1 \ 1 & 0 \ 0 \end{bmatrix} & & \\ & \begin{bmatrix} 1 \ 1 & 1 \ 1 \\ 0 \ 0 & 1 \ 1 \end{bmatrix} & \begin{bmatrix} 0 \ 1 & 1 \ 1 \\ 1 \ 1 & 0 \ 1 \end{bmatrix} & \begin{bmatrix} 0 \ 0 & 0 \ 0 \\ 1 \ 1 & 0 \ 0 \end{bmatrix} & \\ & & \ddots & & \end{bmatrix}$$

onde as colunas destacadas devem ser removidas. Podemos observar facilmente que esta estrutura não caracteriza um código de memória unitária.

◦ Para $\ell \geq m$: Tomemos $\ell = 3$ para obtermos o código puncionado de taxa $R_p = \frac{3}{4}$. Formando blocos a cada $\ell = 3$ instantes de tempo teremos:

$$G = \left[\begin{array}{cccc} \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right] & \left[\begin{array}{cc} 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{array} \right] & \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array} \right] & \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{array} \right] \\ & & \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right] & \left[\begin{array}{cc} 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{array} \right] & \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array} \right] & \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{array} \right] \\ & & & & & \ddots \end{array} \right],$$

onde as colunas destacadas são removidas. Percebemos facilmente que esta estrutura caracteriza um código de memória unitária com submatrizes geradoras do tipo:

$$G'_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad e \quad G'_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

É importante ressaltar que o codificador original apresenta um número de estados dado por:

$$E_0 \leq q^{bm},$$

enquanto que no codificador puncionado de memória unitária equivalente, este valor é fornecido por:

$$E_p = q^{b\ell}.$$

Em geral $E_p > E_0$, e a decodificação sobre o código original é menos complexa e deve ser adotada. Adicionalmente o conjunto codificador/decodificador do código original é único para qualquer taxa $R_p = \frac{\ell b}{\ell b + 1}$, ou seja, a estrutura da treliça é a mesma, mudando apenas a quantidade e as posições dos símbolos a serem puncionados. Conclusivamente devemos medir a complexidade de decodificação dos códigos puncionados pelo valor $E_0 \leq q^{bm}$, bem como fazer uso da treliça correspondente ao código original no processo de decodificação, onde as posições puncionadas são consideradas como "erasures".

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. Palazzo Jr., Códigos de Treliza Fixos e Variantes no Tempo, Tese de Livre-Docência, UNICAMP, 1987.
- [2] V.A. Gurian, Uma Contribuição à Forma Combinada de Codificação e Modulação Utilizando Códigos de Memória Unitária, Tese de Mestrado, UNICAMP, 1989.
- [3] G.D. Forney Jr., "Convolutional Codes I: Algebraic Structure", *IEEE Trans. Information Theory*, Vol. IT-16, pp. 720-738, Nov. 1970.
- [4] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding*, (McGraw-Hill, 1979).
- [5] S. Lin and D.J. Costello Jr., *Error Control Coding Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [6] J.B. Cain, G.L. Clark Jr. and J.M. Geist, "Punctured Convolutional Codes of Rate $(n-1)/n$ and Simplified Maximum Likelihood Decoding", *IEEE Trans. Information Theory*, Vol. IT-25, pp. 97-100, Jan., 1979.
- [7] Y. Yasuda, K. Kashiki and Y. Hirata, "High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding", *IEEE Trans. Communications*, Vol. COM-32, No. 3, pp. 315-319, Mar., 1984.
- [8] J.L. Massey and M.K. Sain, "Inverser of Linear Sequential Circuits", *IEEE Trans. Comput.*, C.17, pp. 330-337, Apr., 1968.

CAPÍTULO 3

LIMITANTES DE DESEMPENHO VIA AS FUNÇÕES DE TRANSFERÊNCIA GENERALIZADAS

3.1. INTRODUÇÃO

No capítulo que segue, dissertamos sobre o procedimento elaborado por Omura e Simon [1] na determinação de um limitante para o desempenho de um sistema discreto no tempo e de estado finito genérico, quando o algoritmo de Viterbi é utilizado para estimar a sequência de saída do sistema, depois que esta passa por um canal ruidoso. O sistema genérico a ser considerado deve gerar uma sequência de saída, segundo um modelo de estado finito, com sinais que podem apresentar diversas formas (ex.: n -uplas q -árias, senoides, etc.). Na recepção utilizamos uma métrica (para o algoritmo de Viterbi) e adotamos uma medida de distorção genéricas, as quais também podem assumir várias formas particulares, segundo a natureza do sinal e o tipo de medida com relação a qual se deseja avaliar o desempenho.

Embora esta análise seja demasiadamente ampla para o nosso propósito particular, ela permite um entendimento mais profundo da avaliação de desempenho dos códigos convolucionais, bem como realça a simplificação conseguida pela presença da linearidade nestes.

Iniciamos com a descrição do sistema, seguida da apresentação do algoritmo de Viterbi. Caracterizamos os eventos de erro e a distorção a eles associada. Seguidamente avaliamos a distorção média como uma medida para o desempenho do sistema, onde identificamos a Função de Transferência Generalizada. Um procedimento sistemático para a obtenção desta última é então apresentado, quando definimos os "super estados". Finalmente toda a análise é particularizada para o caso dos códigos convolucionais, que é ilustrado com um exemplo.

3.2. DESCRIÇÃO DO SISTEMA GENÉRICO

O sistema discreto no tempo e de estado finito que vamos considerar é mostrado na Figura 3.1. Vamos assumir que os sinais de entrada $\{u_k\}$ são discretos no tempo, estatisticamente independentes e identicamente distribuídos (i.i.d.), o que caracteriza uma fonte sem memória, cuja estatística é dada por

$$q(u), \quad u \in \mathcal{U}, \quad (3.1)$$

onde \mathcal{U} é um alfabeto finito de tamanho $|\mathcal{U}|$.

Definindo s_k como o estado do sistema no instante k , vamos considerar a seguinte relação de estado:

$$s_{k+1} = g(s_k, u_k), \quad (3.2)$$

onde s_k assume valores em um alfabeto finito S de tamanho $|S|$. Portanto, o valor $|S|$ especifica a quantidade de estados, enquanto $|\mathcal{U}|$ ramos divergem de cada estado.

Uma restrição com relação a função $g(\cdot, \cdot)$ deve ser imposta, no sentido de não permitir que haja transições paralelas entre estados, ou seja, deve haver no máximo uma possibilidade para a transição de um estado $s_k = s$ para qualquer estado $s_{k+1} = s'$. Isto pode ser matematicamente caracterizado pela condição abaixo:

$$\text{"Se } u \neq u', \text{ então } g(s, u) \neq g(s, u')\text{"},$$

com $s \in S$ e $u, u' \in \mathcal{U}$.

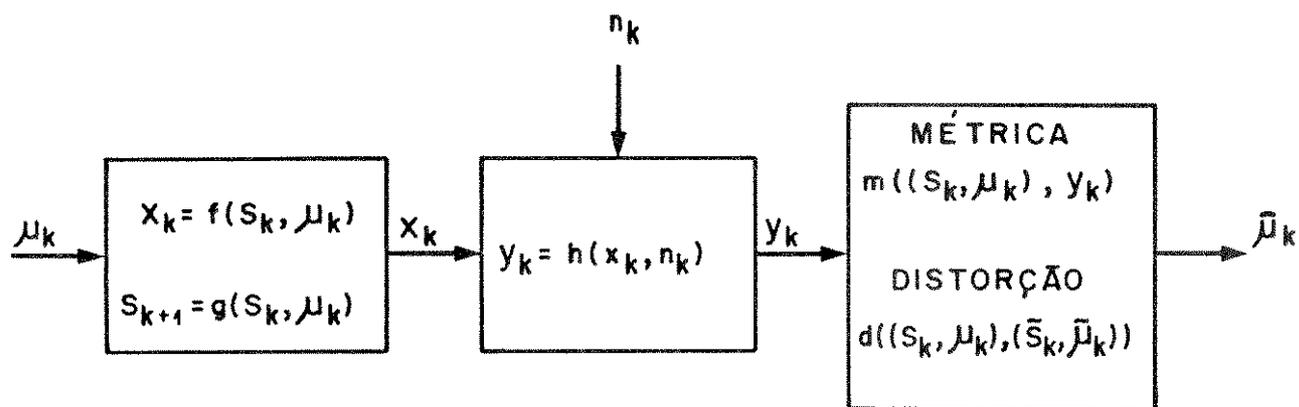


Figura 3.1 - Diagrama em Blocos do Sistema, Canal e Receptor.

Os sinais de saída $\{x_k\}$, segundo o modelo de estado finito, são descritos de acordo com a relação:

$$x_k = f(s_k, u_k), \quad (3.3)$$

e devem assumir valores em um alfabeto X de tamanho $|X|$.

O alfabeto X é feito coincidir com o alfabeto de entrada de um canal discreto (ou com uma extensão deste) cuja saída y_k é descrita por:

$$y_k = h(x_k, n_k), \quad (3.4)$$

onde $\{n_k\}$ são variáveis aleatórias i.i.d. que podem apresentar valores contínuos ou discretos.

Na recepção, o Algoritmo de Viterbi é empregado para estimar a sequência de estados $\{s_k\}$, ou equivalentemente a sequência de entrada $\{u_k\}$, a partir das observações $\{y_k\}$. Para tanto, o algoritmo faz uso de uma métrica

$$m(x_k, y_k) = m\left((s_k, u_k), y_k\right), \quad (3.5)$$

a qual deve ser avaliada para cada ramo do diagrama de treliça, como é descrito na próxima seção. Conforme a generalidade do problema a métrica em (3.5) pode assumir diversas formas. Tipicamente encontramos na literatura métricas tais como:

a) *Máxima Verossimilhança* :

$$\begin{aligned} m\left((s_k, u_k), y_k\right) &= \log_e p\left(y_k \mid (s_k, u_k)\right) \\ &= \log_e p(y_k \mid x_k) \end{aligned} \quad (3.5a)$$

b) *Máxima A Posteriori* :

$$\begin{aligned} m\left((s_k, u_k), y_k\right) &= \log_e \left[p\left(y_k \mid (s_k, u_k)\right) q(u_k) \right] \\ &= \log_e p(y_k \mid x_k) + \log_e q(u_k) \end{aligned} \quad (3.5b)$$

c) *Mínimo Erro Quadrático Médio* :

$$m\left((s_k, u_k), y_k\right) = -(y_k - x_k)^2 \quad (3.5c)$$

Com o intuito de medir o desempenho do sistema, definimos uma medida de distorção como uma função não negativa que satisfaz

$$d\left((s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right) = 0 \quad ; \quad \text{se } u_k = \tilde{u}_k \text{ e } s_k = \tilde{s}_k. \quad (3.6)$$

Tipicamente, são utilizadas medidas de distorção tais como:

a) *Distorção de Erro* :

$$d\left((s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right) = \begin{cases} 1 & ; \quad u_k \neq \tilde{u}_k \\ 0 & \quad u_k = \tilde{u}_k \end{cases} \quad (3.6a)$$

b) *Erro Quadrático Médio* :

$$d\left((s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right) = \alpha(\tilde{s}_k - s_k)^2 + \beta(\tilde{u}_k - u_k)^2; \quad \alpha, \beta \geq 0. \quad (3.6b)$$

A escolha da métrica e da distorção depende da natureza dos sinais e da medida com relação a qual se deseja avaliar o desempenho. Para os códigos convolucionais, a escolha da métrica de máxima verossimilhança e da distorção de erro fornece a probabilidade de erro de bit.

3.3. O ALGORITMO DE VITERBI

Em 1967, Viterbi propôs um esquema de decodificação para os códigos convolucionais. Estudando esta proposta, Omura a reconheceu como uma técnica de Pesquisa Operacional conhecida como Programação Dinâmica. Posteriormente, Forney determinou que se tratava de um decodificador de máxima verossimilhança. Nesta seção, discorreremos sobre esta proposta, conhecida como o algoritmo de Viterbi, numa abordagem tão geral quanto a que vimos adotando até o momento. Na verdade, o algoritmo de Viterbi é um algoritmo ótimo para qualquer métrica, segundo o modelo de sistema que vimos adotando. Ele é particularmente um estimador de máxima verossimilhança para o caso dos códigos convolucionais, ou qualquer outro sistema que adote a métrica de máxima verossimilhança.

A representação mais conveniente para processar o algoritmo de Viterbi é o diagrama de treliça. Um exemplo desta representação para $|S| = 4$ e $|U| = 3$ é encontrado na Figura 3.2, onde 3 ramos divergem de cada um dos 4 estados.

Vamos admitir que o sistema começa a operar no instante $t = 0$, partindo de um estado inicial s_0 . Semelhantemente ao que ocorre com os códigos convolucionais (Capítulo 2), todas as possíveis seqüências $\{(\tilde{s}_k, \tilde{u}_k)\}$ são representadas pelos possíveis caminhos no diagrama de treliça. A partir das observações $\{y_k\}$, estamos particularmente interessados em estimar a seqüência $\{\hat{s}_k\}$, ou correspondentemente a seqüência $\{\hat{u}_k\}$, que maximiza a métrica total acumulada

$$\sum_{k=0}^{\infty} m\left(\tilde{s}_k, \tilde{u}_k, y_k\right)$$

sobre todas as possíveis seqüências $\{(\tilde{s}_k, \tilde{u}_k)\}$. Idealmente, o receptor deveria

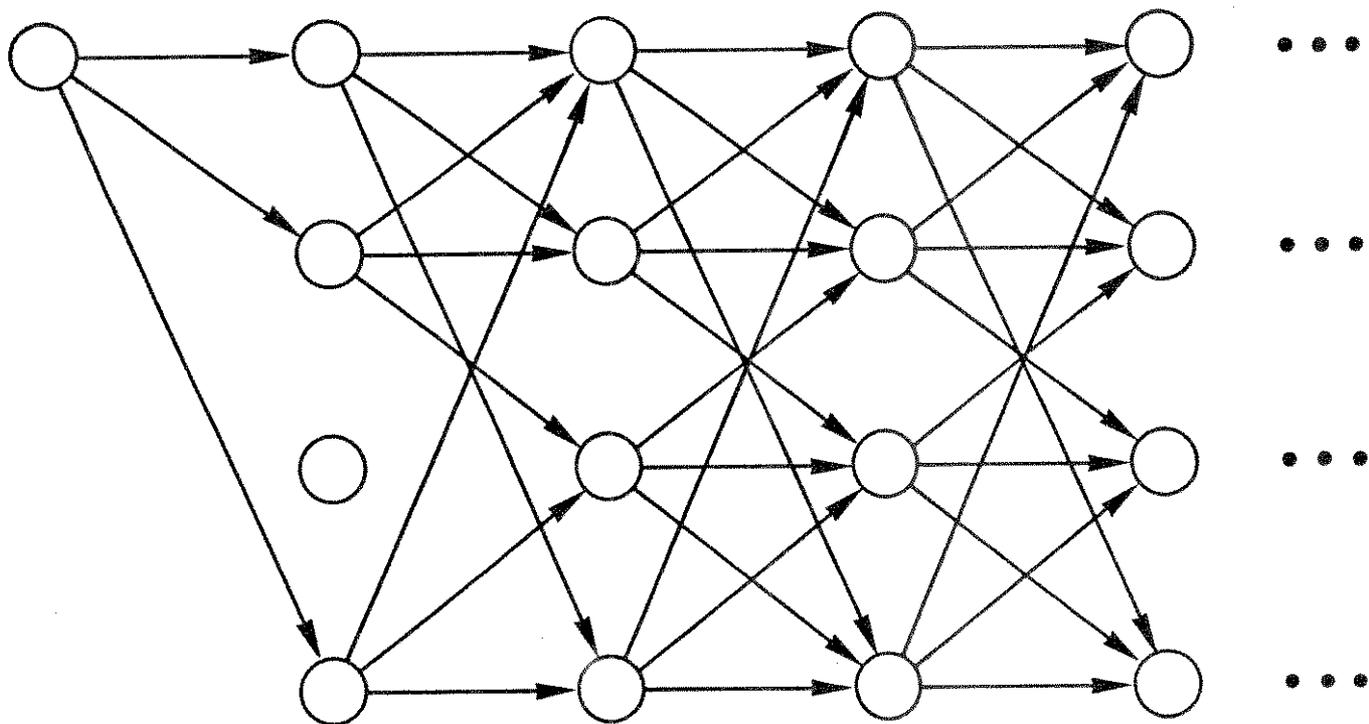


Figura 3.2 - Diagrama de Treliça para $|S| = 4$ e $|U| = 3$.

calcular a métrica total acumulada para cada seqüência possível, e escolher em um instante $t \rightarrow \infty$ a seqüência que conduziisse à maior métrica. Naturalmente este procedimento é infactível uma vez que o número de caminhos cresce exponencialmente com o tempo, $|U|^t$. O algoritmo de Viterbi elimina a maior parte dos possíveis caminhos, reduzindo enormemente a complexidade, sem nenhuma perda de otimalidade. Adicionalmente o processo pode ser truncado em um instante de tempo finito $t = L$, onde a decisão é tomada, com uma perda de otimalidade desprezível quando comparada ao caso ideal, $t \rightarrow \infty$. Os parágrafos que seguem demonstram estas propriedades profícuas do algoritmo de Viterbi.

Vamos considerar um sistema com δ estados,

$$S = \{e_1, e_2, \dots, e_\delta\}. \quad (3.7)$$

Basicamente, o algoritmo de Viterbi calcula a métrica para cada ramo da treliça, a medida que as observações $\{y_k\}$ alcançam o receptor. Desta maneira, a métrica total acumulada em um instante de tempo $t = n + 1$, para uma certa seqüência $\{(\tilde{s}_k, \tilde{u}_k)\}$, é dada por

$$\sum_{k=0}^n m\left((\tilde{s}_k, \tilde{u}_k), y_k\right).$$

Esta seqüência é representada por um caminho na treliça o qual é mostrado na Figura 3.3.

Suponhamos que no instante de tempo $t = n + 1$ dois caminhos correspondentes às seqüências $\{(s_k, u_k)\}$ e $\{(\tilde{s}_k, \tilde{u}_k)\}$, com métricas acumuladas

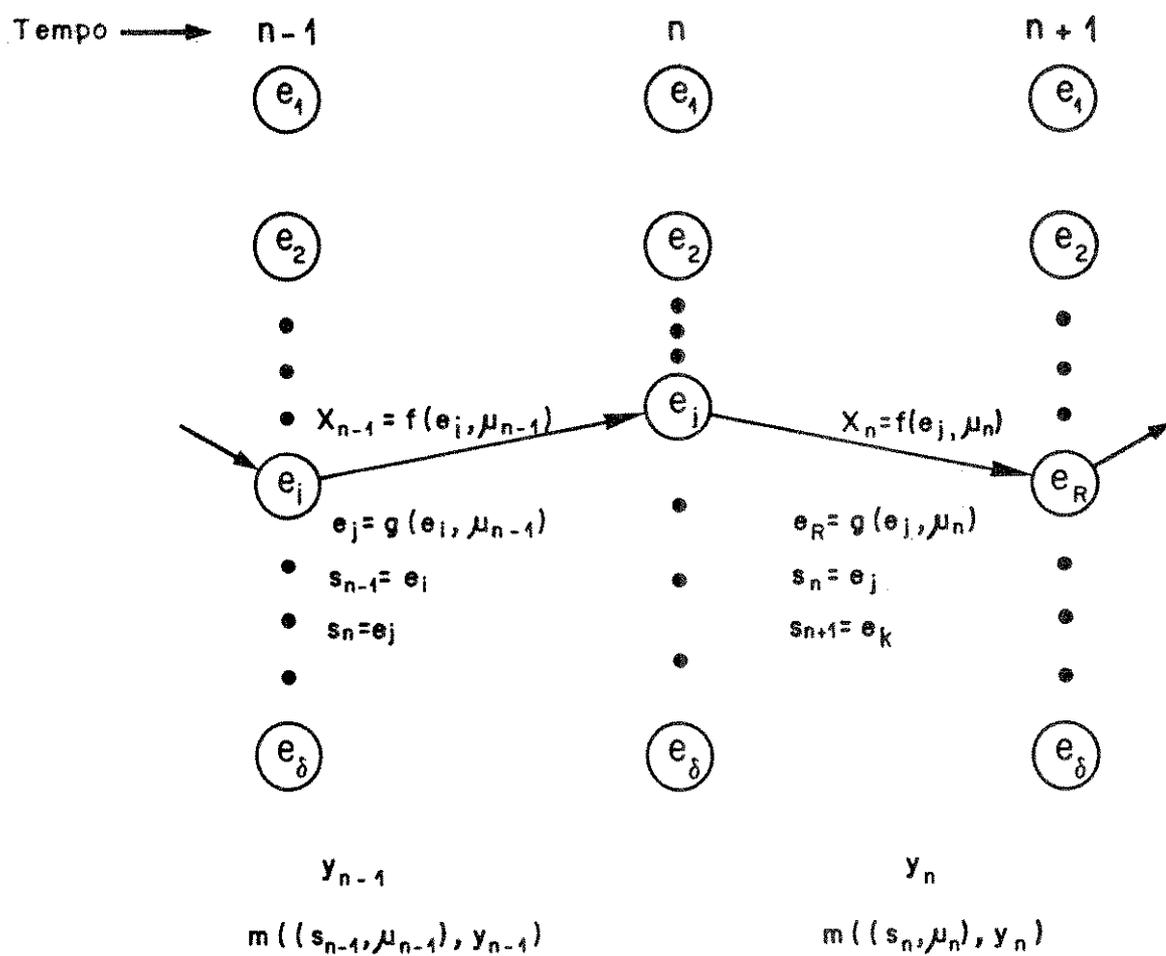


Figura 3.3 - Caminho e Métricas na Treliza.

$$\sum_{k=0}^n m\left((s_k, u_k), y_k\right)$$

e

$$\sum_{k=0}^n m\left((\tilde{s}_k, \tilde{u}_k), y_k\right),$$

respectivamente, converjam para um mesmo estado $s_{n+1} = \tilde{s}_{n+1} = e_k$. Uma vez que o algoritmo selecionará apenas o caminho de maior métrica acumulada, podemos sem qualquer perda de otimalidade eliminar o caminho de menor métrica acumulada no instante $t = n + 1$. Isto é verdade independentemente das observações $\{y_k\}_{k=n+1}^{\infty}$. Caso mais de dois caminhos converjam para o mesmo estado, apenas aquele que possui a maior métrica acumulada, o qual será chamado de "sobrevivente", permanece na disputa. O processo de eliminação de caminhos é ilustrado na Figura 3.4.

Realizando a eliminação de caminhos em todos os estados e para todos os instantes de tempo, o algoritmo de Viterbi apresenta $|S| = \delta$ candidatos ao caminho de máxima métrica acumulada a cada instante de tempo. Este valor deve ser comparado ao número de possíveis caminhos $|U|^t$ que cresce exponencialmente com o tempo.

O último passo em direção à factibilidade do estimador é galgado pelo truncamento da memória de caminho para um número finito L . Isto significa armazenar apenas as L últimas transições entre estados na treliça. Assim, quando uma nova observação chega ao receptor, toda a informação na treliça em L instantes de tempo anteriores é eliminada e uma decisão final para aquele instante é processada. Se o valor L é suficientemente grande, com alta probabilidade a decisão acima mencionada corresponde àquela do caminho de máxima métrica. Isto se deve ao fato de que tipicamente os M sobreviventes (entre eles o caminho de máxima métrica) compartilham as mesmas partes iniciais, o que assegura uma es-

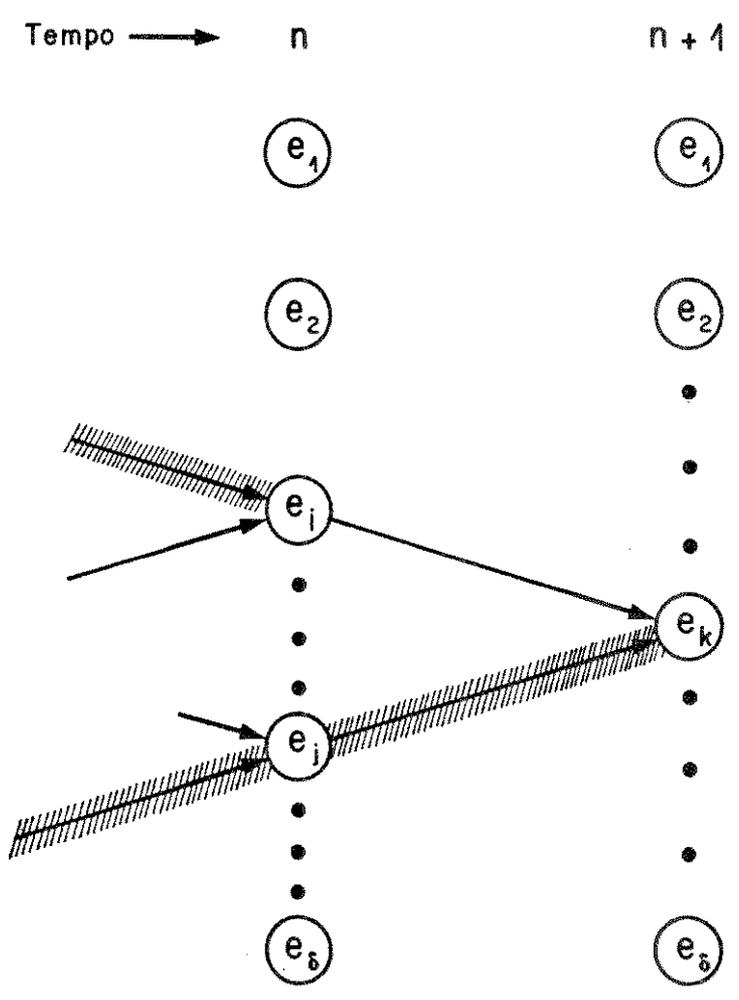


Figura 3.4 - Eliminação de Caminhos no Algoritmo de Viterbi.

timação confiável para L suficientemente grande. Isto é ilustrado na Figura 3.5, onde apenas os sobreviventes são indicados por uma questão de clareza. Os ramos sinuosos constituem o caminho de máxima métrica. Para esta seqüência particular o truncamento para $L = 4$ transições entre estados não induz a perda de otimalidade. Em códigos convolucionais a perda de otimalidade é desprezível quando L é escolhido para ser:

$$L \geq 5 \log_2 \delta, \quad (3.8)$$

obtido experimentalmente [2].

Sumariamente, o algoritmo de Viterbi é um estimador da seqüência $\{(\hat{s}_k, \hat{u}_k)\}$ que maximiza a métrica acumulada

$$\sum_{k=0}^{\infty} m\left(\left(\tilde{s}_k, \tilde{u}_k\right), y_k\right),$$

para o sistema aqui considerado, e que é implementável.

3.4. OS EVENTOS DE ERRO

A partir do que segue, vamos concentrar os esforços no sentido de avaliar o desempenho do sistema em foco, quando o algoritmo de Viterbi é empregado para estimar a seqüência transmitida. Para tanto, vamos nos referir à medida de distorção

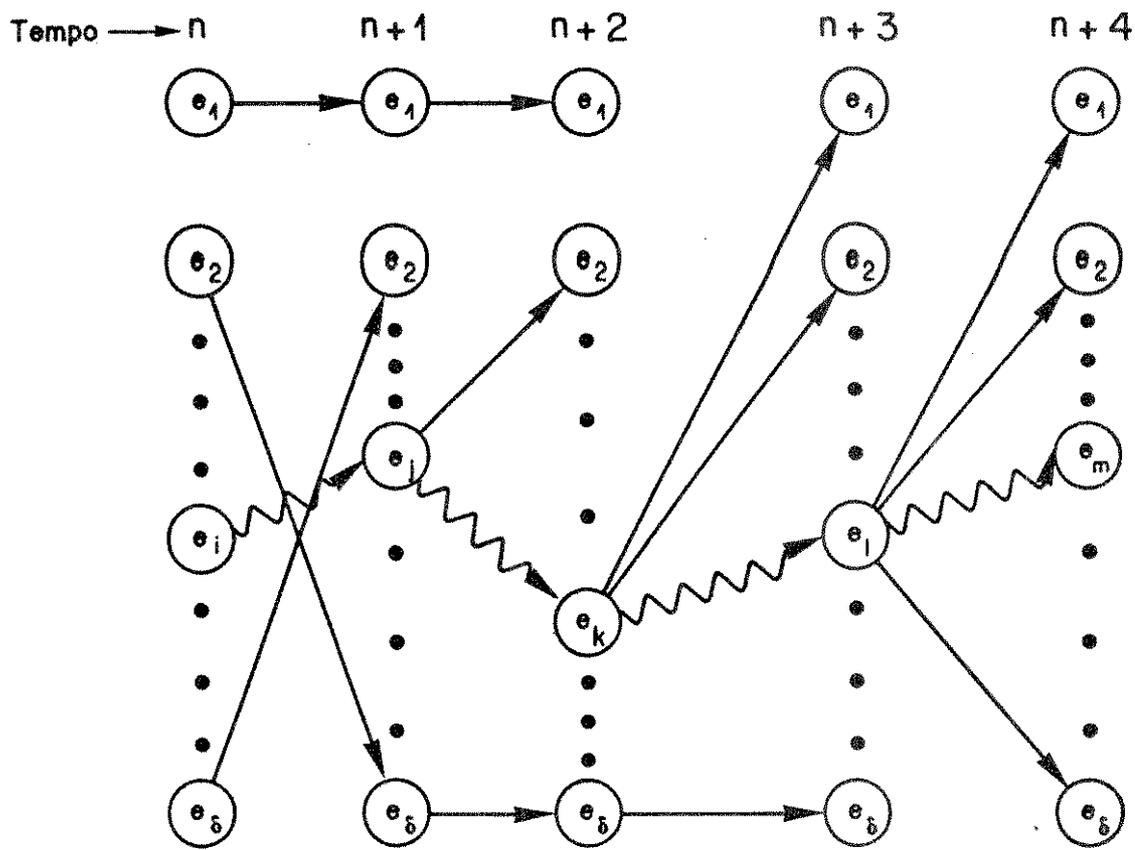


Figura 3.5 - Efeito do Truncamento da Memória de Caminho.

$$d(x_n, \hat{x}_n) = d\left((s_n, u_n), (\hat{s}_n, \hat{u}_n)\right),$$

como uma função não negativa sujeita à restrição dada em (3.6), onde (s_n, u_n) é o estado e o sinal de entrada verdadeiros no instante $t = n$, e (\hat{s}_n, \hat{u}_n) é o estado e o sinal de entrada da seqüência de máxima métrica acumulada, ou seja, selecionados pelo algoritmo de Viterbi neste mesmo instante.

Quando ocorrer de $(s_n, u_n) \neq (\hat{s}_n, \hat{u}_n)$, a medida de distorção assume um valor não negativo em $t = n$. Esta situação acontece quando o caminho de máxima métrica acumulada diverge do caminho correto (na treliça) em algum instante $0 \leq i \leq n$, e estes voltam a se encontrar em um instante $j > n$. Quando isto acontece dizemos que houve um "evento de erro", o qual é ilustrado na Figura 3.6. Matematicamente vamos caracterizar um evento de erro da forma abaixo

$$s_i = \hat{s}_i ; s_j = \hat{s}_j$$

$$s_k \neq \hat{s}_k ; i < k < j$$

$$\sum_{k=i}^{j-1} m\left((\hat{s}_k, \hat{u}_k), y_k\right) \geq \sum_{k=i}^{j-1} m\left((s_k, u_k), y_k\right). \quad (3.9)$$

Devemos notar que existem vários eventos de erro associados à condição $(s_n, u_n) \neq (\hat{s}_n, \hat{u}_n)$, e que devem ser computados na avaliação de desempenho. Isto porque qualquer subsequência $\{\tilde{s}_k\}_{k=i}^j$ na treliça satisfazendo

$$s_i = \tilde{s}_i ; s_j = \tilde{s}_j$$

$$s_k \neq \tilde{s}_k ; i < k < j, \quad (3.10)$$

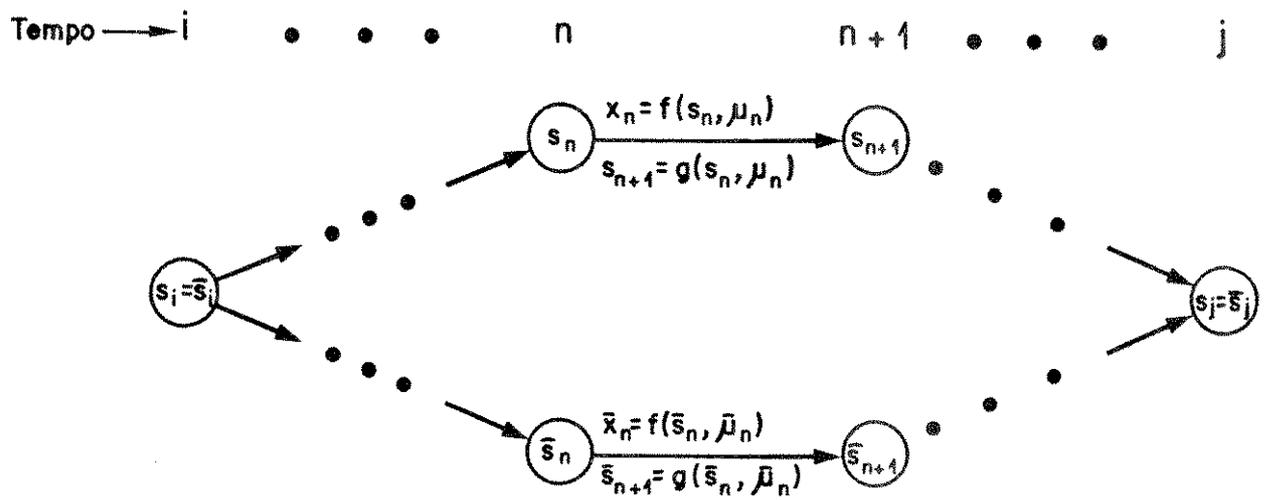


Figura 3.6 - Evento de Erro.

pode pertencer ao caminho de maior métrica. Assim, sendo $\{s_k\}_{k=i}^j$ a subsequência correta e $\{\tilde{s}_k\}_{k=i}^j$ uma possível subsequência, denotamos a probabilidade deste evento de erro por

$$\begin{aligned} P\left(\{s_k\}_{k=i}^j \rightarrow \{\tilde{s}_k\}_{k=i}^j\right) &= \Pr\left\{\sum_{k=i}^{j-1} m\left(\tilde{s}_k, \tilde{u}_k, y_k\right) \geq \sum_{k=i}^{j-1} m\left(s_k, u_k, y_k\right) \mid \{s_k\}\right\} \\ &= \Pr\left\{\sum_{k=i}^{j-1} \left[m\left(\tilde{s}_k, \tilde{u}_k, y_k\right) - m\left(s_k, u_k, y_k\right)\right] \geq 0 \mid \{s_k\}\right\}. \end{aligned} \quad (3.11)$$

Aplicando o limitante de Chernoff com parâmetro $\lambda \geq 0$ (ver Apêndice A) em (3.11) ficamos com

$$P\left(\{s_k\}_{k=i}^j \rightarrow \{\tilde{s}_k\}_{k=i}^j\right) \leq E\left\{\exp\left[\lambda \sum_{k=i}^{j-1} \left\{m\left(\tilde{s}_k, \tilde{u}_k, y_k\right) - m\left(s_k, u_k, y_k\right)\right\}\right] \mid \{s_k\}\right\}. \quad (3.12)$$

Devido a independência estatística das variáveis $\{n_k\}$, e, como consequência, das variáveis $\{y_k\}$ a expressão em (3.12) se torna

$$\begin{aligned} P\left(\{s_k\}_{k=i}^j \rightarrow \{\tilde{s}_k\}_{k=i}^j\right) &\leq E\left\{\exp\left[\lambda \sum_{k=i}^{j-1} \left\{m\left(\tilde{s}_k, \tilde{u}_k, y_k\right) - m\left(s_k, u_k, y_k\right)\right\}\right] \mid \{s_k\}\right\} \\ &= \prod_{k=i}^{j-1} E\left\{\exp\left[\lambda \left\{m\left(\tilde{s}_k, \tilde{u}_k, y_k\right) - m\left(s_k, u_k, y_k\right)\right\}\right] \mid \{s_k\}\right\} \\ &= \prod_{k=i}^{j-1} D\left(\lambda, \left(\tilde{s}_k, \tilde{u}_k\right), \left(s_k, u_k\right)\right), \end{aligned} \quad (3.13)$$

onde na última passagem assumimos que as mesmas variáveis independentes são também identicamente distribuídas.

Na expressão em (3.13) identificamos a função

$$D\left(\lambda, (\tilde{s}_k, \tilde{u}_k), (s_k, u_k)\right) = E\left\{\exp\left[\lambda\left(m\left((\tilde{s}_k, \tilde{u}_k), y_k\right) - m\left((s_k, u_k), y_k\right)\right)\right] \mid \{s_k\}\right\} \quad (3.14)$$

que assume valores conhecidos para as métricas comumente utilizadas. No caso particular da métrica de máxima verossimilhança dada em (3.5a) teremos [2]

$$D\left(\lambda, (\tilde{s}_k, \tilde{u}_k), (s_k, u_k)\right) = \sum_{y_k} p(y_k | x_k) \cdot \left\{ \frac{p(y_k | \tilde{x}_k)}{p(y_k | x_k)} \right\}^\lambda,$$

que apresenta um mínimo para $\lambda = 1/2$:

$$D\left(1/2, (\tilde{s}_k, \tilde{u}_k), (s_k, u_k)\right) = \sum_{y_k} \sqrt{p(y_k | x_k) p(y_k | \tilde{x}_k)} \quad (3.15)$$

Esta forma final é conhecida como a Função de Bhattacharyya.

3.5. A DISTORÇÃO MÉDIA

Na Seção 3.4, derivamos uma expressão para a probabilidade de um

evento de erro, gerado quando uma possível subsequência $(\tilde{s}_k)_{k=i}^j$ apresenta uma métrica acumulada maior que a da subsequência $(s_k)_{k=i}^j$, tida como verdadeira. Nesta mesma seção foi dito que existem vários eventos de erro do tipo $(\tilde{s}_k)_{k=i}^j$ quando uma seqüência $(s_k)_{k=i}^j$ é assumida como verdadeira. Vamos colecionar estes eventos de erro em um conjunto que será denotado por

$$S(i, j | (s_k)_{k=i}^j) \triangleq \left\{ (\tilde{s}_k)_{k=i}^j : s_i = \tilde{s}_i, s_j = \tilde{s}_j, s_k \neq \tilde{s}_k, i < k < j \right\} \quad (3.16)$$

A partir de (3.16) notamos que se $i \leq n < j$, a distorção em um instante de tempo $t = n$ para um evento de erro $(\tilde{s}_k)_{k=i}^j \in S(i, j | (s_k)_{k=i}^j)$ é igual a

$$d\left((s_n, u_n), (\tilde{s}_n, \tilde{u}_n)\right).$$

Por outro lado, o desempenho do sistema em um instante $t = n$, quando a seqüência transmitida é conhecida, é fornecido pela distorção média entre a seqüência de máxima métrica acumulada $\left\{(\hat{s}_k, \hat{u}_k)\right\}$ e a seqüência verdadeira conhecida $\left\{(s_k, u_k)\right\}$, neste mesmo instante. Esta quantidade pode ser expressa matematicamente por

$$E\left\{d\left((s_n, u_n), (\hat{s}_n, \hat{u}_n)\right) \mid (s_k)\right\} = \sum_{0 \leq i \leq n} \sum_{j > n} \sum_{S(i, j | (s_k)_{k=i}^j)} d\left((s_n, u_n), (\tilde{s}_n, \tilde{u}_n)\right) \cdot \Pr\left\{(\hat{s}_k)_{k=i}^j = (\tilde{s}_k)_{k=i}^j \mid (s_k)\right\}. \quad (3.17)$$

Para que possamos prosseguir com a análise acima, é necessário intro-

duzir a condição de estacionaridade a qual é estabelecida quando supomos $n \gg 0$. Sem esta suposição a referida análise se mostra bastante complicada e até mesmo de pouco interesse. Nestes termos, podemos imaginar uma treliça que se estende de $t = -\infty$ a $t = \infty$ e permitir que os eventos de erro possam começar em qualquer instante de tempo no intervalo $(-\infty, n]$. Esta situação leva a um limitante superior para a expressão dada em (3.17):

$$E\left\{d\left((s_n, u_n), (\hat{s}_n, \hat{u}_n)\right) \mid \{s_k\}\right\} \leq \sum_{i \leq n} \sum_{j > n} \sum_{S(i, j | \{s_k\}_{k=i}^j)} d\left((s_n, u_n), (\tilde{s}_n, \tilde{u}_n)\right) \cdot \Pr\left(\{\hat{s}_k\}_{k=i}^j = \{\tilde{s}_k\}_{k=i}^j \mid \{s_k\}\right). \quad (3.18)$$

Em (3.18) identificamos $\Pr\left(\{\hat{s}_k\}_{k=i}^j = \{\tilde{s}_k\}_{k=i}^j \mid \{s_k\}\right)$ como a probabilidade da subsequência $\{\tilde{s}_k\}_{k=i}^j$ apresentar a maior métrica acumulada dentre todas as subsequências em $S(i, j | \{s_k\}_{k=i}^j)$. Esta quantidade é obviamente menor que $\Pr\left(\{s_k\}_{k=i}^j \rightarrow \{\tilde{s}_k\}_{k=i}^j\right)$, que é a probabilidade da subsequência $\{\tilde{s}_k\}_{k=i}^j$ apresentar uma métrica acumulada maior que a da subsequência verdadeira $\{s_k\}_{k=i}^j$ somente. Como esta última probabilidade é conhecida de (3.13), vamos substituí-la pela primeira probabilidade em (3.18) às expensas da introdução de mais uma pequena imprecisão no limitante. Deste modo, (3.18) é dado por

$$E\left\{d\left((s_n, u_n), (\hat{s}_n, \hat{u}_n)\right) \mid \{s_k\}\right\} \leq \sum_{i \leq n} \sum_{j > n} \sum_{S(i, j | \{s_k\}_{k=i}^j)} d\left((s_n, u_n), (\tilde{s}_n, \tilde{u}_n)\right) \cdot \Pr\left(\{s_k\}_{k=i}^j \rightarrow \{\tilde{s}_k\}_{k=i}^j\right). \quad (3.19)$$

Em geral, nos sistemas práticos comumente encontrados, não se conhece

a seqüência transmitida. Adicionalmente, o limitante em (3.19) pode variar segundo a seqüência que está sendo transmitida*. Com isto, a forma mais geral para o limitante superior de desempenho do sistema em consideração é fornecida ao se tomar a média de (3.19) sobre todas as possíveis seqüências transmitidas. Para tanto, vamos denotar o conjunto de todos os pares de subseqüências possíveis por

$$S(i, j) \triangleq \left\{ \left(\{s_k\}_{k=i}^j, \{\tilde{s}_k\}_{k=i}^j \right) : s_i = \tilde{s}_i, s_j = \tilde{s}_j, s_k \neq \tilde{s}_k; i < k < j \right\}, \quad (3.20)$$

onde $\{s_k\}_{k=i}^j$ é a subseqüência verdadeira e $\{\tilde{s}_k\}_{k=i}^j$ é um evento de erro.

Sendo assim, a distorção média no instante $t = n$ é fornecida por

$$E \left\{ d \left((s_n, u_n), (\hat{s}_n, \hat{u}_n) \right) \right\} = \sum_{\{s_k\}} P \left(\{s_k\} \right) E \left\{ d \left((s_n, u_n), (\hat{s}_n, \hat{u}_n) \right) \mid \{s_k\} \right\}, \quad (3.21)$$

onde $P(\{s_k\})$ é a probabilidade da seqüência $\{s_k\}$ ter sido transmitida. Percebe-se que as subseqüências $\{s_k\}_{k=i}^j$ da seqüência transmitida representam uma cadeia de Markov de primeira ordem. Deste modo a probabilidade destes eventos é descrita por

$$P \left(\{s_k\}_{k=i}^j \right) = p(s_i) \prod_{k=i}^{j-1} q(u_k), \quad (3.22)$$

onde $p(s_i)$ é a probabilidade do estado s_i em regime estacionário.

* Esta situação não ocorre no caso dos códigos convolucionais, como será visto na Seção 3.7.

Substituindo (3.22) e (3.19) em (3.21) esta se torna

$$\begin{aligned}
 E\left\{d\left((s_n, u_n), (\hat{s}_n, \hat{u}_n)\right)\right\} &\leq \sum_{(s_k)} P\left((s_k)\right) \sum_{i \leq n} \sum_{j > n} \sum_{S(i, j) | \{s_k\}_{k=i}^j} \\
 &\quad d\left((s_n, u_n), (\tilde{s}_n, \tilde{u}_n)\right) P\left(\{s_k\}_{k=i}^j \rightarrow \{\tilde{s}_k\}_{k=i}^j\right) \\
 &= \sum_{i \leq n} \sum_{j > n} \sum_{S(i, j)} d\left((s_n, u_n), (\tilde{s}_n, \tilde{u}_n)\right) \cdot \\
 &\quad \cdot P(s_i) \prod_{k=i}^{j-1} q(u_k) D\left(\lambda, (s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right) \quad (3.23)
 \end{aligned}$$

onde (3.13) foi substituído em (3.19).

A forma final da distorção média é obtida explorando-se algumas propriedades oriundas da estacionaridade imposta no princípio da análise, quais sejam:

$$P\left(\{s_k\}_{k=i}^j\right) = P\left(\{s_k\}_{k=i+L}^{j+L}\right) \quad (3.24a)$$

e

$$P\left(\{s_k\}_{k=i}^j \rightarrow \{\tilde{s}_k\}_{k=i}^j\right) = P\left(\{s_k\}_{k=i+L}^{j+L} \rightarrow \{\tilde{s}_k\}_{k=i+L}^{j+L}\right), \quad (3.24b)$$

para todo L inteiro. Das propriedades em (3.24) podemos concluir que o "ensemble" das subsequências $S(i, j)$ coincide com o "ensemble" das subsequências $S(i+L, j+L)$. Por outro lado em (3.23), fizemos os instantes de início e

término ("i" e "j", respectivamente) dos eventos de erro variarem na região $i \leq n < j$, enquanto o instante $t = n$ para a distorção era mantido constante. Segundo as consequências da estacionaridade, isto é equivalente a fixar o instante de início de evento de erro na origem ($i = 0$), fazer o instante de término de evento de erro variar na região $j > 0$, e computar todas as distorções nos instantes que vai de $\ell = 0$ até $\ell = j - 1$, para cada par de subsequências em $S(0, j)$. Nestes termos, a expressão (3.23) independe da variável n e é equivalente a

$$\bar{d} \leq \sum_{j=1}^{\infty} \sum_{S(0,j)} \left[\sum_{\ell=0}^{j-1} d\left((s_{\ell}, u_{\ell}), (\tilde{s}_{\ell}, \tilde{u}_{\ell})\right) \right] p(s_0) \prod_{k=0}^{j-1} q(u_k) D\left(\lambda, (s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right), \quad (3.25)$$

onde \bar{d} substitui $E\left\{d\left((s_n, u_n), (\hat{s}_n, \hat{u}_n)\right)\right\}$ para ressaltar a independência com relação ao instante n .

Como foi mencionado anteriormente, devemos expressar o limitante de desempenho dado em (3.25) pela Função de Transferência Generalizada. Isto é conseguido aplicando-se a igualdade

$$X = \frac{d}{dI} I^X \Big|_{I=1} \quad (3.26)$$

na somatória em ℓ de (3.25), o que conduz a

$$\begin{aligned}
 \left[\sum_{\ell=0}^{j-1} d((s_{\ell}, u_{\ell}), (\tilde{s}_{\ell}, \tilde{u}_{\ell})) \right] &= \frac{d}{dI} \prod_{I=1}^{j-1} \left[\sum_{\ell=0}^{j-1} d((s_{\ell}, u_{\ell}), (\tilde{s}_{\ell}, \tilde{u}_{\ell})) \right] \Bigg|_{I=1} \\
 &= \frac{d}{dI} \left\{ \prod_{\ell=0}^{j-1} \prod_{I=1} d((s_{\ell}, u_{\ell}), (\tilde{s}_{\ell}, \tilde{u}_{\ell})) \right\} \Bigg|_{I=1}
 \end{aligned} \tag{3.27}$$

Agora substituindo (3.27) em (3.25) obtemos finalmente a forma almejada para o limitante de desempenho:

$$\bar{d} \leq \frac{d}{dI} T(I) \Bigg|_{I=1}, \tag{3.28}$$

onde

$$T(I) = \sum_{j=1}^{\infty} \sum_{S(0,j)} P(s_0) \prod_{k=0}^{j-1} \prod_{I=1} d((s_{\ell}, u_{\ell}), (\tilde{s}_{\ell}, \tilde{u}_{\ell})) \cdot q(u_k) D\left(\lambda, (s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right) \tag{3.29}$$

é a Função de Transferência Generalizada.

3.6. OBTENÇÃO DA FUNÇÃO DE TRANSFERÊNCIA GENERALIZADA

No que segue, apresentamos uma maneira sistemática de se obter a Função de Transferência Generalizada dos sistemas genéricos que vimos adotando.

Tal facilidade é imprescindível para a avaliação de desempenho destes sistemas, uma vez que esta segue trivialmente da Função de Transferência, como indicado em (3.28).

Vamos então supor que o sistema apresenta δ estados:

$$S = \{e_1, e_2, \dots, e_\delta\}, \quad (3.30)$$

onde reescrevemos o alfabeto de estados dado em (3.7) por questão de conveniência. Admitamos também que o alfabeto dos sinais de entrada tenha tamanho $|\mathcal{U}| = \varphi$:

$$\mathcal{U} = \{a_1, a_2, \dots, a_\varphi\}. \quad (3.31)$$

Para uma conveniente caracterização dos eventos de erro devemos definir os "super-estados" como os elementos do conjunto $S^2 = S \times S$, onde "x" denota produto cartesiano entre dois conjuntos:

$$S^2 = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{\delta^2}\}. \quad (3.32)$$

De maneira análoga definimos as "super-entradas" do alfabeto $\mathcal{U}^2 = \mathcal{U} \times \mathcal{U}$ como

$$\mathcal{U}^2 = \{\alpha_1, \alpha_2, \dots, \alpha_{\varphi^2}\}. \quad (3.33)$$

Sendo assim, os "super-estados" e as "super-entradas" em um instante de tempo $t = k$ são denotados por

$$S_k = (s_k, \tilde{s}_k) \in S^2 \quad (3.34)$$

$$U_k = (u_k, \tilde{u}_k) \in U^2, \quad (3.35)$$

respectivamente.

Para nossa conveniência, vamos colecionar em um único conjunto os "super-estados" que caracterizam os inícios e términos dos eventos de erro, ou seja, aqueles δ "super-estados" constituídos de pares idênticos de estados, a saber

$$S_{(I)}^2 = \{ \varepsilon_1, \varepsilon_2, \dots, \varepsilon_\delta \}. \quad (3.36)$$

A escolha dos δ primeiros elementos de S^2 para pertencerem a $S_{(I)}^2$ foi feita sem perda de generalidade por uma questão de conveniência.

Obviamente, o complemento do conjunto $S_{(I)}^2$ contém os "super-estados" formados por pares distintos de estados, que representam a parte intermediária dos eventos de erro, a saber

$$S_{(D)}^2 = \{ \varepsilon_{\delta+1}, \varepsilon_{\delta+2}, \dots, \varepsilon_{\delta^2} \}. \quad (3.37)$$

Assim sendo, o diagrama de "super-estados" é mostrado na Figura 3.7.

A partir da nova representação temos a seguinte equivalência:

$$\begin{aligned} S(0, j) &\stackrel{\Delta}{=} \left\{ \left(\{s_k\}_{k=0}^j, \{\tilde{s}_k\}_{k=0}^j \right) : s_0 = \tilde{s}_0; s_j = \tilde{s}_j, s_k \neq \tilde{s}_k; 0 < k < j \right\} \\ &= \left\{ \{S_k\}_{k=0}^j : S_0, S_j \in S_{(I)}^2, S_k \in S_{(D)}^2; 0 < k < j \right\}, \end{aligned} \quad (3.38)$$

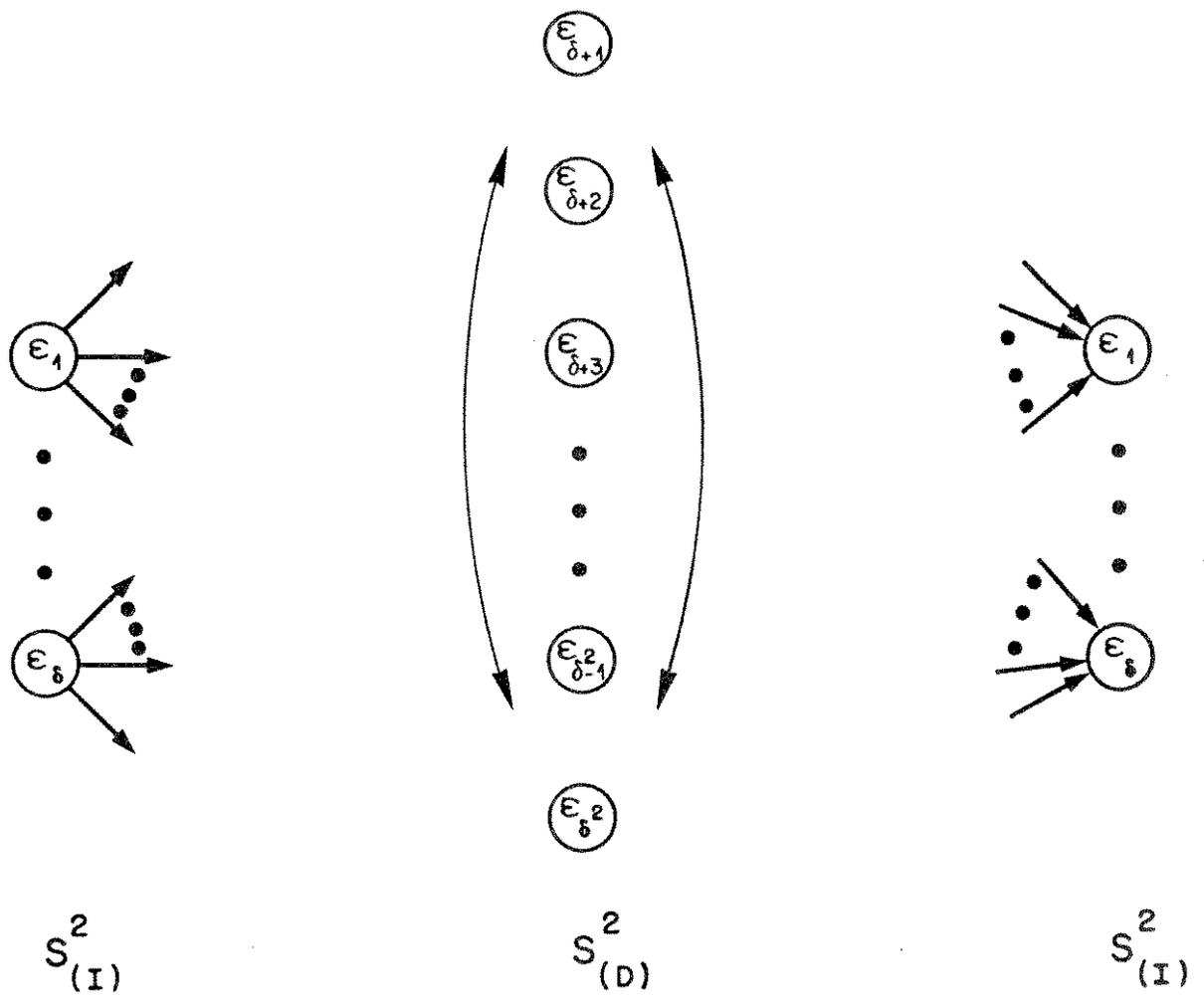


Figura 3.7 - Diagrama de "Super-Estados".

onde

$$\{S_k\}_{k=0}^j = (S_0, S_1, \dots, S_j). \quad (3.39)$$

Algumas representações mais compactas, em função dos "super-estados" e "super-entradas", são também adotadas:

$$p(S_0) \triangleq p(s_0), \quad (3.40a)$$

$$d(S_k, U_k) \triangleq d\left((s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right), \quad (3.40b)$$

$$q(U_k) \triangleq q(u_k), \quad (3.40c)$$

$$D(\lambda, S_k, U_k) \triangleq D\left(\lambda, (s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right), \quad (3.40d)$$

e as equações de estado

$$s_{k+1} = g(s_k, u_k)$$

$$\tilde{s}_{k+1} = g(\tilde{s}_k, \tilde{u}_k)$$

são representadas por

$$S_{k+1} = G(S_k, U_k). \quad (3.40e)$$

Nestes termos, a Função de Transferência Generalizada da equação (3.29) pode ser expressa na forma equivalente:

$$T(I) = \sum_{j=1}^{\infty} \sum_{S(0,j)} p(S_0) \prod_{k=0}^{j-1} I^{d(S_k, U_k)} \cdot q(U_k) \cdot D(\lambda, S_k, U_k). \quad (3.41)$$

Em termos de diagrama de "super-estados", a função acima computa todos os caminhos que começam e terminam em um "super-estado" pertencendo a $S_{(I)}^2$ e com todos os "super-estados" intermediários pertencendo a $S_{(D)}^2$.

Devemos notar ainda nesta função que não existe "super-estados" intermediários quando $j = 1$. Na verdade se considerarmos a restrição sobre a função $g(\cdot, \cdot)$ da seção 3.2 deveremos ter que $S(0, 1) = \phi$, ou seja, não existe nenhum par de caminhos que divirja de um estado e a ele retorne sem atingir um estado intermediário. Portanto, a equação (3.41) pode ser equivalentemente reescrita como

$$T(I) = \sum_{j=2}^{\infty} \sum_{S(0,j)} p(S_0) \prod_{k=0}^{j-1} I^{d(S_k, U_k)} \cdot q(U_k) \cdot D(\lambda, S_k, U_k). \quad (3.42)$$

Com o propósito de representar $T(I)$ em uma forma matricial, vamos definir o ganho do ramo que vai do "super-estado" ϵ_i ao "super-estado" ϵ_j , denotado por a_{ij} , como sendo:

$$a_{ij} = \begin{cases} I^{d(\epsilon_i, U)} \cdot q(U) D(\lambda, \epsilon_i, U); & \text{se existe } U \in \mathcal{U}^2 \text{ tal que } \epsilon_j = G(\epsilon_i, U) \\ 0 & ; \text{ caso contrário.} \end{cases} \quad (3.43)$$

Segundamente, vamos definir a matriz $(\delta^2 - \delta) \times (\delta^2 - \delta)$:

$$\underline{A} = \begin{bmatrix} a_{\delta+1, \delta+1} & a_{\delta+2, \delta+1} & \cdot & \cdot & \cdot & a_{\delta^2, \delta+1} \\ a_{\delta+1, \delta+2} & a_{\delta+2, \delta+2} & \cdot & \cdot & \cdot & a_{\delta^2, \delta+2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{\delta+1, \delta^2} & a_{\delta+2, \delta^2} & \cdot & \cdot & \cdot & a_{\delta^2, \delta^2} \end{bmatrix} \quad (3.44)$$

e os vetores $(\delta^2 - \delta) \times 1$:

$$\underline{b}_{-i} = \begin{bmatrix} a_{i, \delta+1} \\ a_{i, \delta+2} \\ \cdot \\ \cdot \\ a_{i, \delta^2} \end{bmatrix}, \quad \underline{c}_{-i} = \begin{bmatrix} a_{\delta+1, i} \\ a_{\delta+2, i} \\ \cdot \\ \cdot \\ a_{\delta^2, i} \end{bmatrix}; \quad i=1, 2, \dots, \delta, \quad (3.45)$$

onde o sublinhado denota matriz ou vetor. Definamos também $t_i(I)$, $i = \delta+1, \delta+2, \dots, \delta^2$, como sendo a função de transferência a partir de todos os "super-estados" iniciais para o "super-estado" intermediário $\epsilon_i \in S_{(D)}^2$, a saber

$$t_i(I) = \sum_{k=\delta+1}^{\delta^2} a_{k,i} \cdot t_k(I) + \sum_{j=1}^{\delta} p(\epsilon_j) a_{j,i} \quad ; \quad i=\delta+1, \delta+2, \dots, \delta^2 \quad (3.46)$$

Definindo o vetor função de transferência de estados intermediários

$$\underline{t}(I) = \begin{bmatrix} t_{\delta+1}(I) \\ t_{\delta+2}(I) \\ \cdot \\ \cdot \\ t_{\delta^2}(I) \end{bmatrix} \quad (3.47)$$

e combinando (3.46) e (3.47) chega-se a

$$\underline{t}_i(I) = \underline{A} \underline{t}(I) + \sum_{j=1}^{\delta} p(\epsilon_j) \underline{b}_j \quad (3.48)$$

ou

$$\underline{t}_i(I) = (\underline{I} - \underline{A})^{-1} \left(\sum_{j=1}^{\delta} p(\epsilon_j) \underline{b}_j \right), \quad (3.49)$$

onde \underline{I} é a matriz identidade de ordem $(\delta^2 - \delta) \times (\delta^2 - \delta)$ e não deve ser confundida com a variável I .

A Função de Transferência na forma matricial é então fornecida por

$$T(I) = \sum_{j=1}^{\delta} \underline{c}_j^T \underline{t}(I) = \left(\sum_{j=1}^{\delta} \underline{c}_j \right) (\underline{I} - \underline{A})^{-1} \left(\sum_{i=1}^{\delta} p(\epsilon_i) \underline{b}_i \right), \quad (3.50)$$

onde T denota transposição matricial.

Já é notório que a distorção média é obtida a partir da derivada $\frac{dT(I)}{dI}$. Vamos então denotar as derivadas dos termos envolvidos em (3.50) por:

$$\underline{A}' = \frac{d}{dI} \underline{A}, \quad (3.51a)$$

$$\underline{b}'_i = \frac{d}{dI} \underline{b}_i, \quad (3.51b)$$

$$\underline{c}'_j = \frac{d}{dI} \underline{c}_j. \quad (3.51c)$$

Considerando a identidade

$$\underline{I} = (\underline{I} - \underline{A})^{-1}(\underline{I} - \underline{A}) \quad (3.52)$$

teremos que

$$\begin{aligned} \underline{0} &= \frac{d}{dI} \left\{ (\underline{I} - \underline{A})^{-1}(\underline{I} - \underline{A}) \right\} \\ &= \left(\frac{d}{dI}(\underline{I} - \underline{A})^{-1} \right) (\underline{I} - \underline{A}) + (\underline{I} - \underline{A})^{-1} \left(- \frac{d}{dI} \underline{A} \right) \end{aligned}$$

ou

$$\frac{d}{dI}(\underline{I} - \underline{A})^{-1} = (\underline{I} - \underline{A})^{-1} \underline{A}'(\underline{I} - \underline{A})^{-1} \quad (3.53)$$

Finalmente, usando (3.51) e (3.53) chegamos a

$$\begin{aligned} \frac{dT(I)}{dI} &= \left(\sum_{j=1}^{\delta} \underline{c}'_j \right)^T (\underline{I} - \underline{A})^{-1} \left(\sum_{i=1}^{\delta} p(\epsilon_i) \underline{b}_i \right) + \\ &+ \left(\sum_{j=1}^{\delta} \underline{c}_j \right)^T (\underline{I} - \underline{A})^{-1} \left(\sum_{i=1}^{\delta} p(\epsilon_i) \underline{b}'_i \right) + \\ &+ \left(\sum_{j=1}^{\delta} \underline{c}_j \right)^T (\underline{I} - \underline{A})^{-1} \underline{A}'(\underline{I} - \underline{A})^{-1} \left(\sum_{i=1}^{\delta} p(\epsilon_i) \underline{b}_i \right) \end{aligned} \quad (3.54)$$

que conduz ao desempenho do sistema através de (3.28).

3.7. CASO ESPECIAL: OS CÓDIGOS CONVOLUCIONAIS

A análise de desempenho até então desenvolvida tem sido bastante geral. Segue deste fato que as funções $g(\cdot, \cdot)$, $f(\cdot, \cdot)$ e $h(\cdot, \cdot)$ da seção 3.2 são genéricas, como podem ser a métrica e a medida de distorção. A partir de agora vamos particularizar esta situação para um caso onde a distorção média dada em (3.19) independe da seqüência transmitida $\{s_k\}$. Com isto é suficiente escolher uma seqüência conveniente, digamos $\{s_k^0\}$, e avaliar o desempenho que será fornecido por aquela expressão. O fato da média efetuada em (3.21) não ser necessária é suficiente para a redução do número de "super-estados" que deve ser considerado na avaliação da Função de Transferência. Por uma questão de conveniência, vamos assumir que

$$s_k^0 = e_1 \quad \text{para todo } k \quad (3.55)$$

é a seqüência de estados transmitida, para uma seqüência de entrada

$$u_k = u_1 \quad \text{para todo } k. \quad (3.56)$$

A distorção média é então fornecida por:

$$\begin{aligned} \bar{d} &= E\left\{d\left((s_n, u_n), (\hat{s}_n, \hat{u}_n)\right) \mid \{s_k^0\}\right\} \\ &\approx \sum_{j=2}^{\infty} S(0, j \mid \{s_k^0\}_{k=0}^j) \left[\sum_{\ell=0}^{j-1} d\left((e_1, u_1), (\tilde{s}_\ell, \tilde{u}_\ell)\right) \right] \prod_{k=0}^{j-1} D\left((\lambda, (e_1, u_1), (\tilde{s}_k, \tilde{u}_k))\right) \\ &= \frac{d}{d\lambda} T_0(\lambda) \Big|_{\lambda=1}, \end{aligned} \quad (3.57)$$

onde

$$T_0(I) = \sum_{j=2}^{\infty} S(0, j) \sum_{\{s_k^0\}_{k=0}^j} \prod_{k=0}^{j-1} I^{d((e_1, u_1), (\tilde{s}_k, \tilde{u}_k))} \cdot D\left(\lambda, (e_1, u_1), (\tilde{s}_k, \tilde{u}_k)\right). \quad (3.58)$$

Percebe-se facilmente que o número de "super-estados" foi reduzido para δ , onde os conjuntos $S_{(I)}^2$ e $S_{(D)}^2$ são agora

$$S_{(I)}^2 = \{(e_1, e_1)\}$$

e

$$S_{(D)}^2 = \{(e_1, e_2), (e_1, e_3), \dots, (e_1, e_\delta)\}, \quad (3.59)$$

respectivamente. Devemos então redefinir (3.43) para ser:

$$a_{ij} = \begin{cases} I^{d((e_1, u_1), (e_i, u))} \cdot D\left(\lambda, (e_1, u_1), (e_i, u)\right); & \text{se existe } u \in \mathcal{U} \text{ tal que} \\ & e_j = g(e_i, u) \\ 0 & ; \text{ caso contrário.} \end{cases} \quad (3.60)$$

As formas reduzidas da matriz \underline{A} e dos vetores \underline{b} e \underline{c} são dadas por

$$\underline{A} = \begin{bmatrix} a_{2,2} & a_{3,2} & \cdot & \cdot & \cdot & a_{\delta,2} \\ a_{2,3} & a_{3,3} & \cdot & \cdot & \cdot & a_{\delta,3} \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ a_{2,\delta} & a_{3,\delta} & \cdot & \cdot & \cdot & a_{\delta,\delta} \end{bmatrix} \quad (3.61)$$

de ordem $(\delta-1) \times (\delta-1)$, e

$$\underline{b} = \begin{bmatrix} a_{1,2} \\ a_{1,3} \\ \cdot \\ \cdot \\ \cdot \\ a_{1,\delta} \end{bmatrix} \quad (3.62)$$

e

$$\underline{c} = \begin{bmatrix} a_{2,1} \\ a_{3,1} \\ \cdot \\ \cdot \\ \cdot \\ a_{\delta,1} \end{bmatrix}, \quad (3.63)$$

de ordens $(\delta-1) \times 1$, respectivamente. Com isto, a Função de Transferência $T_0(I)$ e sua derivada são uma versão simplificada das equações (3.50) e (3.54), respectivamente:

$$T_0(I) = \underline{c}^T (I - \underline{A})^{-1} \underline{b} \quad (3.64)$$

e

$$\frac{d T_0(I)}{dI} = (\underline{c}')^T (\underline{I} - \underline{A})^{-1} \underline{b} + \underline{c}^T (\underline{I} - \underline{A})^{-1} \underline{b}' + \underline{c}^T (\underline{I} - \underline{A})^{-1} \underline{A}' (\underline{I} - \underline{A})^{-1} \underline{b}. \quad (3.65)$$

Um exemplo típico de sistemas de estado finito cujo desempenho independe da sequência transmitida é o dos códigos convolucionais sobre um canal simétrico. No que segue, devemos abordar dois canais simétricos bem conhecidos: o canal binário simétrico (BSC, do inglês: Binary Symmetric Channel) e o canal de entrada binária com ruído branco Gaussiano aditivo (AWGN, do inglês: Additive White Gaussian Noise).

Tomemos como exemplo o codificador convolucional de taxa $R = \frac{2}{3}$ e comprimento total de memória $K = 2$ que é mostrado na Figura 3.8 [2]. O diagrama de estados correspondente é estabelecido na Figura 3.9. Nesta situação, o alfabeto de entrada do sistema é dado por

$$\mathcal{U} = \{(00), (01), (10), (11)\}. \quad (3.66)$$

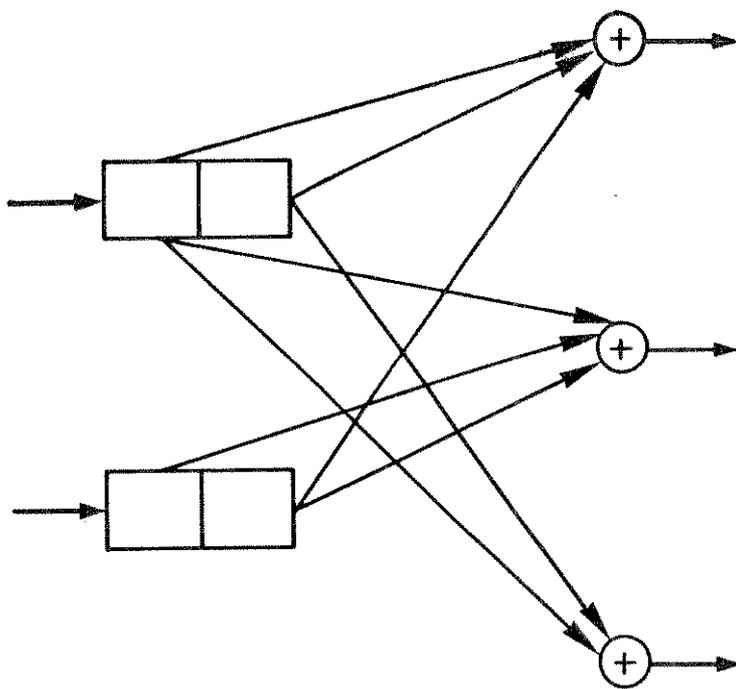
A relação de estado é da forma

$$s_{k+1} = g(s_k, u_k) = u_{k-1}, \quad (3.67)$$

e o alfabeto de estado é dado por

$$S = \mathcal{U} = \{e_1, e_2, e_3, e_4\}, \quad (3.68)$$

segundo o mapeamento $00 \rightarrow e_1$, $01 \rightarrow e_2$, $10 \rightarrow e_3$ e $11 \rightarrow e_4$. O alfabeto de saída do sistema deve ser



$$G_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Figura 3.8 - Codificador Convolutacional $R = \frac{2}{3}$, $K = 2$ [2].

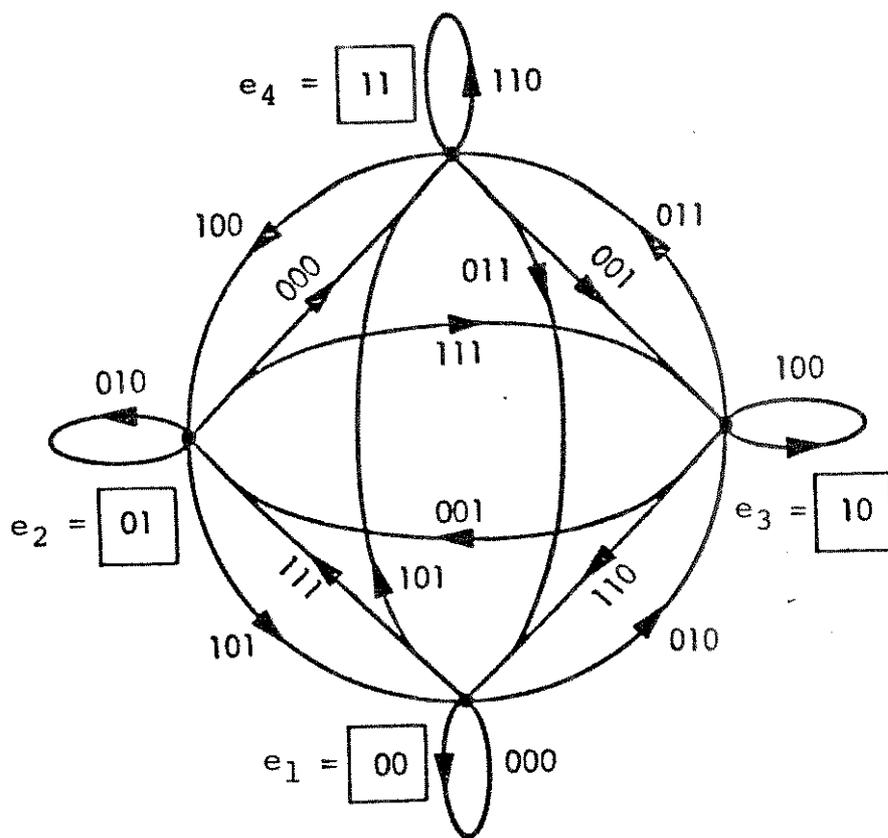


Figura 3.9 - Diagrama de Estado do Codificador da Figura 3.8.

$$X = \left\{ (000), (001), (010), (011), (100), (101), (110), (111) \right\}, \quad (3.69)$$

onde a função $f(\cdot, \cdot)$ é representada pelo codificador convolucional.

3.7.1. Códigos Convolucionais sobre um Canal "BSC"

Vamos adotar um canal do tipo "BSC" para o código convolucional em análise, como indicado na Figura 3.10. Assim, o alfabeto de entrada E e o alfabeto de saída W do canal são

$$E = W = \{0, 1\}. \quad (3.70)$$

A probabilidade de transição do canal "p" assume tipicamente valores em $(0, 1/2)$ e o sinal que alcança o receptor é do tipo:

$$y_k = h(x_k, n_k) = x_k + n_k = \begin{pmatrix} x_{k_1} & x_{k_2} & x_{k_3} \end{pmatrix} + \begin{pmatrix} n_{k_1} & n_{k_2} & n_{k_3} \end{pmatrix} \quad (3.71)$$

onde n_{k_i} é uma variável aleatória de Bernoulli com média "p". Devemos notar que os alfabetos X e Y coincidem com uma extensão dos alfabetos de entrada e de saída do canal:

$$X = Y = E^3 = W^3. \quad (3.72)$$

De acordo com a Figura 3.10, a probabilidade condicional é dada por

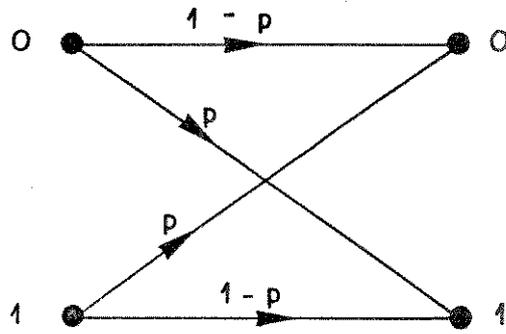


Figura 3.10 - Canal do Tipo "BSC".

$$\begin{aligned}
 p(y_k | x_k) &= \prod_{n=1}^3 p(y_{k_n} | x_{k_n}) \\
 &= p^{d_H(y_k, x_k)} (1-p)^{[3-d_H(y_k, x_k)]}
 \end{aligned}
 \tag{3.73}$$

Se adotarmos a métrica de máxima verossimilhança teremos

$$\begin{aligned}
 m(x_k, y_k) &= \log_e p(y_k | x_k) \\
 &= d_H(y_k, x_k) \log_e p + [3 - d_H(y_k, x_k)] \log_e (1-p) \\
 &= 3 \log_e (1-p) - d_H(y_k, x_k) \log_e \left(\frac{1-p}{p} \right).
 \end{aligned}
 \tag{3.74}$$

Na equação (3.74) notamos que se $p < 1/2$ o primeiro termo bem como o logaritmo no segundo termo são constantes positivas. Portanto, maximizar a métrica $\log_e p(y_k | x_k)$ é equivalente a minimizar $d_H(y_k, x_k)$. Daí é conveniente e igualmente válido adotar a métrica:

$$m(x_k, y_k) = -d_H(y_k, x_k).
 \tag{3.75}$$

Substituindo a métrica em (3.75) na equação (3.14) ficamos com

$$D(\lambda, \tilde{x}_k, x_k) = E \left\{ \exp \left[\lambda \left\{ d_H(y_k, x_k) - d_H(y_k, \tilde{x}_k) \right\} \right] \right\},
 \tag{3.76}$$

onde x_k é o sinal transmitido. Em função das coordenadas dos sinais a equação acima se torna

$$\begin{aligned}
 D(\lambda, \tilde{x}_k, x_k) &= E \left\{ \exp \left[\lambda \left\{ d_H(y_k, x_k) - d_H(y_k, \tilde{x}_k) \right\} \right] \right\} \\
 &= \prod_{n=1}^3 E \left\{ \exp \left[\lambda \left\{ d_H(y_{k_n}, x_{k_n}) - d_H(y_{k_n}, \tilde{x}_{k_n}) \right\} \right] \right\}, \quad (3.77)
 \end{aligned}$$

onde usamos a independência dos símbolos y_{k_n} na segunda passagem. Devemos notar em (3.77) que se $x_{k_n} = \tilde{x}_{k_n}$, o argumento do operador de esperança matemática nesta equação é unitário. Caso contrário, este assume um valor que independe de "n", uma vez que as variáveis aleatórias y_{k_n} são identicamente distribuídas. A partir destas observações, (3.77) pode ser reescrita como

$$\begin{aligned}
 D(\lambda, \tilde{x}_k, x_k) &= \left[E \left\{ \exp \left[\lambda \left\{ d_H(y_{k_1}, x_{k_1}) - d_H(y_{k_1}, \tilde{x}_{k_1}) \right\} \right] \right\} \right]^{d_H(x_k, \tilde{x}_k)} \\
 &= \left[P(y_{k_1} = x_{k_1}) \exp(-\lambda) + P(y_{k_1} \neq x_{k_1}) \exp(\lambda) \right]^{d_H(x_k, \tilde{x}_k)} \\
 &= \left[(1 - p) \exp(-\lambda) + p \exp(\lambda) \right]^{d_H(x_k, \tilde{x}_k)} \quad (3.78)
 \end{aligned}$$

Minimizando (3.78) com relação a λ esta se torna

$$\begin{aligned}
 D \left(\frac{1}{2} \log_e \left[\frac{(1-p)}{p} \right], \tilde{x}_k, x_k \right) &= \left[2\sqrt{p(1-p)} \right]^{d_H(x_k, \tilde{x}_k)} \\
 &= D^{d_H(x_k, \tilde{x}_k)}, \quad (3.79)
 \end{aligned}$$

onde $D = 2\sqrt{p(1-p)}$.

Com o propósito de determinar os rótulos a_{ij} dados em (3.60), e a partir destes avaliar a função de transferência, vamos considerar a seguinte medida de distorção, que leva à probabilidade de erro de bit:

Tabela 3.1 - Distorção de Erro.

		\tilde{u}_k			
		e_1	e_2	e_3	e_4
$d((s_k, u_k), (\tilde{s}_k, \tilde{u}_k))$		00	01	10	11
u_k	00	0	β	α	$\alpha+\beta$
	01	β	0	$\alpha+\beta$	α
	10	α	$\alpha+\beta$	0	β
	11	$\alpha+\beta$	α	β	0

Percebe-se facilmente que a escolha $\alpha = 1, \beta = 0$ leva a probabilidade de erro de bit da primeira entrada do codificador. A correspondente medida para a segunda entrada é conseguida escolhendo-se $\alpha = 0, \beta = 1$. A probabilidade de erro de bit média é fornecida pela escolha $\alpha = \beta = \frac{1}{2}$.

A partir da Tabela 3.1 e do valor de D na equação (3.7a) os rótulos a_{ij} seguem naturalmente e destes a matriz \underline{A} e os vetores \underline{b} e \underline{c} :

$$\underline{A} = \begin{bmatrix} D^3 I^\beta & D^3 I^\beta & D I^\beta \\ D I^\alpha & D I^\alpha & D I^\alpha \\ D^2 I^{\alpha+\beta} & I^{\alpha+\beta} & D^2 I^{\alpha+\beta} \end{bmatrix}, \quad (3.80)$$

$$\underline{b} = \begin{bmatrix} DI^\beta \\ D^3 I^\alpha \\ D^2 I^{\alpha+\beta} \end{bmatrix}, \quad \underline{c} = \begin{bmatrix} D^2 \\ D^2 \\ D^2 \end{bmatrix}. \quad (3.81)$$

As probabilidades de erro de bit mencionadas acima podem ser obtidas diretamente de

$$P_b(\alpha, \beta) \leq \frac{\partial}{\partial I} T_0(D, I) \Bigg|_{\substack{D=2\sqrt{p(1-p)} \\ I=1}} \quad (3.82)$$

onde a derivada é calculada como indicado em (3.65) e a dependência da função $T_0(I)$ com relação a D segue diretamente de (3.80) e (3.81).

Na maioria dos casos práticos, como naquele tratado no Capítulo 5, o interesse maior é na probabilidade de erro de bit média. Nesta situação particular é conveniente definir a medida de distorção como sendo

$$d\left((s_k, u_k), (\tilde{s}_k, \tilde{u}_k)\right) = d_H(u_k, \tilde{u}_k), \quad (3.83)$$

que é conseguido fazendo $\alpha = \beta = 1$ na Tabela 3.1. Com isto a probabilidade de erro de bit média é

$$P_b \leq \frac{1}{2} \frac{\partial}{\partial I} T(D, I) \Bigg|_{\substack{D=2\sqrt{p(1-p)} \\ I=1}} \quad (3.84)$$

Uma outra medida importante e que é utilizada no Capítulo 5, é a probabilidade de erro de bit codificado. Notamos que, enquanto os expoentes da va-

riável I em $T_0(D, I)$ indicam o número de bits de informação não nulos, o número de bits codificados não nulos tem relação semelhante com respeito à variável D . Deste raciocínio, a probabilidade de bit codificado é facilmente obtida pela expressão

$$P_c = \frac{1}{3} D \frac{\partial}{\partial D} T_0(D, I) \Bigg|_{\substack{D=2\sqrt{p(1-p)} \\ I=1}} \quad (3.85)$$

onde o produto por D serve para recompor os expoentes desta variável na função $T_0(D, I)$, que é decrementado de uma unidade depois da derivada. O fator $1/3$ é justificado de maneira semelhante a do fator $1/2$ em (3.84). A derivada em D é obtida por (3.65) com $(\cdot)'$ agora indicando $\frac{\partial(\cdot)}{\partial D}$.

3.7.2. Códigos convolucionais sobre um canal "AWGN"

Nesta seção, o canal que vamos considerar é do tipo gerado pela modulação BPSK (do inglês: Binary Phase Shift Keying) com ruído AWGN de média nula e variância

$$E\left\{n_k^2\right\} = \sigma^2. \quad (3.86)$$

Neste caso o alfabeto de entrada do canal é mantido idêntico ao do BSC (Seção 3.7.1), enquanto que o alfabeto de saída é do tipo $W = (-\infty, \infty)$. Isto caracteriza uma saída contínua, o que permite uma decisão suave (do inglês: Soft

Decision) no receptor. As equações (3.71) e (3.72) também se aplicam ao caso em análise.

A probabilidade condicional é descrita por

$$p(y_k | x_k) = \prod_{n=1}^3 \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_{k_n} - \bar{x}_{k_n})^2\right\}, \quad (3.87)$$

onde a nova variável \bar{x}_{k_n} é definida como

$$\bar{x}_{k_n} = \begin{cases} +\sqrt{E_s} & ; \text{ se } x_{k_n} = 0 \\ -\sqrt{E_s} & ; \text{ se } x_{k_n} = 1 \end{cases}, \quad (3.88)$$

sendo E_s a energia do sinal modulado. A partir de (3.87) a métrica de máxima verossimilhança, na sua forma mais simples, é dada por [1]

$$m(x_k, y_k) = \sum_{n=1}^3 y_{k_n} \cdot \bar{x}_{k_n}. \quad (3.89)$$

Substituindo a métrica acima na equação (3.14) esta se torna

$$D(\lambda, \tilde{x}_k, x_k) = E\left\{\exp\left[\lambda \sum_{n=1}^3 \left\{y_{k_n} (\tilde{x}_{k_n} - \bar{x}_{k_n})\right\}\right]\right\}, \quad (3.90)$$

onde x_k (ou \bar{x}_k) é o sinal transmitido. De maneira equivalente ao caso anterior, esta forma se reduz a

$$\begin{aligned}
D(\lambda, \tilde{x}_k, x_k) &= \left[E \left\{ \exp \left[\lambda \left(\bar{x}_{k_1} + z_{k_1} \right) \left(\tilde{x}_{k_1} - \bar{x}_{k_1} \right) \right] \right\} \right]_{\substack{\bar{x}_{k_1} \neq \tilde{x}_{k_1} \\ \bar{x}_{k_1} = \tilde{x}_{k_1}}} d_H(x_k, \tilde{x}_k) \\
&= \left[\exp(-2\lambda E_s) E \left\{ \exp \left(\pm 2\lambda \sqrt{E_s} n_{k_1} \right) \right\} \right] d_H(x_k, \tilde{x}_k), \tag{3.91}
\end{aligned}$$

onde o sinal + ou - depende do valor de \bar{x}_{k_1} . Esta esperança pode ser diretamente calculada a partir da conhecida "função característica" de uma variável aleatória Gaussiana ν com média nula e variância σ^2 a saber [3]

$$C(\varphi) = E(\exp(\varphi \nu)) = \exp\left\{\frac{\varphi^2 \sigma^2}{2}\right\}. \tag{3.92}$$

Substituindo (3.92) em (3.91) e minimizando esta com relação a λ obtemos

$$D\left(\frac{1}{2\sigma^2}, \tilde{x}_k, x_k\right) = \left[e^{-\frac{E_s}{2\sigma^2}} \right] d_H(x_k, \tilde{x}_k) = D d_H(x_k, \tilde{x}_k) \tag{3.93}$$

onde $D = e^{-\frac{E_s}{2\sigma^2}}$ e a escolha do sinal em (3.91) é irrelevante devido a (3.92).

Percebemos claramente que, a menos dos valores específicos de D , as equações (3.79) e (3.93) são idênticas. A medida de distorção deve ser a mesma do caso anterior e as probabilidades de erro de bit e de bit codificado também são fornecidas por (3.84) e (3.85), respectivamente, com o novo valor de D substituindo aquele dado por (3.79).

Na maioria dos casos práticos, e entre eles aquele estudado no Capítulo 5, esta situação se repete. O valor de D deve ser obtido para cada canal particular. No Capítulo 5, onde sistemas de modulação codificada multinível são abordados, o decodificador de cada estágio recebe o mesmo sinal y_k por canais estatisticamente diferentes, conduzindo a um valor de D para cada estágio.

Devemos encerrar este capítulo com a extensão das equações (3.84) e (3.85), obtidas para um codificador convolucional de taxa $R = 2/3$, para uma taxa genérica $R = b/n$. Neste caso geral as probabilidades de erro de bit e de bit codificado são fornecidas por

$$P_b \leq \frac{1}{b} \frac{\partial}{\partial I} T(D, I) \Bigg|_{\substack{D \\ I=1}} \quad (3.94)$$

e

$$P_c \leq \frac{1}{n} D \frac{\partial}{\partial D} T(D, I) \Bigg|_{\substack{D \\ I=1}}, \quad (3.95)$$

respectivamente, onde D depende do canal considerado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J.K. Omura and M.K. Simon, "Modulation/Demodulation Technique for Satellite Communications", J.P.L. Technical Report, 1981.
- [2] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding*, (McGraw-Hill, 1979).
- [3] H.L. Van Trees, *Detection, Estimation and Modulation Theory*, Part I, (Wiley, 1968).

CAPÍTULO 4

A MODULAÇÃO CODIFICADA

4.1. INTRODUÇÃO

Devido a crescente demanda de serviços que fazem uso das Comunicações Digitais, tem sido compensador o dispêndio de esforços por parte de pesquisadores de todo o mundo no sentido de tornar os sistemas de comunicação digital cada vez mais eficientes. Nesta direção, uma gama muito ampla de problemas matemáticos envoltos em situações de desafio constante surge, tornando esta tarefa de grande excitamento.

De uma maneira geral, em sistemas de comunicação digital, a potência de transmissão bem como a largura de faixa de frequência disponível são limitadas. Este fato conduz, entretanto, à motivação que leva os pesquisadores a elaborarem propostas eficientes, no sentido de transmitir mais informação por unidade de tempo na faixa de frequência disponível e com uma menor potência, ou, de outra maneira, transmitir mais informação com a potência disponível fazendo uso de uma menor faixa de frequência.

Um resultado proveitoso foi obtido com a utilização de modulação eficiente. Conseguia-se, então, um aumento na taxa de transmissão sem que a faixa espectral fosse alterada. Entretanto, um aumento de potência era requerido a fim de preservar a confiabilidade (imunidade ao ruído).

Fez-se, então, necessário a introdução de códigos corretores de erro a fim de se obter a confiabilidade desejada sem a necessidade do aumento de potência. Entretanto, a redundância introduzida pela codificação conduzia a um aumento da faixa de frequência, pelo fato de que mais bits estariam sendo transmitidos para a mesma informação.

Até então, a codificação e a modulação se davam de maneira independente, isto é, a busca do codificador ótimo se caracterizava por perseguir o

código com a maior distância de Hamming, sem que a modulação empregada a posteriori tivesse participação nesta busca. O único manifesto de dependência entre os dois processos supracitados se dava através da codificação de Gray.

Numa etapa posterior, Massey [1] mostrou que ao se tratar codificação e modulação como um processo único, ou de outra forma, ao se introduzir uma dependência entre estes dois processos, algum ganho em termos de desempenho poderia ser conseguido. Tirando proveito desta conjectura, Ungerboeck [2] lançou uma proposta de codificação de canal onde os bits redundantes são acomodados pelo emprego de uma constelação expandida, ou seja, com um número de sinais maior que o necessário para transmitir a mesma informação. Nesta proposta, um codificador convolucional é utilizado e os bits codificados são mapeados nos sinais da constelação expandida. Desta forma os sinais substituem as palavras de ramo na treliça. Esta técnica ficou conhecida como Modulação Codificada por Treliça (TCM, do inglês: Trellis Coded Modulation) por razões óbvias, e provê ganhos da ordem de até 6 dB (com complexidade irrestrita) de potência sem sacrificar a largura de faixa de frequência utilizada. A proposta de Ungerboeck é apresentada na Seção 4.2, pela sua importância histórica e por conter elementos básicos essenciais ao estudo principal deste trabalho.

Um outro tipo de código de treliça é baseado na Modulação por Deslocamento da Frequência com Fase Contínua (CPFSK, do inglês: Continuous Phase Frequency-Shift Keying). O fato da fase ser contínua neste tipo de modulação (em oposição aos saltos de fase no M-PSK, por exemplo) provê propriedades espectrais interessantes e impede que certas seqüências de sinais ocorram. Esta última característica resulta em uma maior distância Euclidiana entre as possíveis seqüências, tal qual ocorre nos códigos de Ungerboeck. Um aprofundamento em Modulação com Fase Contínua (CPM, do inglês: Continuous Phase Modulation) é conseguido em [13], [14].

Na Seção 4.3, apresentamos uma outra classe de Sistemas de Modulação Codificada: a Modulação Codificada Multinível. Para esta classe muito importante um esquema de decodificação multiestágio é convencionalmente empregado. Uma estratégia de decodificação baseada nesta estrutura e o desenvolvimento de um procedimento genérico para derivar um limitante superior para o seu desempenho é a proposta deste trabalho que é considerada no Capítulo 5.

Finalmente, na Seção 4.4, consideramos algumas formas de constelações que originaram novas classes de sistemas de modulação codificada, conceitualmente relacionadas com as classes supracitadas.

4.2. MODULAÇÃO CODIFICADA POR TRELIÇA

4.2.1. Conceitos Básicos

Nos sistemas de comunicação convencionais, em que o codificador e o modulador são descombinados, os bits codificados selecionam sinais de uma constelação previamente determinada. A redundância no codificador gera uma expansão na largura de faixa de frequência, uma vez que há um aumento no número de bits a serem transmitidos sem haver uma expansão da constelação.

Tal qual ocorre com o modulador e o codificador, o demodulador e o decodificador também são descombinados. Desta forma, o processo de demodulação se caracteriza pela quantização do sinal analógico recebido, o que fornece os bits demodulados. Numa etapa posterior, estes bits são decodificados da maneira convencional. Este procedimento é conhecido como decodificação por decisão abrupta (do inglês: *hard decision*) e conduz a uma perda irreversível de infor-

mação, que no caso binário se aproxima de 2 dB na relação sinal-ruído (SNR).

Em resposta a estas deficiências, Ungerboeck [2] lançou uma proposta de codificação (combinada com a modulação) que provê ganhos de desempenho sem sacrificar a largura de faixa espectral, mantendo constante a taxa de informação. Isto é conseguido às expensas de uma expansão da constelação, para acomodar os bits codificados. No seu estudo, limitou-se às constelações em uma e duas dimensões dando margem ao uso dos esquemas de modulação multinível (fase e/ou amplitude) tais como M-AM, M-PSK e M-QAM. Ainda considerou os canais com ruído do tipo AWGN, com faixa limitada e livres de interferência intersimbólica e quaisquer problemas de sincronismo.

A questão fundamental tratada na proposta de Ungerboeck, e a partir desta em todos os esquemas de modulação codificada, é que efetivamente sinais analógicos são transmitidos no canal e, desta maneira, o efeito do ruído seria menos danoso se possíveis pares de seqüências de sinais apresentassem grande distância Euclidiana quadrática (DEQ). Esta postura da modulação codificada difere daquela da codificação convencional, onde é desejado que as seqüências se mantenham fortemente afastadas segundo a distância de Hamming.

Usando o fato de que numa dada constelação (excetua-se alguns casos bem particulares tais como 2-PSK e 3-PSK) as DEQ entre pares de sinais podem ser diferentes segundo o par considerado, a codificação Ungerboeckiana pode ser definida como "o processo de impor certos padrões nos sinais transmitidos" de tal modo que apenas algumas seqüências de sinais da constelação expandida, determinadas pelo codificador, sejam permitidas. O conjunto de tais seqüências pode ser representado por uma estrutura de treliça, daí o nome: Modulação Codificada por Treliça. Um codificador convolucional é naturalmente escolhido para reger as seqüências de sinais.

Em síntese, o codificador de Ungerboeck pode ser visto como um codi-

ficador convolucional, seguido de um mapeamento que leva os bits codificados em sinais de uma constelação expandida. Desta forma, cada ramo na treliça será rotulado por um sinal desta constelação, e o código ótimo C_{UNG} é aquele que apresenta a máxima distância Euclidiana livre

$$d_{free}^2 = \min_{\substack{\{x_n\}, \{x'_n\} \in C_{UNG} \\ \{x_n\} \neq \{x'_n\}}} \left[\sum_n d_E^2(x_n, x'_n) \right], \quad (4.1)$$

Percebe-se facilmente que os códigos convolucionais binários ótimos conhecidos não são necessariamente a melhor escolha para o caso considerado. Isto se deve ao fato de que estes são projetados segundo a distância de Hamming, o que não traduz necessariamente a otimalidade segundo a distância Euclidiana quadrática. Na verdade, excetuando-se o caso das modulações 2-AM e 4-PSK (nesta quando rotulada segundo a codificação de Gray), não há uma relação monotônica entre as duas distâncias. Por conseguinte, uma nova busca ao codificador de Ungerboeck é requerida, agora tendo como critério de otimalidade a maximização da distância dada em (4.1). Alguns algoritmos que buscam tais codificadores são encontrados em Ungerboeck [2] e Markman [3].

4.2.2. Fator de Expansão da Constelação Via Capacidade de Canal

Na Seção 4.2.1 foi dito que a constelação utilizada por Ungerboeck deveria ser expandida com o propósito de acomodar os bits redundantes provenientes da codificação convolucional. Naturalmente o fator pelo qual a conste-

lação deve ser expandida depende diretamente do número de bits redundantes. Para um codificador convolucional binário de taxa $R = b/n$, por exemplo, devemos expandir a constelação por um fator de $2^{(n-b)}$. Assim cada b bits de informação são transmitidos por um dos 2^n sinais da constelação expandida, enquanto que uma constelação com 2^b sinais seria suficiente para transmitir os mesmos b bits sem codificação.

No seu estudo, Ungerboeck avaliou a capacidade de canal de um canal com ruído do tipo AWGN, para cada modulação em uma e duas dimensões por ele abordada. Com base nesta análise [2], foi mostrado que na transmissão de b bits por sinal com o emprego de uma constelação de 2^{b+1} sinais, quase todo ganho em termos da capacidade de canal era obtido. Como consequência imediata, o codificador convolucional a ser utilizado deve apresentar uma taxa igual a $R = \frac{b}{b+1}$ e a constelação original será sempre expandida por um fator de 2.

Deste fato segue que a estrutura do codificador de Ungerboeck deve ser como indicado no diagrama em blocos da Figura 4.1, onde o mapeamento do segundo bloco é tratado na seção que segue.

4.2.3. Mapeamento por Partição de Conjunto

Já é mais que notório a esta altura que os $b+1$ bits codificados devem ser mapeados em sinais da constelação expandida. No sentido de maximizar a distância dada em (4.1), a dupla codificador convolucional-mapeador deve ser escolhida convenientemente. Embora conjecturável a busca da dupla ótima como um processo único, Ungerboeck a trata separadamente estabelecendo um procedimento heurístico para a obtenção do mapeador ótimo, e, uma vez este determinado, a busca ao codificador convolucional ótimo é então realizada.

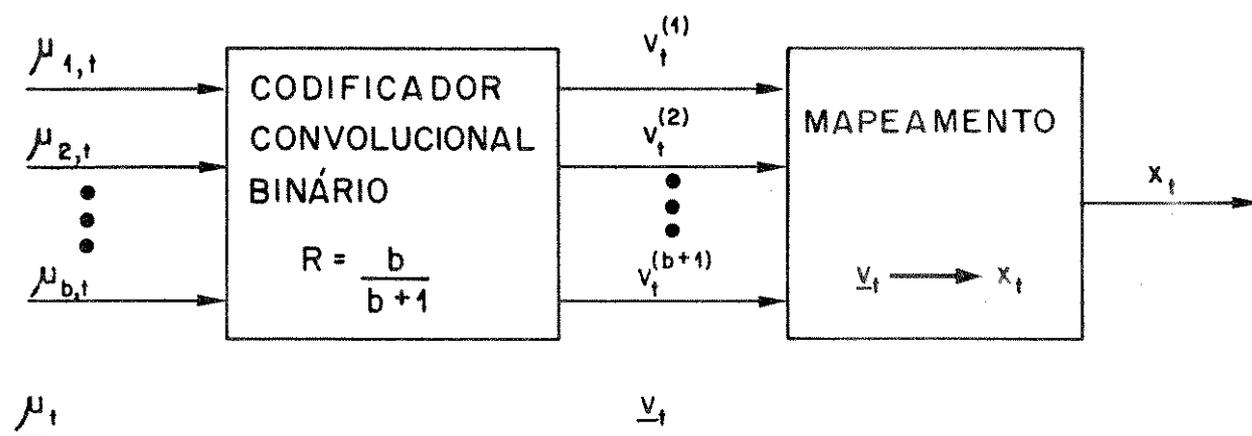


Figura 4.1 - Estrutura do Codificador de Ungerboeck.

A fim de compreender o procedimento adotado por Ungerboeck para a obtenção do mapeador ótimo, uma formulação é introduzida a seguir.

Seja $S = \{S_0, S_1, \dots, S_{M-1}\}$ um conjunto de M sinais no espaço Euclidiano.

Definição 4.2.3.1 : O número de elementos de S , denotado por $|S|$, é chamado a ordem do conjunto S .

Definição 4.2.3.2 : O conjunto S é denominado binário se $|S| = 2^\Psi$, onde Ψ é um inteiro positivo.

Definição 4.2.3.3 : A distância mínima $d(S)$ do conjunto S é a medida:

$$d(S) = \min_{\substack{S_i, S_j \in S \\ i \neq j}} \left\{ d_E(S_i, S_j) \right\}$$

Definição 4.2.3.4 : Uma partição de S , denotada por S/H , é caracterizada por um conjunto de $|S/H|$ subconjuntos disjuntos $H(a)$, $a \in \{0, 1, \dots, |S/H|-1\}$ cuja união é S . O valor $|S/H|$ é denominado a ordem da partição S/H .

Definição 4.2.3.5 : A partição S/H é denominada binária se $|S/H| = 2$.

Definição 4.2.3.6 : A distância intrasubconjunto δ associada à partição S/H é a medida:

$$\delta = \min_{\substack{a \in \{0, 1, \dots, |S/H|-1\} \\ S', S'' \in H(a) \\ S' \neq S''}} \left\{ d_E(S', S'') \right\}.$$

Definição 4.2.3.7 : A partição S/H é denominada eqüitativa se todos os subconjuntos H(a) possuem a mesma ordem $|S|/|S/H|$ e a mesma distância mínima δ .

Considerando a partição S/H do conjunto S, deveremos estabelecer uma nova partição em cada um dos $|S/H|$ subconjuntos, e repetir o processo até que cada subconjunto seja um e somente um elemento de S. As definições subseqüentes dizem respeito a esta situação.

Definição 4.2.3.8 : Uma cadeia de partição de Ψ níveis $S \equiv S^{(0)}/S^{(1)}/\dots/S^{(\Psi)}$ é caracterizada por sucessivas partições do conjunto $S^{(0)}$. Isto é, o conjunto $S^{(0)}$ é primeiramente dividido em $|S^{(0)}/S^{(1)}(a_0)|$ subconjuntos $S^{(1)}(a_0)$, onde a distância intrasubconjunto é δ_1 . Cada um destes, em um segundo nível, é subdividido em $|S^{(1)}(a_0)/S^{(2)}(a_0, a_1)|$ subconjuntos $S^{(2)}(a_0, a_1)$, onde a distância intrasubconjunto é δ_2 . E assim por diante até atingir o Ψ -ésimo nível, onde cada subconjunto $S^{(\Psi)}(\underline{a})$, $\underline{a} = (a_0, a_1, \dots, a_{\Psi-1})$, contém um e somente um elemento de $S^{(0)}$.

Definição 4.2.3.9 : Uma cadeia de partição de Ψ níveis $S \equiv S^{(0)}/S^{(1)}/\dots/S^{(\Psi)}$ é dita ser binária se todas as Ψ partições envolvidas são binárias.

Definição 4.2.3.10 : Uma cadeia de partição de Ψ níveis $S \equiv S^{(0)}/S^{(1)}/\dots/S^{(\Psi)}$ é dita ser eqüitativa se todas as Ψ partições envolvidas são eqüitativas.

Em termos das definições acima devemos caracterizar o "mapeamento por partição de conjunto", adotado por Ungerboeck, da maneira que segue:

- 1) A constelação deve ser um conjunto $S \equiv S^{(0)}$ binário cuja ordem é 2^Ψ .
- 2) A partir de $S^{(0)}$, estabelece-se uma cadeia de partição de Ψ níveis $S \equiv S^{(0)}/S^{(1)}/\dots/S^{(\Psi)}$ com as seguintes características:
 - a cadeia deve ser binária e eqüitativa;
 - as distâncias intrasubconjunto δ_i , $i=0,1,\dots,\Psi$, associadas às partições envolvidas devem satisfazer:

$$\delta_0 < \delta_1 < \dots < \delta_\Psi, \quad (4.2)$$

onde δ_0 é a distância mínima de $S^{(0)}$ e $\delta_\Psi = \infty$ por definição.

- 3) O vetor $\underline{a} = (a_0, a_1, \dots, a_{\Psi-1})$, $a_j \in \{0, 1\}$, da Definição 4.2.3.8 é naturalmente escolhido para rotular os subconjuntos $S^{(\Psi)}(\underline{a})$, ou mais precisamente, os subconjuntos no j -ésimo nível são rotulados pelas j primeiras coordenadas do vetor \underline{a} . Portanto, no Ψ -ésimo nível, cada sinal de $S^{(0)}$ é rotulado pelo vetor \underline{a} completo $S^{(\Psi)}(a_0, a_1, \dots, a_{\Psi-1})$, isto é, cada Ψ -upla binária $\underline{a} = (a_0, a_1, \dots, a_{\Psi-1})$ é mapeada no sinal $S^{(\Psi)}(\underline{a})$ da constelação $S^{(0)}$, daí o nome "mapeamento por partição de conjunto".

A solução do passo 2 não é única (se existir), e, adicionalmente, os rótulos a_j podem ser permutados entre os subconjuntos das partições gerando outros mapeamentos. Entretanto, estes são equivalentes, desde que a condição das distâncias intrasubconjunto crescentes seja satisfeita e apresente o mesmo padrão. Esta condição em (4.2) é indubitavelmente o ponto mais importante no mapeamento por partição de conjunto e conduz ao que Forney [4] chamou de Lema da distância de Partição, cuja prova é perfeitamente dispensável se os conceitos introduzidos nesta seção foram entendidos.

Lema 4.2.3.1 (Lema da Distância de Partição [4]) : Se $S^{(0)}/S^{(1)}/\dots/S^{(\Psi)}$ é uma cadeia de partições de Ψ níveis cujas distâncias δ_i satisfazem (4.2) segundo Ungerboeck, e \underline{a} e \underline{a}' são os rótulos dos subconjuntos (elementos de $S^{(0)}$) $S^{(\Psi)}(\underline{a})$ e $S^{(\Psi)}(\underline{a}')$, respectivamente, então

$$d_E\left(S^{(\Psi)}(\underline{a}), S^{(\Psi)}(\underline{a}')\right) \geq \delta_j,$$

onde

$$j = \min_{0 \leq i < \Psi} \{i : a_i \neq a'_i\}.$$

Esta propriedade é fundamentalmente importante quando a Modulação Codificada Multinível (Seção 4.3) é considerada.

A fim de ilustrar o procedimento adotado na partição e mapeamento de Ungerboeck, dois exemplos clássicos são fornecidos. A Figura B.1 (veja Apêndice B) considera a constelação circular 8-PSK, enquanto que na Figura B.2 (veja Apêndice B) a partição se dá sobre a constelação retangular 16-QAM. Nestas fizemos $C_{i+1} = a_i$ por questão de conveniência, segundo o desenvolvimento do Capítulo 5.

As condições impostas por Ungerboeck sobre a partição são muito restritivas. Alguns sistemas de modulação codificada [8], [12] empregam partições (e também constelações) não binárias, o que requer a adoção de códigos não binários como consequência. Algumas considerações sobre constelações e partições são feitas na Seção 4.4

4.2.4. Exemplo e Decodificação de Sistemas TCM

Nesta seção fornecemos um exemplo de um codificador de Ungerboeck e fazemos alguns comentários sobre decodificação e avaliação de desempenho dos sistemas TCM.

Exemplo 4.1 :Vamos considerar a constelação 8-PSK da Figura B.1 (veja Apêndice B). O codificador de Ungerboeck deve ser do tipo convolucional com taxa $R = \frac{2}{3}$, segundo a Seção 4.2.2, e portanto, uma transmissão de 2 bits/sinal é alcançada. A Figura 4.2 mostra o codificador ótimo para 4 estados ($K=2$) e a correspondente treliça. Nesta, a distância Euclidiana livre é limitada pelos sinais das transições paralelas:

$$d_{free}^2 = d_E^2(S_0, S_4) = 4. \quad (4.3)$$

A partir do exemplo acima algumas observações relevantes devem ser mencionadas:

- 1) Devemos notar que a segunda entrada do codificador é não codificada. Devido à condição (4.2), podemos em geral deixar algumas entradas sem codificação

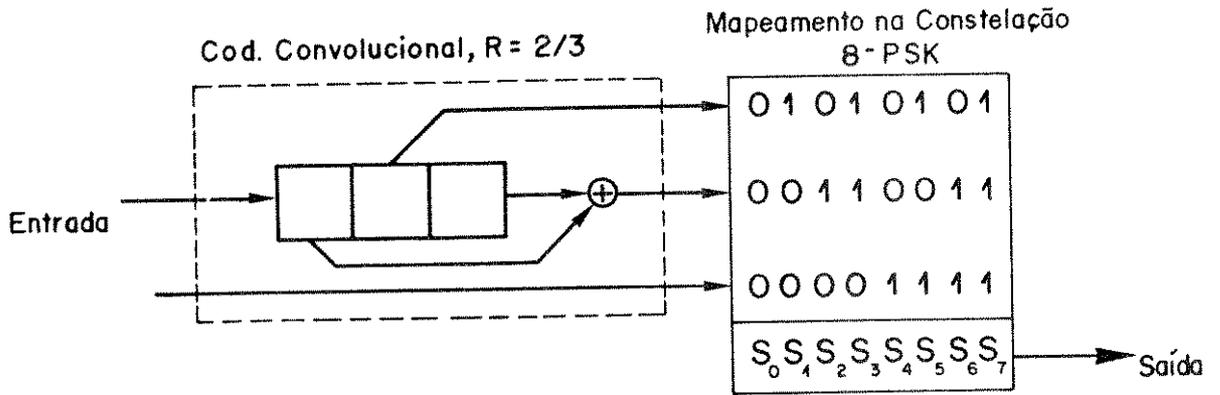
sem que a distância livre seja diminuída. Isto porque para uma certa partição, digamos $\Psi' \leq \Psi$, a distância intrasubconjunto $\delta_{\Psi'}$ é grande o suficiente para garantir a distância livre acima de um determinado valor. Deste modo, para transmitir Ψ bits/sinal, é suficiente codificar $\tilde{\Psi} \leq \Psi$ bits por meio de um codificador convolucional de taxa $R = \tilde{\Psi}/(\tilde{\Psi} + 1)$. Os $\tilde{\Psi} + 1$ bits codificados seleccionam um dos $2^{\tilde{\Psi}+1}$ subconjuntos da $(\tilde{\Psi} + 1)$ -ésima partição. Os $\Psi - \tilde{\Psi}$ bits de informação restantes especificam um dos $2^{\Psi-\tilde{\Psi}}$ sinais pertencentes ao subconjunto seleccionado no passo anterior.

- 2) Percebemos facilmente que o codificador convolucional da Figura 4.2, apesar de ser ótimo segundo a distância Euclidiana, apresenta uma distância livre de Hamming unitária. Isto o classifica como um péssimo codificador, uma vez que pode-se conseguir um $d_{free} = 3$ (ver codificador da Figura 3.8), segundo a distância de Hamming.

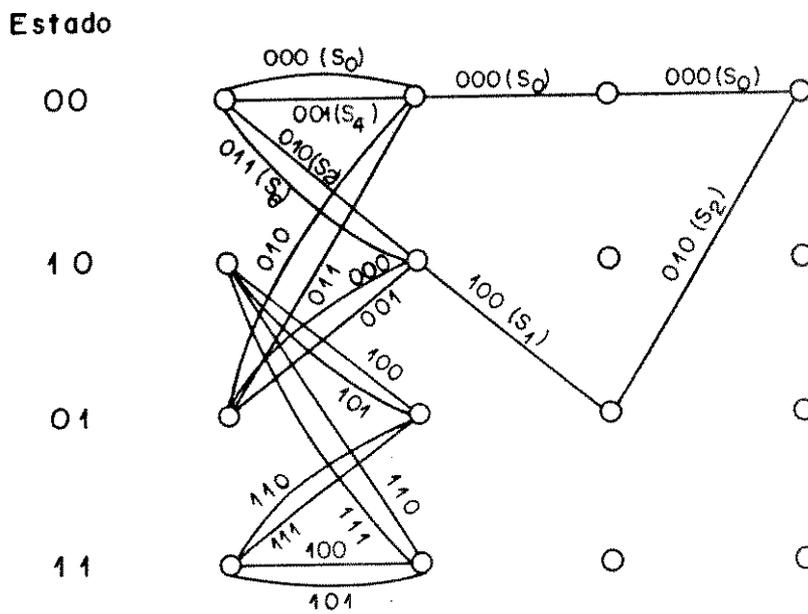
No que concerne ao processo de decodificação dos sistemas TCM, devemos perceber que estes seguem o modelo de estado finito descrito no Capítulo 3. Nestes termos, as seqüências de sinais naqueles sistemas devem ser estimadas pelo Algoritmo de Viterbi, onde a métrica de máxima verossimilhança é adotada. Para este caso particular, a métrica se reduz à forma equivalente:

$$m(x_k, y_k) = -d_E^2(x_k, y_k). \quad (4.4)$$

Por outro lado, o desempenho dos sistemas TCM não é avaliado com facilidade. Primeiramente pelo fato da não linearidade que é, em geral, uma característica destes sistemas. Como consequência o número de estados ("super-estados") a ser considerado é igual ao quadrado do que é necessário em sistemas onde a linearidade está presente, como apresentado na Seção 3.6. E, finalmente,



(a)



(b)

Figura 4.2 - (a) Codificador de Ungerboeck de 4 Estados para 8-PSK
(b) Treliça.

o procedimento do Capítulo 3 se restringe a sistemas sem transições paralelas (Seção 3.2), que geralmente não é o caso dos sistemas TCM.

Zehavi e Wolf [15] analisaram o desempenho dos sistemas TCM através de uma Função de Transferência modificada, que segue o procedimento do Capítulo 3. Entretanto, tal análise se restringia a uma classe particular destes sistemas.

Duas medidas um tanto quanto desairosas, mais que são amplamente utilizadas para avaliar o desempenho dos sistemas TCM são:

1) O Ganho de Codificação Assintótico¹:

$$G_a = 10 \log_{10} \left(\frac{d_{free}^2}{d_{ref}^2} \right), \quad (4.5)$$

onde d_{ref} é a distância Euclidiana mínima de um esquema de modulação não codificado operando com a mesma energia média.

2) A Probabilidade de Erro de Evento Assintótico:

$$\Pr(e) \leq N(d_{free}) \cdot Q \left(\frac{d_{free}}{2\sigma} \right), \quad (4.6)$$

onde $N(d_{free})$ é o número médio de eventos de erro com distância d_{free} , σ^2 é a potência (variância) do ruído por dimensão e $Q(\cdot)$ é a função de probabilidade de erro gaussiana [16]:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \cdot \int_x^{\infty} \exp \left(-\frac{t^2}{2} \right) dt. \quad (4.7)$$

¹O termo assintótico corresponde a $SNR \rightarrow \infty$.

4.3. MODULAÇÃO CODIFICADA MULTINÍVEL

4.3.1. Conceitos Básicos

Antes de Ungerboeck ter introduzido o conceito de "mapeamento por partição de conjunto" e de ter estabelecido de fato a dependência entre codificação e modulação, Imai e Hirakawa [5] propuseram um método de codificação multinível (ou multicamadas) que originou posteriormente uma classe importante de Sistemas de Modulação Codificada.

A Modulação Codificada Multinível [6]-[12] tem recebido grande destaque entre tais sistemas, devido ao fato de apresentar uma relativa facilidade de projeto e permitir o uso de uma estratégia de decodificação multiestágio, que é subótima, mas apresenta vantagens na relação desempenho/complexidade sobre os sistemas TCM da Seção 4.2.

Seguindo o esquema de Imai e Hirakawa, um diagrama em blocos geral de um sistema de modulação codificada de Ψ níveis é mostrado na Figura 4.3.

Um bloco de informação binária I é particionado em Ψ blocos I_1, I_2, \dots, I_Ψ , denominados blocos componentes. Cada bloco componente I_i tem comprimento k_i e é codificado pelo codificador correspondente E_i , que produz o bloco codificado C_i . A taxa do i -ésimo codificador, definida como $R_i = \frac{k_i}{n_i}$, é determinada segundo alguns critérios a serem definidos.

Em um certo instante t (t um inteiro), um vetor binário $v^{(t)} = (C_1^{(t)}, C_2^{(t)}, \dots, C_\Psi^{(t)})$ é formado pelas componentes dos Ψ blocos codificados no instante t , $C_i^{(t)}$, $i=1,2,\dots,\Psi$. Na Figura 4.3 este corresponde a uma coluna na matriz que se forma nas saídas dos codificadores E_i . Através do mapeamento por partição de conjunto, o vetor $v^{(t)}$ é mapeado em um sinal de uma constelação contendo 2^Ψ sinais, X , o qual é seguidamente transmitido por um canal ruidoso. O exemplo que segue ilustra melhor este processo de codificação.

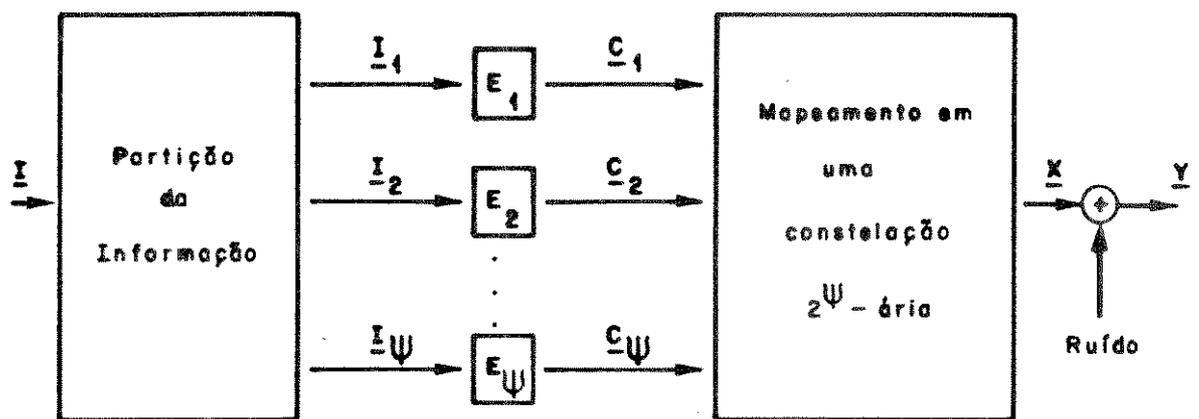


Figura 4.3 - Sistema de Modulação Codificada de Ψ Níveis.

Exemplo 4.2 :Vamos considerar um Sistema de Modulação Codificada de 3 níveis e a constelação 8-PSK, particionada na Figura B.1 (Apêndice B). Assim, se as palavras código são:

$$\begin{array}{|c|c|c|c|} \hline C_1^{(t)} & C_1^{(t+1)} & C_1^{(t+2)} & \dots \\ \hline C_2^{(t)} & C_2^{(t+1)} & C_2^{(t+2)} & \dots \\ \hline C_3^{(t)} & C_3^{(t+1)} & C_3^{(t+2)} & \dots \\ \hline \end{array} \equiv \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & \dots \\ \hline 1 & 0 & 0 & \dots \\ \hline 1 & 1 & 0 & \dots \\ \hline \end{array} ,$$

os vetores formados pelas componentes das palavras código são dados por:

$$\nu^{(t)} = (0, 1, 1); \quad \nu^{(t+1)} = (1, 0, 1); \quad \nu^{(t+2)} = (1, 0, 0); \dots$$

Seguindo o mapeamento por partição de conjunto da Figura B.1, a sequência de sinais transmitida será então:

$$\begin{array}{cccc} \nu^{(t)} & \nu^{(t+1)} & \nu^{(t+2)} & \dots \\ \downarrow & \downarrow & \downarrow & \\ S_6 & S_5 & S_1 & \end{array}$$

A questão fundamental neste sistema é que os bits codificados do i -ésimo nível selecionam subconjuntos da i -ésima partição, que está associada a uma distância intrasubconjunto δ_i .

A distância Euclidiana mínima é facilmente determinada quando consideramos os códigos componentes como sendo de bloco e de mesmo comprimento [6], isto é, $E_i = (n, k_i, d_i)$, onde d_i é a distância mínima do i -ésimo código. Neste caso, a cada bloco de informação \underline{I} contendo $\sum_{i=1}^{\Psi} k_i$ bits deve corresponder uma matriz código de ordem $\Psi \times n$, formada pelas palavras código dos códigos componentes. Assim sendo, uma matriz é chamada de matriz código se e somente se cada linha é uma palavra código do código correspondente.

Se considerarmos duas palavras código distintas no primeiro nível, o número de bits em que elas diferem é pelo menos d_1 . Como para cada posição em que elas diferem uma distância Euclidiana quadrática de pelo menos δ_1^2 separa os correspondentes sinais no espaço Euclidiano, teremos que a distância Euclidiana quadrática mínima da matriz código deve satisfazer.

$$d_M^2 \geq d_1 \delta_1^2.$$

Se as duas palavras código no primeiro nível são idênticas, as palavras código do segundo nível devem diferir (se distintas) em pelo menos d_2 bits. Sendo δ_2 a distância intrasubconjunto da segunda partição deveremos ter:

$$d_M^2 \geq d_2 \delta_2^2.$$

Raciocinando de maneira análoga nas demais partições, chegamos à distância Euclidiana quadrática mínima do código total como sendo:

$$d_{\min}^2 = \min \left(d_1 \delta_1^2, d_2 \delta_2^2, \dots, d_\Psi \delta_\Psi^2 \right). \quad (4.8)$$

O resultado dado em (4.8) também é válido quando códigos convolucionais são utilizados em cada partição, onde d_i é entendido como sendo a distância livre de Hamming do i -ésimo código. Nestes os comprimentos dos blocos n_i podem ser distintos e a matriz código pode ser considerada com um número infinito de colunas (para uma comparação com os códigos de blocos).

Uma característica dos sistemas de modulação codificada multinível de suma importância decorre da equação (4.8). A partir desta percebemos que, para uma dada constelação e partição, aumentar as distâncias de Hamming dos códigos componentes d_i traz, como consequência, um aumento na DEQ livre do sistema. Nestes termos, um bom sistema deve apresentar os códigos ótimos segundo a distância de Hamming, em oposição aos sistemas TCM onde uma nova busca é requerida.

A escolha das taxas deve ser feita segundo a eficiência espectral desejada:

$$R = \sum_{i=1}^{\Psi} R_i \quad \text{bits/sinal}, \quad (4.9)$$

e a dos códigos componentes para satisfazer uma DEQ livre, fornecida a partir da equação (4.8), determinada pelo projetista.

Em geral, de acordo com a Seção 4.2.2, a eficiência espectral é escolhida para ser $R = \Psi - 1$ bits/sinal, e os produtos $d_i \delta_i^2$ feitos aproximadamente constantes para uma proteção mais uniforme dos bits de informação. Para este último caso, os códigos das primeiras partições são naturalmente mais poderosos para compensar as pequenas distâncias intrasubconjunto. A medida que descemos na cadeia, códigos com menor capacidade de correção são requeridos.

Alguns exemplos de sistemas de modulação codificada multinível são fornecidos no Capítulo 5.

4.3.2. Decodificação Multiestágio

A própria estrutura multicamada sugere uma decodificação estágio por estágio, uma vez que a decodificação por máxima verossimilhança para este sistema é bastante complexa (sobretudo para $\Psi \geq 3$). Na Figura 4.4 é mostrado o circuito de decodificação multiestágio sugerido por Imai e Hirakawa [5].

No primeiro estágio, a sequência $\{Y\}$ entra na Unidade de Cômputo de Métrica (UCM). Esta unidade computa a métrica seguindo duas etapas. Na primeira etapa, o número de sinais a ser considerado no processo é reduzido para 2 (pois a partição é binária). Cada um destes sinais representa (designa) uma subconstelação da partição. Eles são escolhidos como os dois sinais mais próximos do

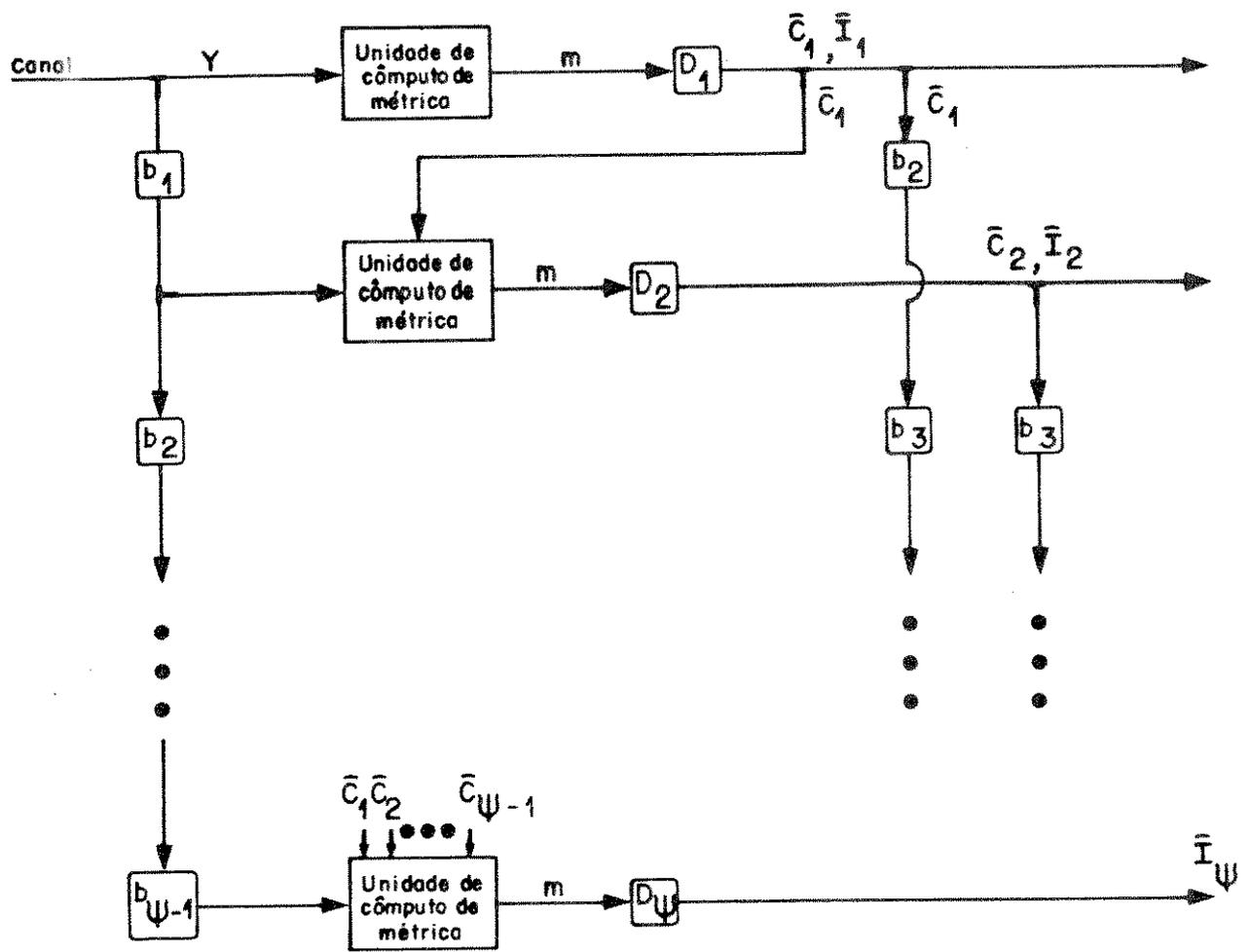


Figura 4.4 - Circuito de Decodificação.

sinal recebido Y no espaço Euclidiano, e constituem o que Pottie e Taylor [8] chamaram de "constelação básica". A segunda e última etapa é o próprio cômputo da métrica, o qual é realizado sobre a constelação básica. A sequência de métricas (m) obtida é então usada no decodificador D_1 , o qual produz a sequência código estimada $\{\hat{C}_1\}$. Esta por sua vez é utilizada como informação lateral indicando as subconstelações da primeira partição que deverão ser consideradas pelos estágios subsequentes. A sequência de informação estimada $\{\hat{I}_1\}$ é prontamente extraída da sequência $\{\hat{C}_1\}$. O canal visto pelo decodificador D_1 é simplesmente o canal Gaussiano.

No segundo estágio, e daí por diante, o mesmo procedimento é adotado para estimar a sequência código $\{\hat{C}_i\}$, baseada nas subconstelações designadas pela informações laterais $\{\hat{C}_j\}$, $j < i$. Nesta situação, o canal visto pelo decodificador D_i é o canal Gaussiano com a adicional incerteza das informações laterais. O tempo demandado na decodificação do i -ésimo estágio é compensado pela unidade de retardo b_i , que torna os sinais Y e \hat{C}_i disponíveis para os estágios subsequentes.

A partir do que foi dito fica claro que, se os códigos componentes são de bloco, então toda a palavra deve ser decodificada em um dado estágio para que o decodificador do estágio seguinte comece a operar. No caso de códigos convolucionais, todos os decodificadores operam continuamente, devido às razões fornecidas na Seção 3.3 sobre o truncamento da memória de caminho.

Esta estratégia de decodificação é inerentemente subótima uma vez que, se a informação lateral passada aos estágios subsequentes não for correta, estes últimos de maneira catastrófica cometerão erros de estimação com alta probabilidade. Por outro lado, a complexidade é fortemente diminuída quando comparada ao processo de decodificação por máxima verossimilhança (MLD). No caso dos códigos de bloco, por exemplo, o número de comparações (que é o número de matrizes código) na decodificação MLD é $2^{\sum k}$. Este número na decodificação

multiestágio cai para $\Sigma 2^{k_1}$. Quando códigos convolucionais são utilizados, o comprimento total de memória K (ou o número de estados) é considerado para a comparação. Em termos efetivos, a redução de complexidade transcende a subotimalidade, sobretudo para altas relações sinal-ruído.

4.4. ALGUMAS CONSIDERAÇÕES SOBRE CONSTELAÇÕES DE SINAIS

É evidente, a partir do que foi apresentado neste capítulo, que tanto a constelação como a partição desta ocupam um papel de grande importância nos sistemas de modulação codificada. O objetivo desta seção suplementar é fazer referências a algumas formas de constelações e partições que deram origem a novas classes de sistemas de modulação codificada, embora conceitualmente relacionadas com as classes descritas neste capítulo.

Uma primeira mostra é encontrada em [17] (Divsalar et.al.), onde se verifica que a introdução de uma conveniente assimetria nas constelações MPSK, M-AM e QAM proporciona um ganho no desempenho do sistema TCM correspondente.

Uma forma alternativa de constelação para sistemas TCM é proposta por Padovani e Wolf [18], onde uma combinação das modulações FSK e PSK bem como FSK e QAM é sugerida. Este casamento gera constelações em mais de duas dimensões. Alguns sistemas TCM (codificadores convolucionais) utilizando tais constelações foram projetados por Markman [3].

Calderbank e Sloane [19] constataram que uma constelação poderia ser considerada como um subconjunto finito de um reticulado n -dimensional (conjunto de pontos no espaço R^n que formam um grupo abeliano sobre a operação de adição [20]). Os subconjuntos que constituem a partição são designados para serem um

subreticulado e as suas classes laterais (cosets). Os sistemas TCM baseados nesta construção foram posteriormente chamados de códigos de classes laterais ("coset codes"), segundo Forney [4].

A mesma abordagem de reticulados/subreticulados foi feita para sistemas de modulação codificada multinível por Calderbank [7]. Posteriormente, Pottie e Taylor [8] chamaram de código de partição qualquer código que envolve partição de conjuntos, para a qual uma relação de equivalência sempre pode ser associada, onde os subconjuntos da partição são classes de equivalência distintas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J.L. Massey, "Coding and Modulation in Digital Communications", Proc. 1974 Int. Zurich Seminar on Digital Comm., Zurich, Switzerland, p. E2(1)-(4), Mar. 1974.
- [2] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", *IEEE Trans. Information Theory*, Vol. IT-28, pp. 55-67, Jan. 1982.
- [3] I. Markman, Esquemas de Modulação Codificada em Três e Quatro Dimensões - Tese de Mestrado, Universidade Estadual de Campinas, SP, Dez. 1989.
- [4] G.D. Forney Jr., "Coset Codes - Part I: Introduction and Geometrical Classification", *IEEE Trans. Information Theory*, Vol. 34, No. 5, pp. 1123-1151, Sep 1988.
- [5] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error Correcting Codes", *IEEE Trans. Information Theory*, Vol. IT-23, pp. 371-377, May 1977.
- [6] S.I. Sayegh, "A Class of Optimum Block Codes in Signal Space", *IEEE Trans. Communications*, Vol. COM-34, No. 10, pp. 1043-1045, Oct. 1986.
- [7] A.R. Calderbank, "Multilevel Codes and Multistage Decoding", *IEEE Trans. Communications*, Vol. 37, No. 3, pp. 222-229, Mar. 1989.
- [8] G.J. Pottle and D.P. Taylor, "Multilevel Codes Based on Partitioning", *IEEE Trans. Information Theory*, Vol. IT-35, No. 1, pp. 87-98, Jan 1989.

- [9] Y. Kofman, S. Shamai (Shitz) and E. Zehavi, "Analysis of a Multilevel Coded Modulation System", BILKNET'90, Ankara, Turkey, Jul. 1990.
- [10] B.F. Uchôa Filho and R. Palazzo Jr., "Performance Analysis of a Multistage Decoding Strategy for Multilevel Coded Modulation Systems", aceito no The Third IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Boston, Massachusetts, Oct. 1992.
- [11] B.F. Uchôa Filho and R. Palazzo Jr., "Performance Analysis of a Multistage Decoding Strategy for a Class of Geometrically Uniform Signal Sets", IEEE ITW92 - Information Theory Workshop 1992, Salvador, Bahia, Jun. 1992 e a ser submetido para publicação no *IEEE Trans. Communications*.
- [12] V.V. Ginzburg, "Multidimensional Signals for a Continuous Channel", *Problemy Peredachi Informatsii* [Problem on Information Transmission], Vol. 20, No. 1, pp. 28-46, Mar. 1984.
- [13] C.E. Sundberg, "Continuous Phase Modulation", *IEEE Communications Magazine*, Vol. 24, No. 4, pp. 25-38, Apr. 1986.
- [14] "Special Section on Modulation and Encoding", *IEEE Trans. Communications*, Vol. COM-29, pp. 185-297, Mar. 1981.
- [15] E. Zehavi and J.K. Wolf, "On the Performance Evaluation of Trellis Codes", *IEEE Trans. Information Theory*, Vol. 33, No. 2, pp. 196-202, Mar. 1987.
- [16] S. Haykin, *Digital Communications* (Wiley, 1988).
- [17] D. Divsalar, M.K. Simon and J.H. Yuen, "Trellis Coding with Asymmetric Modulations", *IEEE Trans. Communications*, Vol. COM-35, No. 2, pp. 130-141, Feb. 1987.

- [18] R. Padovani and J.K. Wolf, "Coded Phase/Frequency Modulation", *IEEE Trans. Communications*, Vol. COM-34, No. 5, pp. 446-453, May 1986.
- [19] A.R. Calderbank and N.J.A. Sloane, "New Trellis Codes Based on Lattices and Cosets", *IEEE Trans. Information Theory*, Vol. IT-33, No. 2, pp. 177-195, Mar. 1987.
- [20] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, New York, Springer-Verlag, 1988.

CAPÍTULO 5

A DECODIFICAÇÃO MULTISTÁGIO GENERALIZADA

5.1. INTRODUÇÃO

Na Seção 4.3, discorremos sobre os Sistemas de Modulação Codificada Multinível, que formam uma classe importante da Modulação Codificada. A Decodificação Multiestágio, para aqueles sistemas, foi referida como sendo um esquema alternativo que é subótimo, mas apresenta vantagens na relação desempenho/complexidade sobre a Decodificação por Máxima Verossimilhança. Esta subotimalidade surge a partir da incerteza da informação lateral passada aos estágios subsequentes.

Da forma que é sugerida por Imai e Hirakawa [1], a Decodificação Multiestágio (aqui referida como decodificação convencional) não envolve nenhum tratamento no sentido de reduzir o efeito da informação lateral errada. Kofman *et. al.* [2], apresentaram uma versão modificada do decodificador convencional, onde a amplitude da saída do canal Gaussiano é convenientemente limitada (por meio de um limitador suave) antes de serem processadas pelos estágios de decodificação. Esta limitação do sinal favorece a probabilidade de erro quando a informação lateral está errada. Por outro lado há um desfavorecimento para a situação oposta. Adicionalmente, esta estratégia se aplica apenas às constelações circulares (M-PSK). Naquele trabalho, apenas a constelação 8-PSK foi levada em consideração, onde foi conseguido um ganho (de aproximadamente 0.2 dB), para altas SNR, sobre o Código de Ungerboeck (TCM) para a mesma complexidade.

Nesta tese, propomos uma estratégia de decodificação [13] e [14] que apresenta um desempenho superior ao da decodificação convencional. Na verdade, a estratégia proposta gera uma classe de decodificadores multiestágio, da qual o decodificador multiestágio convencional, bem como o decodificador multiestágio ótimo (a ser determinado nas seções que seguem), são casos particulares. Na Seção 5.2, a estratégia proposta é apresentada. A análise de desempenho desta,

e conseqüentemente do decodificador convencional (uma vez que este é visto como um caso particular), é tratada na Seção 5.3. Nesta, expressões para a probabilidade de erro de bit (que levam em consideração o efeito da informação lateral errada) para as constelações 2^{τ} -PSK e 4^{τ} -QAM são fornecidas. Finalmente dois exemplos são encontrados na Seção 5.4.

5.2. A ESTRATÉGIA DE DECODIFICAÇÃO PROPOSTA

Na seção que segue, apresentamos a estratégia de decodificação proposta, mencionada na Seção 5.1. A idéia básica nesta proposta é considerar a informação lateral dos estágios anteriores, quando em erro, como um sinal confundidor (do inglês: jamming signal). Deste modo, o canal visto a partir de cada estágio (exceto pelo primeiro) pode ser modelado segundo a Figura 5.1, onde a variável z representa o estado do canal, que pode assumir os valores $z=0$, indicando que o sinal confundidor não está presente (ou que a informação lateral está correta); ou $z=1$, indicando a situação contrária. O decodificador correspondente é então tratado segundo uma analogia aos sistemas "anti-jam" considerados em [3].

O circuito de decodificação proposto é mostrado na Figura 5.2, o qual apresenta a mesma estrutura básica do decodificador convencional da Figura 4.4, a menos da nova unidade que estima* o estado do canal (UEC). O procedimento de decodificação no primeiro estágio segue da mesma maneira do decodificador convencional. No segundo estágio, e daí por diante, devemos considerar o estado do canal estimado \hat{z} , fornecido pela UEC. A partir desta informação, um custo $c(\hat{z})$

* Embora este seja reconhecido como um problema de detecção usaremos este termo.

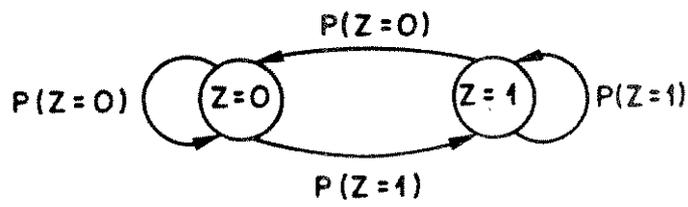


Figura 5.1 - Modelo de um Canal com Confundidor (jammer).

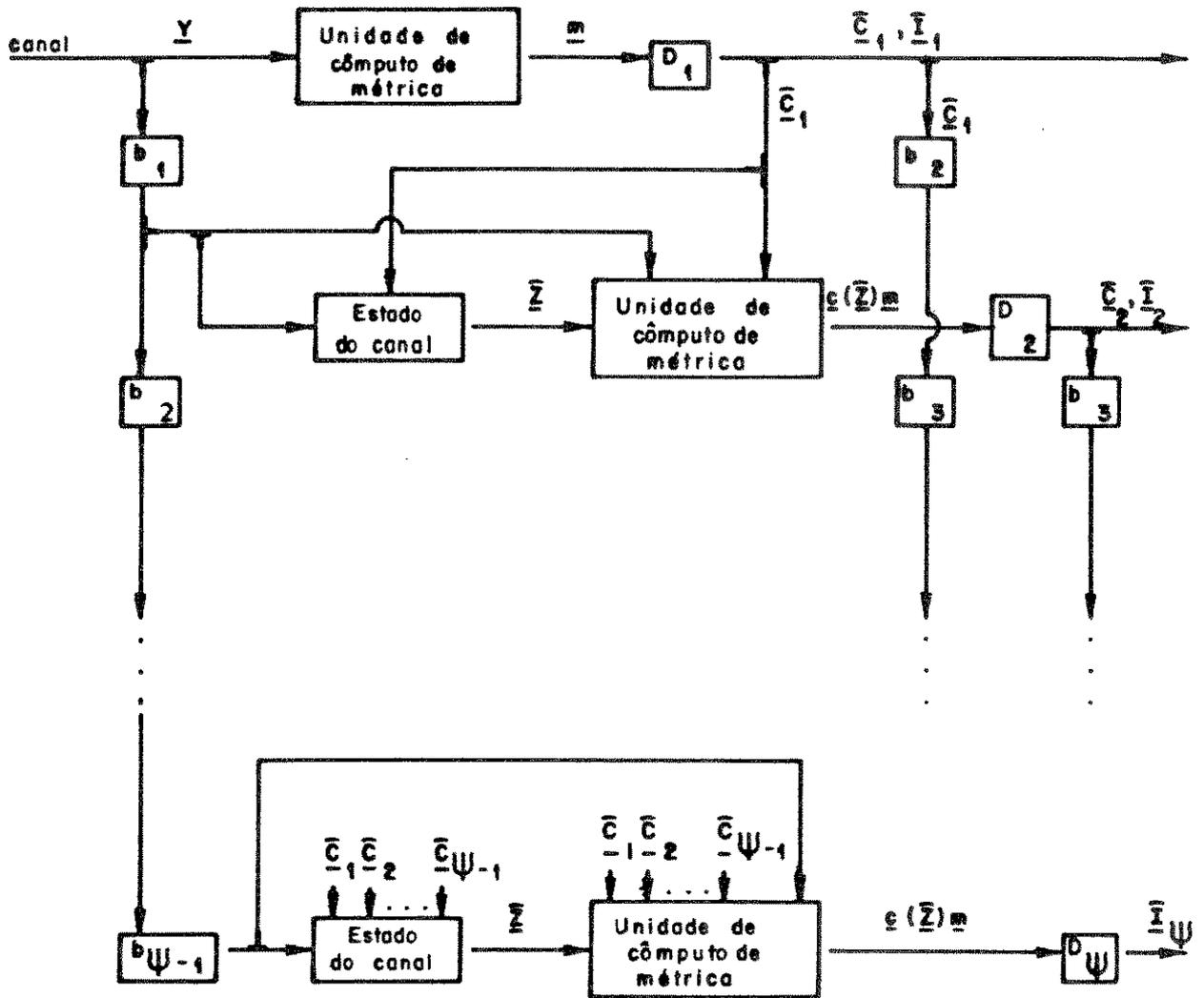


Figura 5.2 - Circuito de Decodificação Proposto.

será multiplicado pela métrica, que se torna $c(\hat{z})$.m. O custo $c(\hat{z})$ indica para a UCM quão confiável, ou quão relevante, deverá ser a métrica correspondente no processo de decodificação. Afora a integração do custo à métrica, o procedimento de decodificação é o mesmo do decodificador convencional.

Devemos notar que o estado do canal \hat{z} no i -ésimo estágio, é estimado a partir da informação lateral dos estágios anteriores \hat{C}_j , $j = 1, 2, \dots, i-1$, bem como do sinal recebido Y . Portanto, podemos escrever esta variável como sendo $\hat{z} = \hat{z}(\hat{C}_1, \hat{C}_2, \dots, \hat{C}_{i-1}, Y)$. Na seção seguinte apresentaremos alguns casos particulares, quando a análise de desempenho é considerada.

5.3. ANÁLISE DE DESEMPENHO DO DECODIFICADOR MULTISTÁGIO GENERALIZADO

A partir da Figura 5.2, identificamos que o decodificador convencional é um caso particular quando fazemos $c(0)=c(1)=1$. Nesta seção, estabeleceremos um procedimento geral para a análise de desempenho quando os custos $c(\hat{z})$ podem assumir qualquer valor no intervalo $[0,1]$. Neste intervalo, para cada estágio e para uma dada relação sinal-ruído, os custos são determinados de modo a minimizar a probabilidade de erro. Isto conduz ao decodificador multistádio ótimo, que, portanto, também é visto como um caso particular desta classe geral.

5.3.1. Procedimento Geral

O sistema que consideramos utiliza um código convolucional em cada

estágio. O algoritmo de Viterbi é selecionado para estimar as seqüências código $\{\hat{C}_i\}$, e a métrica $c(\hat{z})_m$ não quantizada permite uma decisão suave. A análise de desempenho segue da técnica de Função de Transferência como abordada no Capítulo 3. Seguindo o procedimento adotado por Wolf [4], uma treliça e correspondentemente uma Função de Transferência para códigos de bloco pode ser obtida, onde os estados (os vértices da treliça) são determinados pela combinação das colunas da matriz de paridade H . Isto torna a análise a ser desenvolvida extensível a esta classe de códigos.

O canal é assumido ser Gaussiano sem memória e os erros são aleatoriamente distribuídos. Entrelaçadores ideais podem ser utilizados com o propósito de garantir esta última suposição.

A probabilidade de erro de bit do sistema total, denotada por P_b , é fornecida por

$$P_b = \frac{1}{R} \sum_{i=1}^{\Psi} R_i P_{b_i}, \quad (5.1)$$

onde R é a eficiência espectral dada em (4.9) e P_{b_i} é a probabilidade de erro de bit do i -ésimo decodificador. Na nossa análise, devemos usar a probabilidade de erro de bit codificado como uma forma de medir a confiabilidade da informação lateral passada aos estágios subsequentes. As quantidades P_{b_i} e P_{c_i} são fornecidas a partir das expressões (3.94) e (3.95), respectivamente, que são reescritas aqui por conveniência:

$$P_{b_i} \leq \frac{1}{k_i} \left. \frac{\partial T_i(D^{(i)}, I)}{\partial I} \right|_{\substack{I=1 \\ D^{(i)}}}, \quad (5.2)$$

e

$$P_{c_i} \leq \frac{1}{n_i} D^{(i)} \left. \frac{\partial T_i(D^{(i)}, I)}{\partial D^{(i)}} \right|_{\substack{I=1 \\ D^{(i)}}}, \quad (5.3)$$

onde o índice i indica o i -ésimo estágio.

Estamos agora interessados em avaliar $D^{(i)}$ para cada estágio quando uma dada constelação no espaço Euclidiano é considerada. Para tanto, vamos reescrever a equação (3.14) numa forma simplificada e com os sinais definidos para o sistema em foco:

$$D(\lambda) = E \left[\exp \left(\lambda [m(\hat{x}, y; \hat{z}) - m(x, y; \hat{z})] \right) \mid x \right], \text{ para qualquer } \lambda \geq 0. \quad (5.4)$$

Na equação acima, o número de sinais vizinhos [5] devem ser computados. As definições que seguem estendem o conceito de vizinho para uma situação nova que surge em sistemas de Modulação Codificada Multinível.

Definição 5.3.1.1 : Seja $S = \{S_0, S_1, \dots, S_{M-1}\}$ um conjunto de M sinais no espaço Euclidiano, e considere um subconjunto $C \subset S$. Para um dado sinal transmitido $S_t \in C$, um sinal $S_j \in C$ é dito ser vizinho em C , se o seguinte é verdade:

$$d_E(S_t, S_j) = \min_{\substack{S_k \in C \\ S_k \neq S_t}} \left\{ d_E(S_t, S_k) \right\}. \quad (5.5)$$

A nova situação supramencionada aparece quando, se um erro de decodificação ocorre no i -ésimo nível da partição, ou seja, se a subconstelação é

erroneamente selecionada neste nível, então todas as subconstelações em qualquer nível $j > i$ que serão consideradas para cômputo de métricas *não contêm* o sinal transmitido. Para este caso se aplica a seguinte definição de vizinho.

Definição 5.3.1.2 : Seja $S = \{S_0, S_1, \dots, S_{M-1}\}$ um conjunto de M sinais no espaço Euclidiano, e considere um subconjunto $C \subset S$. Para um dado sinal transmitido $S_i \in (S - C)$, um sinal $S_j \in C$ é dito ser PW-vizinho (do inglês: Probability-wise-neighbor) de um sinal referência $S_i \in C$ em C , se o seguinte é verdade:

$$P(S_i \rightarrow S_j) = \max_{\substack{S_k \in C \\ S_k \neq S_i}} \left\{ P(S_i \rightarrow S_k) \right\} \Big| x_i \text{ é transmitido} \quad (5.6)$$

A questão relevante é que o sinal transmitido *não pertence* ao subconjunto considerado C . Para incluir ambas as situações acima em (5.4), devemos introduzir a *métrica generalizada* a qual se aplica ao ruído Gaussiano com média nula e variância σ^2 por dimensão e é definida como

$$m(x, x', y; \hat{z}) = - \frac{c(\hat{z}) \cdot d_E^2(y, x')}{\alpha \cdot \sigma^2} \Big| x \text{ é transmitido} \quad (5.7)$$

onde: x é o sinal transmitido; y é o sinal recebido; x' é um sinal referência; \hat{z} é o estado do canal estimado e a constante $1/\alpha\sigma^2$ é adotada por conveniência com $\alpha > 0$.

Vamos agora considerar o espaço \mathbb{R}^2 com os sinais definidos acima, como indicado na Figura 5.3. A saída do canal perturbada pelo ruído aditivo no instante t é

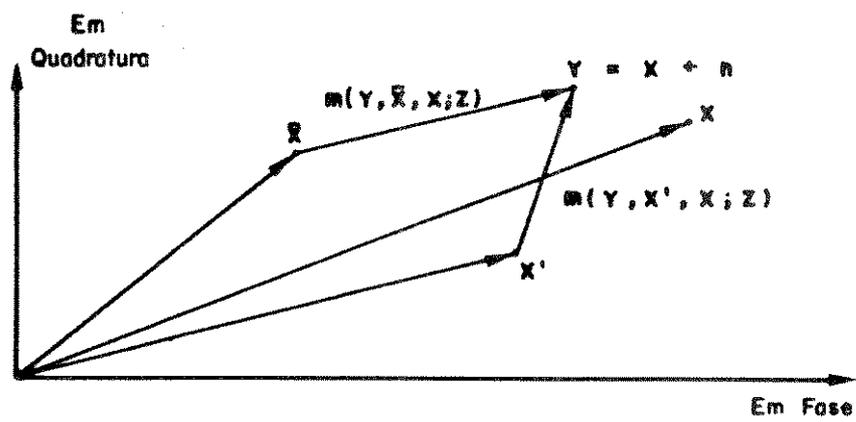


Figura 5.3 - Espaço Euclidiano \mathbb{R}^2 .

$$y(t) = x(t) + n(t), \quad (5.8)$$

onde $n(t)$ é um ruído faixa-estreita [6] (do Inglês: narrow-band noise) com média zero e variância σ^2 . Expandindo (5.8), esta se torna

$$y(t) = \left[x_p + n_p(t) \right] \cos(w_c t) - \left[x_q + n_q(t) \right] \sin(w_c t), \quad (5.9)$$

onde os índices "p" e "q" denotam as componentes em fase e em quadratura, respectivamente. Os sinais e o ruído ocupam a mesma faixa cuja frequência central é $w_c/2\pi$ (Hz). As componentes em fase e em quadratura do sinal $y(t)$, são respectivamente:

$$y_p = x_p + n_p \quad (5.10)$$

e

$$y_q = x_q + n_q, \quad (5.11)$$

onde o tempo t foi suprimido por questão de conveniência. Devemos notar que y_p e y_q representam variáveis aleatórias gaussianas, independentes e com médias x_p e x_q , respectivamente. Substituindo a métrica generalizada dada em (5.7) na equação (5.4), teremos

$$\begin{aligned} D(\lambda) &\leq E \left[\exp \left\{ \lambda c(\hat{z}) \left[d^2(y, x') - d^2(y, \hat{x}) \right] / \alpha \cdot \sigma^2 \right\} \middle| \hat{x} \right]_{\hat{x} \neq x'} \\ &= E \left[\exp \left\{ \lambda c(\hat{z}) \left[(y_p - x'_p)^2 + (y_q - x'_q)^2 - (y_p - \hat{x}_p)^2 - (y_q - \hat{x}_q)^2 \right] / \alpha \cdot \sigma^2 \right\} \middle| \hat{x} \right]_{\hat{x} \neq x'} \\ &= E \left[\exp \left\{ \lambda c(\hat{z}) \left[2 \left(y_p (\hat{x}_p - x'_p) + y_q (\hat{x}_q - x'_q) \right) + (x'^2_p - \hat{x}^2_p) + (x'^2_q - \hat{x}^2_q) \right] / \alpha \cdot \sigma^2 \right\} \middle| \hat{x} \right]_{\hat{x} \neq x'} \end{aligned} \quad (5.12)$$

Vamos assumir, por questão de brevidade, que $c(\hat{z})$ é uma quantidade determinística, digamos ξ . Então deveremos ter

$$D(\lambda) = \exp\left\{\lambda\xi\left[(x_p'^2 - \hat{x}_p^2) + (x_q'^2 - \hat{x}_q^2)\right]/\alpha.\sigma^2\right\} \cdot E\left[\exp\left\{\lambda\xi\left[2\left(y_p(\hat{x}_p - x_p') + y_q(\hat{x}_q - x_q')\right)\right]/\alpha.\sigma^2\right\}\right]_{\substack{\hat{x} \\ x \neq x'}} \quad (5.13)$$

Utilizando a função característica de uma variável aleatória gaussiana v com média μ e variância σ^2 [7]:

$$C(jw) = E[\exp\{jwv\}] = \exp\left\{jw\mu - \frac{w^2\sigma^2}{2}\right\},$$

ou substituindo $\varphi = jw$

$$C(\varphi) = E[\exp\{\varphi v\}] = \exp\left\{\varphi\mu + \frac{\varphi^2\sigma^2}{2}\right\}, \quad (5.14)$$

na equação (5.13) esta se torna

$$D(\lambda) = \exp\left\{\frac{4\alpha\lambda\xi\left[x_p(\hat{x}_p - x_p') + x_q(\hat{x}_q - x_q')\right] + 4\lambda^2\xi^2\left[(\hat{x}_p - x_p')^2 + (\hat{x}_q - x_q')^2\right]}{2\alpha^2.\sigma^2}\right\} \cdot \exp\left\{\frac{2\alpha\lambda\xi\left[(x_p'^2 - \hat{x}_p^2) + (x_q'^2 - \hat{x}_q^2)\right]}{2\alpha^2.\sigma^2}\right\}. \quad (5.15)$$

A expressão acima pode ser usada no caso da informação lateral corre-

ta bem como errada, com o correspondente número de vizinhos ou PW-vizinhos multiplicando a exponencial.

5.3.2. O Caso da Constelação 8-PSK

Vamos considerar a constelação 8-PSK, a qual é particionada segundo Ungerboeck e mostrada na Figura B.1 (veja Apêndice B). Em conformidade com o Capítulo 4, a eficiência espectral R para este sistema deve ser de 2 bits/sinal.

Vamos assumir, sem perda de generalidade para o caso em análise, que o sinal S_0 foi transmitido.

1º estágio: O canal visto pelo decodificador D_1 é o canal Gaussiano. Substituindo os valores (ver Figura B.1)

$$x_p = x'_p = 1; \quad x_q = x'_q = 0; \quad \hat{x}_p = \cos(2\pi/8); \quad \hat{x}_q = \sin(2\pi/8); \quad \xi = 1$$

em (5.15) e minimizando esta com relação a λ devemos ter

$$D^{(1)} \leq 2 \exp \left\{ - \frac{0.586}{8\sigma^2} \right\}. \quad (5.16)$$

Portanto, as probabilidades de erro de bit e bit codificado do primeiro estágio são obtidas substituindo (5.16) em (5.2) e (5.3), respectivamente.

2º estágio: Neste caso, os possíveis erros do primeiro estágio devem ser leva-

dos em consideração. Eles ocorrem com probabilidade $P(z=1) = P_{c_1}$.

Substituindo os valores (ver subconjuntos $C_1 = 0$ e $C_1 = 1$ na Figura B.1)

$$\left\{ \begin{array}{l} x_p = x'_p = 1 \\ x_q = x'_q = 0 \\ \hat{x}_p = 0, \hat{x}_q = 1 \\ \xi = c(0) \end{array} \right. \quad e \quad \left\{ \begin{array}{l} x_p = 1, x_q = 0 \\ x'_p = \cos(\pi/4), x'_q = \sin(\pi/4) \\ \hat{x}_p = \cos(-\pi/4), \hat{x}_q = \sin(-\pi/4) \\ \xi = c(1) \end{array} \right.$$

para $z=0$ para $z=1$

em (5.15) e minimizando esta com relação a λ devemos ter

$$D^{(2)} \leq 2 \exp\left\{ \frac{-2c(0) + c(0)^2}{4\sigma^2} \right\} + P_{c_1} \exp\left\{ \frac{c(1)^2}{4\sigma^2} \right\}. \quad (5.17)$$

Na expressão acima, está implicado que o estimador do estado do canal sempre acerta. Aqui devemos decidir sobre o estado do canal com um simples *teste de razão de máxima verossimilhança* [7], [8]. No caso em análise, os sinais S_0, S_2, S_4 e S_6 são considerados como a hipótese "0" e os sinais restantes como a hipótese "1". Isto é mostrado na Tabela 5.1.

A partir da Figura B.1 (Apêndice B) e da Tabela 5.1, a probabilidade de ocorrer um erro na estimação do estado do canal, denotada por P_{cs_2} , é dada por:

$$P_{cs_2} = P\left[y \in \left(\Lambda_1 U \Lambda_3 U \Lambda_5 U \Lambda_7 \right) \mid S_0 \right] = \int_{\Lambda_1 U \Lambda_3 U \Lambda_5 U \Lambda_7} p_{\Theta}(\Theta) d\Theta \quad (5.18)$$

onde [6], [8]:

Tabela 5.1 - Determinação de $\hat{z} = \hat{z}(\hat{C}_1, y)$

REGIÕES CONTENDO O SINAL Y	\hat{C}_1	\hat{z}
$\Lambda_0, \Lambda_2, \Lambda_4, \Lambda_6$	0	0
$\Lambda_0, \Lambda_2, \Lambda_4, \Lambda_6$	1	1
$\Lambda_1, \Lambda_3, \Lambda_5, \Lambda_7$	0	1
$\Lambda_1, \Lambda_3, \Lambda_5, \Lambda_7$	1	0

$$P_{\Theta}(\Theta) = \frac{1}{2\pi} \exp\left\{-\frac{1}{2\sigma^2}\right\} + \frac{1}{[2\pi\sigma^2]^{1/2}} \cos(\Theta) \exp\left\{-\frac{\sin^2(\Theta)}{2\sigma^2}\right\} \cdot \left[1 - \frac{1}{2} \operatorname{erfc}\left(\frac{\cos(\Theta)}{2\sigma^2}\right)\right] \quad (5.19)$$

é a função densidade de probabilidade da fase do sinal y , quando S_0 é transmitido. A função erro complementar $\operatorname{erfc}(\cdot)$ acima é definida como [6]:

$$\operatorname{erfc}(u) = \frac{2}{\sqrt{\pi}} \int_u^{\infty} \exp\{-v^2\} dv.$$

Quando \hat{z} é estimado erroneamente os custos $c(0)$ e $c(1)$ são permutados. Então, com um ligeiro abuso de notação (5.17) se torna

$$D^{(2)} \leq \left[2 \exp\left\{\frac{-2c(0) + c(0)^2}{4\sigma^2}\right\} + P_{c_1} \exp\left\{\frac{c(1)^2}{4\sigma^2}\right\}\right] (1 - P_{cs_2}) + \left[2 \exp\left\{\frac{-2c(1) + c(1)^2}{4\sigma^2}\right\} + P_{c_1} \exp\left\{\frac{c(0)^2}{4\sigma^2}\right\}\right] P_{cs_2} \quad (5.20)$$

As probabilidades de erro de bit e bit codificado para este estágio são então fornecidas substituindo (5.20) em (5.2) e (5.3), respectivamente.

3º estágio: Agora erros de ambos os estágios anteriores devem ser considerados.

A probabilidade de erro é $P(z=1) = P(z_1=1, z_2=0) + P(z_1=0, z_2=1) + P(z_1=1, z_2=1) = P(z_1=1 \text{ ou } z_2=1)$. Este valor pode ser limitado superiormente pela "Union Bound" $P(z=1) \leq P_{c_1} + P_{c_2}$. Devemos considerar

a subconstelação mais provável para o caso em que a informação lateral está errada, ou seja, $C_1 C_2 = 01$ (ver Figura B.1). Substituindo os valores:

$$\left\{ \begin{array}{l} x_p = x'_p = 1 \\ x_q = x'_q = 0 \\ \hat{x}_p = -1, \hat{x}_q = 0 \\ \xi = c(0) \end{array} \right. \quad e \quad \left\{ \begin{array}{l} x_p = 1, x'_p = 0 \\ x_q = 0, x'_q = 1 \\ \hat{x}_p = 0, \hat{x}_q = -1 \\ \xi = c(1) \end{array} \right.$$

para $z=0$ para $z=1$

em (5.15) e minimizando esta com relação a λ devemos ter

$$D^{(3)} \leq \exp\left\{\frac{-c(0) + 0.5c(0)^2}{\sigma^2}\right\} + \left(P_{c_1} + P_{c_2}\right) \exp\left\{\frac{c(1)^2}{2\sigma^2}\right\}. \quad (5.21)$$

Neste caso, o estado do canal é fornecido pela Tabela 5.2 e a probabilidade de ocorrer um erro na estimação do estado do canal, denotada por P_{cs_3} , é dada por:

$$P_{cs_3} \leq P\left[y \in \left(\Lambda_1 \cup \Lambda_2 \cup \Lambda_3 \cup \Lambda_5 \cup \Lambda_6 \cup \Lambda_7\right)\right] = \int_{\Lambda_1 \cup \Lambda_2 \cup \Lambda_3 \cup \Lambda_5 \cup \Lambda_6 \cup \Lambda_7} p_{\Theta}(\Theta) d\Theta. \quad (5.22)$$

A desigualdade acima segue do fato de que os eventos de número 7,8, 10,12,14 e 15, na Tabela 5.2, não representam um erro de estimação se S_0 é o sinal transmitido. Por exemplo, no 7º evento da Tabela 5.2, $\hat{z} = 1$ é uma estimação correta desde que os valores $\hat{C}_1 \hat{C}_2 = 10$ são diferentes dos valores verdadeiros $C_1 C_2 = 00$. Entretanto a integral em (5.22) é incondicionalmente sobre as regiões Λ_2 e Λ_6 .

Então (5.21) se torna

$$D^{(3)} \leq \left[\exp\left\{\frac{-c(0) + 0.5c(0)^2}{\sigma^2}\right\} + \left(P_{c_1} + P_{c_2}\right) \exp\left\{\frac{c(1)^2}{2\sigma^2}\right\} \right] \left(1 - P_{cs_3}\right) + \left[\exp\left\{\frac{-c(1) + 0.5c(1)^2}{\sigma^2}\right\} + \left(P_{c_1} + P_{c_2}\right) \exp\left\{\frac{c(0)^2}{2\sigma^2}\right\} \right] P_{cs_3} \quad (5.23)$$

Tabela 5.2 - Determinação de $\hat{z} = \hat{z}(\hat{C}_1, \hat{C}_2, y)$

EVENTO	REGIÕES CONTENDO O SINAL Y	$\hat{C}_1 \hat{C}_2$	\hat{z}
1	Λ_0, Λ_4	0 0	0
2	Λ_0, Λ_4	0 1	1
3	Λ_0, Λ_4	1 0	1
4	Λ_0, Λ_4	1 1	1
5	Λ_2, Λ_6	0 0	1
6	Λ_2, Λ_6	0 1	0
7	Λ_2, Λ_6	1 0	1
8	Λ_2, Λ_6	1 1	1
9	Λ_1, Λ_5	0 0	1
10	Λ_1, Λ_5	0 1	1
11	Λ_1, Λ_5	1 0	0
12	Λ_1, Λ_5	1 1	1
13	Λ_3, Λ_7	0 0	1
14	Λ_3, Λ_7	0 1	1
15	Λ_3, Λ_7	1 0	1
16	Λ_3, Λ_7	1 1	0

Como nos estágios anteriores, as probabilidades de erro de bit e bit codificado podem ser obtidas substituindo (5.23) em (5.2) e (5.3), respectivamente.

A partir dos valores de $D^{(i)}$ para cada estágio, fornecidos por (5.16), (5.20) e (5.23), devemos otimizar (5.20) e (5.23) com relação aos custos $c(\hat{z})$, o que conduz ao decodificador ótimo. Agora se os custos são escolhidos para serem $c(0)=c(1)=1$, o limitante de desempenho é particularizado para o caso do decodificador convencional.

5.3.3. O Caso da Constelação 2^{τ} -PSK (Generalização)

A distribuição circular dos sinais na constelação M-PSK, quando M é uma potência de 2, apresenta uma simetria que torna possível uma extensão do caso da seção 5.3.2. Neste caso, os valores de $D^{(i)}$ para cada estágio são fornecidos da forma que segue:

1º estágio: Teremos

$$D^{(1)} \leq 2 \exp \left\{ \frac{1 - \cos \left(\frac{2\pi}{M} \right)}{4\sigma^2} \right\} . \quad (5.24)$$

i-ésimo estágio: Aqui devemos substituir os valores

$$\left\{ \begin{array}{l} x_p = x'_p = 1 \\ x_q = x'_q = 0 \\ \xi = c(0) \\ \hat{x}_p = \cos\left(\frac{2^i \pi}{M}\right); \\ \hat{x}_q = \sin\left(\frac{2^i \pi}{M}\right), \text{ para } z=0 \end{array} \right. \quad e \quad \left\{ \begin{array}{l} x_p = 1 \\ x_q = 0 \\ x'_p = \cos\left(\frac{2^{i-1} \pi}{M}\right) \\ x'_q = \sin\left(\frac{2^{i-1} \pi}{M}\right) \\ \xi = c(1) \\ \hat{x}_p = \cos\left(\frac{2^{i-1} \pi}{M}\right) \\ \hat{x}_q = -\sin\left(\frac{2^{i-1} \pi}{M}\right), \text{ para } z=1, \end{array} \right.$$

em (5.15) e minimizar esta com relação a λ . Assim, para o i -ésimo estágio, devemos ter

$$\begin{aligned} D^{(i)} \leq & \left[2 \exp \left\{ \frac{\left[2c(0) - c(0)^2 \right] \left[\cos\left(\frac{2^i \pi}{M}\right) - 1 \right]}{4\sigma^2} \right\} + \left(\sum_{j=1}^{i-1} P_{c_j} \right) \right] \\ & \cdot \exp \left\{ \frac{c(1)^2 \sin^2\left(\frac{2^{i-1} \pi}{M}\right)}{2\sigma^2} \right\} \left(1 - P_{c_i} \right) + \\ & + \left[2 \exp \left\{ \frac{\left[2c(1) - c(1)^2 \right] \left[\cos\left(\frac{2^i \pi}{M}\right) - 1 \right]}{4\sigma^2} \right\} + \left(\sum_{j=1}^{i-1} P_{c_j} \right) \right] \\ & \cdot \exp \left\{ \frac{c(0)^2 \sin^2\left(\frac{2^{i-1} \pi}{M}\right)}{2\sigma^2} \right\} P_{c_i}; \quad i=2,3,\dots,(\log_2 M)-1, \end{aligned} \quad (5.25)$$

e

$$\begin{aligned}
 D^{(i)} \leq & \left[\exp \left\{ \frac{\left[2c(0) - c(0)^2 \right] \left[\cos \left(\frac{2^i \pi}{M} \right) - 1 \right]}{4\sigma^2} \right\} + \left(\sum_{j=1}^{i-1} P_{c_j} \right) \right] \\
 & \cdot \exp \left\{ \frac{c(1)^2 \sin^2 \left(\frac{2^{i-1} \pi}{M} \right)}{2\sigma^2} \right\} \cdot \left(1 - P_{cs_i} \right) + \\
 & + \left[\exp \left\{ \frac{\left[2c(1) - c(1)^2 \right] \left[\cos \left(\frac{2^i \pi}{M} \right) - 1 \right]}{4\sigma^2} \right\} + \left(\sum_{j=1}^{i-1} P_{c_j} \right) \right] \\
 & \cdot \exp \left\{ \frac{c(0)^2 \sin^2 \left(\frac{2^{i-1} \pi}{M} \right)}{2\sigma^2} \right\} P_{cs_i} ; \quad i = \log_2 M \quad (5.26)
 \end{aligned}$$

onde

$$P_{cs_i} \leq 2 \sum_{j=1}^{M/2^i} \int_{\frac{(4j-3)\pi}{M \cdot 2^{(2-i)}}}^{\frac{(4j-1)\pi}{M \cdot 2^{(2-i)}}} p_{\Theta}(\Theta) d\Theta ; \quad i=2,3,\dots,\log_2 M. \quad (5.27)$$

5.3.4. O Caso da Constelação 4^{τ} -QAM (Generalização)

Para uma constelação retangular contendo $M = 4^{\tau}$ (τ um inteiro positivo) sinais, uma cadeia de partições binária de 2τ níveis é realizada segundo

Ungerboeck. Em conformidade com o Capítulo 4, a eficiência espectral R para este sistema deve ser de $(2\tau - 1)$ bits/sinal.

Na Figura 5.4 uma representação construtiva para a constelação retangular é mostrada. As letras A, B, C e D representam pontos de sinais no espaço Euclidiano bi-dimensional. Definindo S_A como sendo o conjunto de todos os sinais com rótulo A, S_{AB} como sendo o conjunto de todos os sinais com rótulos A ou B, e assim por diante, o conjunto S_{AB} representa a constelação em análise. Começando com o caso mais simples, a saber $\tau = 1$ (4-QAM), podemos construir a constelação 16-QAM repetindo a estrutura anterior três vezes como indicado na Figura 5.4. Repetindo o mesmo procedimento mais uma vez a constelação 64-QAM é gerada, e assim por diante.

Deve ser notado que as constelações S_C , S_{AB} e S_{ABCD} são subconjuntos do reticulado Z^2 (o conjunto de todos os pontos $(x, y) : x, y \in Z$), e que as constelações S_A , S_B , S_D e S_{ABC} são subconjuntos do reticulado D^2 (uma versão do reticulado Z^2 rotacionada de 45° e expandida por um fator multiplicativo de $2^{1/2}$).

No caso das constelações retangulares, o número de vizinhos e PW-vizinhos merecem uma atenção especial. Os teoremas que seguem dizem respeito a estas quantidades, que serão apresentadas primeiro que a derivação de $D^{(i)}$. Vamos assumir que todos os sinais da constelação são equi-prováveis.

Teorema 5.3.4.1 : Na definição 5.3.1.1 se S é o conjunto S_{ABCD} e C o subconjunto S_{AB} , então o número médio de vizinhos em C sobre todos os possíveis sinais transmitidos $S_i \in C$ é dado por:

$$N_1(\tau) = 4(1 - 2^{-\tau}), \quad (5.28)$$

onde 4^τ é o número de sinais em S_{AB} .

$\tau = 1$				$\tau = 2$				$\tau = 3$											
A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D
D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C
B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D
D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C
A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D
D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C
B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D
D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C
A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D
D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C
B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D	B	D	A	D
D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C

Figura 5.4 - Representação Construtiva para os Conjuntos Retangulares, S_{AB} é a Constelação 4^{τ} -QAM.

Prova : Na constelação retangular o número de vizinhos depende da posição em que o sinal transmitido se encontra. Se ele é um dos n_I sinais no interior da constelação, ele tem quatro vizinhos, enquanto que os n_L sinais das laterais e os n_V sinais dos vértices têm, cada um deles, três e dois vizinhos, respectivamente. Na constelação S_{AB} estes números são $n_I = 2^{2\tau} - 2^{\tau+2} + 4$, $n_L = 4(2^\tau - 2)$ e $n_V = 4$. Como todos os sinais são equiprováveis, o número médio de vizinhos é dado por:

$$N_I(\tau) = \frac{4n_I + 3n_L + 2n_V}{n_I + n_L + n_V}.$$

Substituindo os valores de n_I , n_L e n_V em $N_I(\tau)$, após algumas manipulações algébricas chegamos a:

$$N_I(\tau) = 4(1 - 2^{-\tau}).$$

■

Teorema 5.3.4.2 : Na definição 5.3.1.1 se S é o conjunto S_{ABCD} e C o subconjunto S_A , então o número médio de vizinhos em C sobre todos os possíveis sinais transmitidos $S_i \in C$ é dado por:

$$N_2(\tau) = 4(1 - 2^{-2\tau} - 2^{1-\tau}), \quad (5.29)$$

onde $4^{\tau/2}$ é o número de sinais em S_A .

Prova : Como no teorema 5.3.4.1, se o sinal transmitido é um dos n_I sinais no interior da constelação, ele tem quatro vizinhos, enquanto que os n_L das laterais e os n_V dos vértices têm, cada um deles, dois e um vizinho, respectivamente. Na constelação S_A estes números são $n_I = 2^{2\tau-1} - 2 - 4(2^{\tau-1} - 1)$, $n_L = 4(2^{\tau-1} - 1)$ e $n_V = 2$. Como todos os sinais são equiprováveis, o número médio de vizinhos é dado por:

$$N_2(\tau) = \frac{4n_I + 2n_L + n_V}{n_I + n_L + n_V}.$$

Substituindo os valores de n_I , n_L e n_V em $N_2(\tau)$ teremos

$$N_2(\tau) = 4(1 - 2^{-2\tau} - 2^{1-\tau}).$$

Teorema 5.3.4.3 : Na definição 5.3.1.2, se S é o conjunto S_{ABC} e C o subconjunto S_C , então o número médio de PW-vizinhos em C sobre todos os possíveis sinais transmitidos $S_t \in S_{AB}$ (o sinal referência é sempre escolhido como sendo o sinal mais próximo de S_t) é dado por:

$$P_1(\tau) = 2(1 - 2^{-2\tau} - 2^{-\tau}), \quad (5.30)$$

onde 4^τ é o número de sinais em S_{AB} .

Prova : Se o sinal transmitido $S_t \in S_{AB}$ é um dos n_{VA} sinais dos vértices que pertencem a S_A , o sinal referência tem dois PW-vizinhos, enquanto que se ele é um dos n_{VB} sinais dos vértices que pertencem a S_B , existe apenas um PW-vizinho. Se o sinal transmitido é um dos n_{L1} sinais da lateral superior ou esquerda, o sinal referência tem apenas um PW-vizinho, enquanto se ele é um dos n_{E2} sinais da lateral inferior ou direita existem dois PW-vizinhos. Finalmente, se ele é um dos n_I pontos no interior da constelação, o sinal referência tem dois PW-vizinhos. Na constelação S_{AB} estes números são: $n_{VA} = n_{VB} = 2$, $n_{L1} = n_{L2} = 2(2^\tau - 2)$ e $n_I = 2^{2\tau} - 2^{\tau+2} + 4$. Como todos os sinais são equiprováveis, o número médio de PW-vizinhos é dado por:

$$P_1(\tau) = \frac{2n_{VA} + n_{VB} + n_{L1} + 2n_{L2} + 2n_I}{n_{VA} + n_{VB} + n_{L1} + n_{L2} + n_I}$$

Substituindo os valores de n_I , n_{VA} , n_{VB} , n_{L1} e n_{L2} em $P_1(\tau)$ resulta em:

$$P_1(\tau) = 2(1 + 2^{-2\tau} - 2^{-\tau}).$$

A partir do procedimento utilizado nas provas dos teoremas acima, fica claro que não nos preocupamos se o sinal transmitido (para os dois primeiros teoremas) ou o sinal referência (para o terceiro teorema) são rotulados com "0" ou "1", ver Figura 5.5. Adicionalmente, os vizinhos e PW-vizinhos são sempre rotulados com o valor oposto do sinal considerado, uma vez que na estrutura de partição de Ungerboeck as constelações são naturalmente rotuladas segundo Gray, como visto na Figura 5.5. Na prova do teorema que segue, devemos considerar os sinais rotulados só com "0", ou só com "1", mas não com ambos. Devido à simetria, qualquer uma das opções conduz ao mesmo resultado.

Teorema 5.3.4.4 : Na definição 5.3.1.2, se S é o conjunto S_{AB} e C o subconjunto S_B , então o número médio de PW-vizinhos em C sobre todos os possíveis sinais transmitidos $S_t \in S_A$ é dado por:

$$P_2(\tau) = 2(1 - 2^{-\tau}), \quad (5.31)$$

onde $\frac{4^\tau}{2}$ é o número de sinais em S_A .

Prova : Como no teorema 5.3.4.3, se o sinal transmitido $S_t \in S_A$ é um dos n_V sinais dos vértices, então o sinal referência tem apenas um PW-vizinho, enquanto se ele é um dos n_I sinais no interior da constelação, existem dois PW-vizinhos. Se ele é um dos n_{L1} sinais da lateral inferior ou esquerda, ele

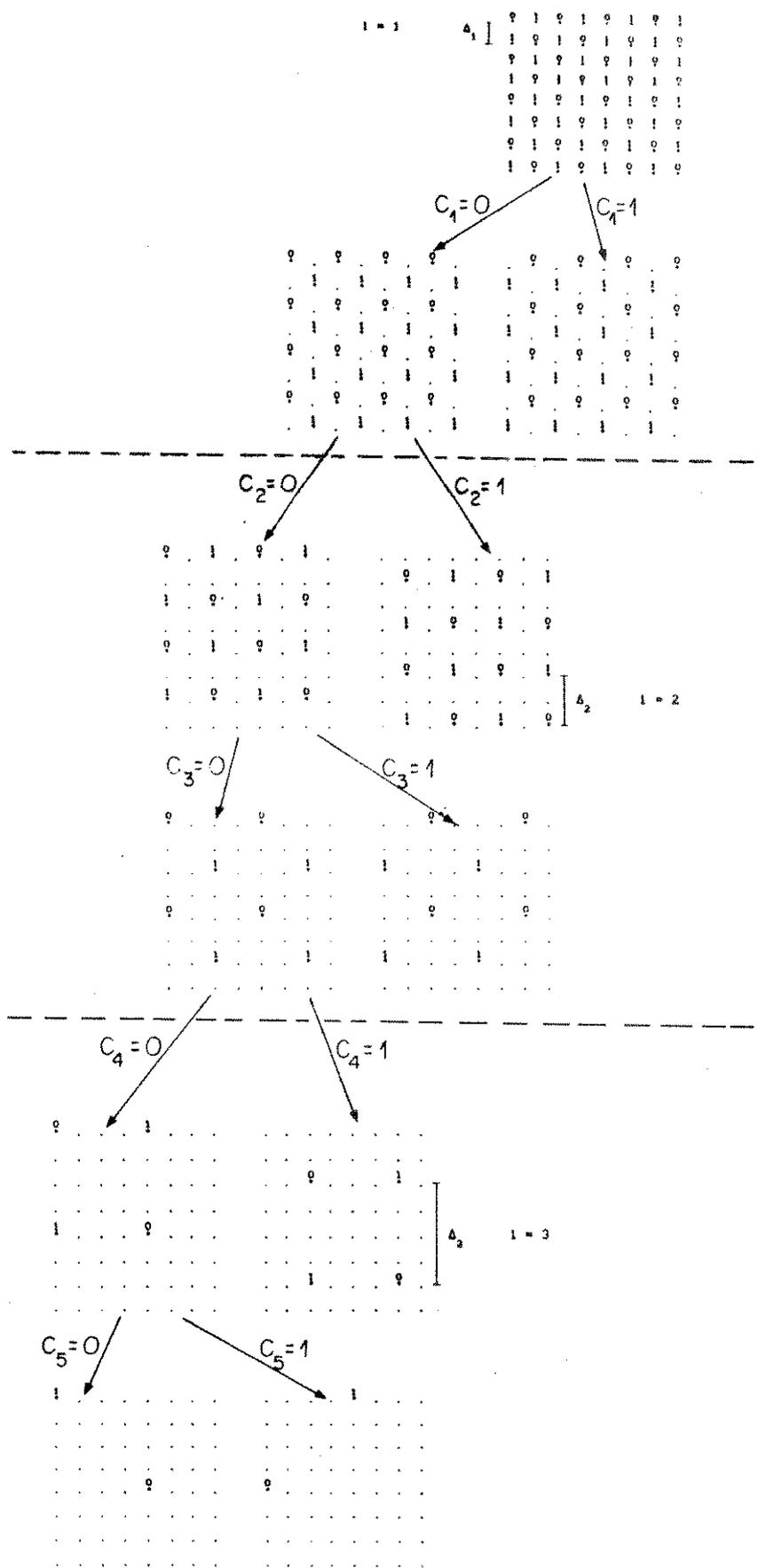


Figura 5.5 - Estruturas Repetitivas na Constelação 64-QAM.

tem dois PW-vizinhos quando o sinal referência rotulado com "0" é considerado ou apenas um PW-vizinho quando o sinal referência é rotulado com "1". Quando o sinal transmitido é um dos n_{L2} sinais da lateral superior ou direita, estes valores são permutados. Portanto, a escolha por este ou aquele rótulo é irrelevante. Na constelação S_A estes números são: $n_V = 2$, $n_I = 2^{2^{\tau-1}} - 2 - 4(2^{\tau-1} - 1)$ e $n_{L1} = n_{L2} = 2(2^{\tau-1} - 1)$. Como todos os sinais são equiprováveis, o número médio de PW-vizinhos é dado por:

$$P_2(\tau) = \frac{n_V + 2n_I + 2n_{L1} + n_{L2}}{n_V + n_I + n_{L1} + n_{L2}}.$$

Substituindo os valores de n_I , n_V , n_{L1} e n_{L2} na equação acima resulta

$$P(\tau) = 2(1 - 2^{-\tau}).$$

Na avaliação de $D^{(1)}$ para as constelações retangulares, devemos explorar as estruturas repetivas que surgem na árvore de partição, como salientado na Figura 5.5. Vamos assumir que o sinal transmitido* é aquele que corresponde a $(C_1, C_2, \dots, C_{2\tau}) = (0, 0, \dots, 0)$ e vamos considerar apenas a parte mais provável da árvore sob esta condição. Para a constelação 64-QAM ($\tau=3$) da Figura 5.5, percebemos que as três estruturas são semelhantes, a menos de um fator de escala de 2 e do número de sinais. É plausível admitir que τ destas estruturas devam existir na constelação 4^τ -QAM. É suficiente, portanto, considerar apenas uma delas, por exemplo aquela para $i=2$ na Figura 5.5. Usando o fato de que a probabilidade de erro independe da origem do sistema, vamos, por uma questão de simplicidade, escolher esta para ser o ponto $\left(\frac{\Delta_1}{2}, -\frac{\Delta_1}{2} \right)$. Substituindo os valores:

* Este sinal está no interior da constelação, que é o pior caso para a probabilidade de erro.

$$\begin{cases}
 x_p = x'_p = 0 \\
 x_q = x'_q = 0 \\
 \hat{x}_p = \Delta_2 ; \quad \hat{x}_q = 0 \\
 \xi = c(0)
 \end{cases}
 \quad e \quad
 \begin{cases}
 x_p = 0 ; \quad x'_p = \frac{\Delta_2}{2} \\
 x_q = 0 ; \quad x'_q = \frac{\Delta_2}{2} \\
 \hat{x}_p = \frac{\Delta_2}{2} ; \quad \hat{x}_q = \frac{\Delta_2}{2} \\
 \xi = c(1)
 \end{cases}$$

para $z=0$ para $z=1$

na equação (5.15) e minimizando esta com relação a λ , o valor de D para o primeiro estágio da estrutura analisada é:

$$D^{(3)} \leq N_1(2) \exp \left\{ \frac{c(0)^2 \Delta_2^2 - 2c(0)\Delta_2^2}{8\sigma^2} \right\} + (P_{c_1} + P_{c_2}) P_1(2) \exp \left\{ \frac{c(1)^2 \Delta_2^2}{8\sigma^2} \right\}. \quad (5.32)$$

Agora substituindo os valores:

$$\begin{cases}
 x_p = x'_p = 0 \\
 x_q = x'_q = 0 \\
 \hat{x}_p = \Delta_2 ; \quad \hat{x}_q = \Delta_2 \\
 \xi = c(0)
 \end{cases}
 \quad e \quad
 \begin{cases}
 x_p = 0 ; \quad x'_p = \Delta_2 \\
 x_q = 0 ; \quad x'_q = 0 \\
 \hat{x}_p = 0 ; \quad \hat{x}_q = \Delta_2 \\
 \xi = c(1)
 \end{cases}$$

para $z=0$ para $z=1$

na equação (5.15) e minimizando esta com relação a λ , o valor de D para o segundo estágio da estrutura analisada é:

$$D^{(4)} \leq N_2(2) \exp \left\{ \frac{c(0)^2 \Delta_2^2 - 2c(0)\Delta_2^2}{4\sigma^2} \right\} + (P_{c_1} + P_{c_2} + P_{c_3}) P_2(2) \exp \left\{ \frac{c(1)^2 \Delta_2^2}{4\sigma^2} \right\}. \quad (5.33)$$

Notando que

$$\Delta_{i+1} = 2\Delta_i \quad ; \quad i=1, \dots, \tau$$

ou

$$\Delta_i = 2^{i-1}\Delta_1 \quad ; \quad i=1, \dots, \tau \quad (5.34)$$

e usando a equivalência entre as estruturas mencionadas e os teoremas acima, podemos facilmente estender este caso para uma constelação 4^τ -QAM, como mostrado abaixo:

$$D^{(1)} \leq N_1(\tau) \exp\left\{\frac{-\Delta_1^2}{8\sigma^2}\right\}, \quad (5.35)$$

$$D^{(2i-1)} \leq N_1(\tau - i + 1) \exp\left\{\frac{c(0)^2(2^{i-1}\Delta_1)^2 - 2c(0)(2^{i-1}\Delta_1)^2}{8\sigma^2}\right\} + \left(\sum_{j=1}^{2i-2} P_{c_j}\right) P_1(\tau - i + 1) \exp\left\{\frac{c(1)^2(2^{i-1}\Delta_1)^2}{8\sigma^2}\right\}; \quad i=2, \dots, \tau, \quad (5.36)$$

e

$$D^{(2i)} \leq N_2(\tau - i + 1) \exp\left\{\frac{c(0)^2(2^{i-1}\Delta_1)^2 - 2c(0)(2^{i-1}\Delta_1)^2}{4\sigma^2}\right\} + \left(\sum_{j=1}^{2i-1} P_{c_j}\right) P_2(\tau - i + 1) \exp\left\{\frac{c(1)^2(2^{i-1}\Delta_1)^2}{4\sigma^2}\right\}; \quad i=1, 2, \dots, \tau, \quad (5.37)$$

onde $N_1(\cdot)$, $N_2(\cdot)$, $P_1(\cdot)$ e $P_2(\cdot)$ são fornecidos pelos teoremas 5.3.4.1, 5.3.4.2, 5.3.4.3 e 5.3.4.4.

Vamos agora considerar a estimação do estado do canal, \hat{z} . Devemos evitar as longas tabelas para a variável \hat{z} , através de uma expressão para esta, se seguirmos o rotulamento dos sinais, bem como das suas regiões de decisão, da maneira que segue. Começamos definindo uma função que transforma um vetor binário na sua representação em base decimal, a saber

$$\text{dec}(\Psi; C_1, C_2, \dots, C_\Psi) = \sum_{j=1}^{\Psi} C_j 2^{j-1}. \quad (5.38)$$

O sinal correspondente ao vetor $(C_1, C_2, \dots, C_\Psi)$ deve ser rotulado como

$$S_{\text{dec}(\Psi; C_1, C_2, \dots, C_\Psi)}, \quad (5.39)$$

e a sua correspondente região de decisão como

$$\Lambda_{\text{dec}(\Psi; C_1, C_2, \dots, C_\Psi)}, \quad (5.40)$$

a ser determinada pelo teste de razão de máxima verossimilhança já mencionado.

Usando esta notação em qualquer das constelações tratadas neste capítulo, a variável \hat{z} , no i -ésimo estágio, pode ser determinada como segue:

$$\hat{z} = \hat{z}(\hat{C}_1, \hat{C}_2, \dots, \hat{C}_{i-1}, Y) = \begin{cases} 0; & \text{se } Y \in \left(K_1, K_{i+1}, \dots, K_{\Psi} \in \{0,1\} \cup \Lambda_{\text{dec}(\Psi; \hat{C}_1, \hat{C}_2, \dots, \hat{C}_{i-1}, K_1, K_{i+1}, \dots, K_{\Psi})} \right) \\ 1; & \text{caso contrário.} \end{cases} \quad (5.41)$$

Devemos notar que (5.41) fornece o mesmo resultado das Tabelas 5.1 e 5.2 para o caso da constelação 8-PSK.

Um limitante igualmente válido e extensível a todas as constelações deste capítulo para a probabilidade de ocorrer um erro na estimação do estado do canal é fornecido por:

$$P_{cs_i} \leq 1 - P[Y \in \Lambda_0], \quad i=2,3,\dots,\Psi. \quad (5.42)$$

Para caso da constelação M-QAM este limitante é:

$$P_{cs_i} = P_{cs} \leq 1 - \left[\int_{-\frac{\Delta_1}{2}}^{\frac{\Delta_1}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{\nu^2}{2\sigma^2}\right\} d\nu \right]^2$$

$$\approx 2 \operatorname{erfc}\left(\frac{\Delta_1}{2^{3/2}\sigma}\right); \quad i=2,3,\dots,\Psi. \quad (5.43)$$

Incluindo (5.43) em (5.36) e (5.37) o valor de $D^{(1)}$ é finalmente fornecido por:

$$D^{(2i-1)} \leq \left[N_1(\tau - i + 1) \exp\left\{\frac{c(0)^2(2^{i-1}\Delta_1)^2 - 2c(0)(2^{i-1}\Delta_1)^2}{8\sigma^2}\right\} + \right.$$

$$\left. + \left[\sum_{j=1}^{2i-2} P_{cj} \right] P_1(\tau - i + 1) \exp\left\{\frac{c(1)^2(2^{i-1}\Delta_1)^2}{8\sigma^2}\right\} \right] \cdot (1 - P_{cs}) +$$

$$+ \left[N_1(\tau - i + 1) \exp\left\{\frac{c(1)^2(2^{i-1}\Delta_1)^2 - 2c(1)(2^{i-1}\Delta_1)^2}{8\sigma^2}\right\} + \right.$$

$$\left. + \left[\sum_{j=1}^{2i-2} P_{cj} \right] P_1(\tau - i + 1) \exp\left\{\frac{c(0)^2(2^{i-1}\Delta_1)^2}{8\sigma^2}\right\} \right] \cdot P_{cs}; \quad i=2,\dots,\tau, \quad (5.44)$$

e

$$\begin{aligned}
D^{(2i)} = & \left[N_2(\tau - i + 1) \exp \left\{ \frac{c(0)^2(2^{i-1}\Delta_1)^2 - 2c(0)(2^{i-1}\Delta_1)^2}{4\sigma^2} \right\} \right] + \\
& + \left[\sum_{j=1}^{2^{i-1}} P_{c_j} \right] P_2(\tau - i + 1) \exp \left\{ \frac{c(1)^2(2^{i-1}\Delta_1)^2}{4\sigma^2} \right\} \cdot (1 - P_{cs}) + \\
& + \left[N_2(\tau - i + 1) \exp \left\{ \frac{c(1)^2(2^{i-1}\Delta_1)^2 - 2c(1)(2^{i-1}\Delta_1)^2}{4\sigma^2} \right\} \right] + \\
& + \left[\sum_{j=1}^{2^{i-1}} P_{c_j} \right] P_2(\tau - i + 1) \exp \left\{ \frac{c(0)^2(2^{i-1}\Delta_1)^2}{4\sigma^2} \right\} \cdot P_{cs}; \quad i=1,2,\dots,\tau. \quad (5.45)
\end{aligned}$$

A partir dos valores de $D^{(i)}$ para cada estágio fornecidos por (5.35), (5.44) e (5.45), podemos otimizar (5.44) e (5.45) com relação aos custos $c(\hat{z})$, o que conduz ao decodificador ótimo.

5.4. EXEMPLOS E RESULTADOS OBTIDOS

Nesta seção fornecemos alguns exemplos de Sistemas de Modulação Codificada Multinível, os quais fazem uso da estratégia de decodificação proposta.

Exemplo 5.1 : Vamos considerar o sistema de modulação codificada de 3 níveis que faz uso da constelação 8-PSK, particionada como mostra a Figura B.1 (veja Apêndice B). Os códigos componentes utilizados foram:

$$C_1: R_1 = 5/20, \quad 32 \text{ estados e } d_{\text{free}} = 20 \text{ [9];}$$

$$C_2: R_2 = 5/6, \quad 32 \text{ estados e } d_{\text{free}} = 4 \text{ [10];}$$

$$C_3: R_3 = 11/12, \quad 32 \text{ estados e } d_{\text{free}} = 3 \text{ [10],}$$

com uma eficiência espectral de 2 bits/sinal. A partir da equação (4.8) determinamos a distância Euclidiana quadrática do sistema como sendo 8. As Funções de Transferência destes códigos foram determinadas em computador [11], e os custos ótimos foram determinados graficamente para $SNR = 9\text{dB}$, quais sejam

- 2º estágio: $c(0) = 1$ (para normalização), $c(1) = 1.0$ (ver Fig. 5.6(a))
- 3º estágio: $c(0) = 1$ (para normalização), $c(1) = 0.58$ (ver Fig. 5.7(a)).

Devemos ressaltar que o fato do segundo estágio apresentar o custo ótimo $c(1) = 1$ implica que o estimador não contribui no processo de decodificação pois o código C_2 é capaz o suficiente para decodificar corretamente quando a informação lateral do primeiro estágio é confiável. Por outro lado, no terceiro estágio, vemos que o custo ótimo $c(1) = 0.58$ implica que o estimador contribui favoravelmente para o desempenho do sistema. Nas Figuras 5.6(b) e 5.7(b), os custos ótimos para o segundo e terceiro estágios, respectivamente, são fornecidos para várias relações sinal-ruído. Percebemos que para altas SNR , quando os decodificadores dos estágios anteriores podem ser considerados como poderosos, os estimadores contribuem pouco. Para baixas SNR , entretanto, a presença dos estimadores é fundamental.

Combinando (5.2), para $i=1,2,3$, como indicado em (5.1) a probabilidade de erro de bit do sistema versus $SNR = 10 \log_{10} \left(\frac{1}{2\sigma^2} \right)$ (dB) é traçada na Figura 5.8. O decodificador proposto em [2] é também considerado. Podemos observar um ganho de 0.2 dB para altas SNR e de 0.4 dB para baixas SNR a favor do decodificador multiestágio ótimo proposto.

Exemplo 5.2 : Vamos considerar o sistema de modulação codificada de 4 níveis que faz uso da constelação 16-QAM, particionada como mostra a Figura B.2 (veja Apêndice B), onde Δ_1 deve ser igual a 0.632 para potência média unitária. Baseando-se nas tabelas para a distância mínima de códigos lineares fornecidas por

Verhoeff [12], devem existir códigos convolucionais:

$$C_1: R_1 = 12/25 \text{ e } d_{\text{free}} = 20;$$

$$C_2: R_2 = 12/15 \text{ e } d_{\text{free}} = 10;$$

$$C_3: R_3 = 6/7 \text{ e } d_{\text{free}} = 5;$$

$$C_4: R_4 = 11/12 \text{ e } d_{\text{free}} = 3.$$

os quais podem ser usados como códigos componentes. A eficiência espectral R para estas taxas é de 3 bits/sinal. A partir da equação (4.8) determinamos a distância Euclidiana quadrática do sistema como sendo 8. É importante observar que para esta mesma distância do sistema (portanto, para o mesmo desempenho) pode-se, a partir das tabelas em [12], conseguir uma taxa R de aproximadamente 3.23 bits/ sinal com os códigos:

$$C_1: R_1 = 9/13 \text{ e } d_{\text{free}} = 20;$$

$$C_2: R_2 = 14/16 \text{ e } d_{\text{free}} = 10;$$

$$C_3: R_3 = 6/7 \text{ e } d_{\text{free}} = 5;$$

$$C_4: R_4 = 4/5 \text{ e } d_{\text{free}} = 3.$$

Os custos ótimos foram determinados graficamente para $\text{SNR} = 13 \text{ dB}$, quais sejam:

- 2º estágio: $c(0) = 1$ (para normalização), $c(1) = 1.0$ (ver Fig. 5.9(a))
- 3º estágio: $c(0) = 1$ (para normalização), $c(1) = 0.88$ (ver Fig. 5.10(a))
- 4º estágio: $c(0) = 1$ (para normalização), $c(1) = 0.49$ (ver Fig. 5.11(a))

Nas Figuras 5.9(b), 5.10(b) e 5.11(b) os custos ótimos para o segundo, terceiro e quarto estágios, respectivamente, são fornecidos para várias relações sinal-ruído. A probabilidade de erro de bit do sistema versus SNR é traçada na Figura 5.12. O decodificador convencional de Imai e Hirakawa é também considerado. O sistema proposto apresentou um ganho para baixas SNR de 0.4 dB, que é um ganho considerável face ao pequeno acréscimo de complexidade.

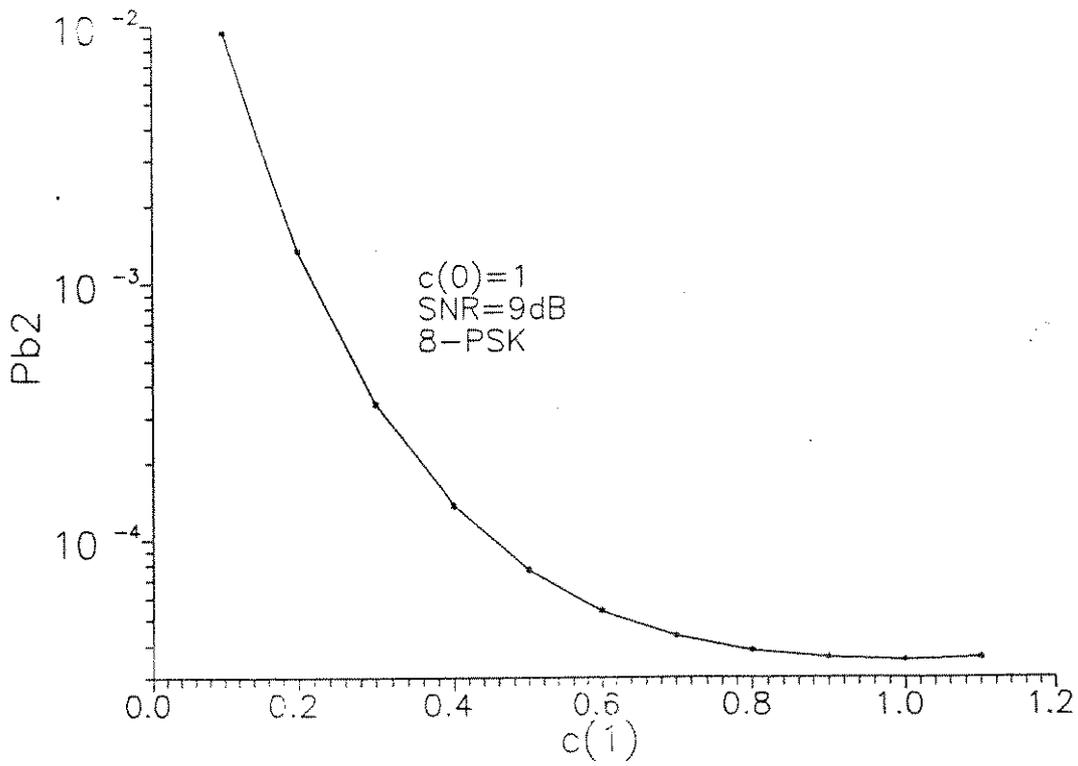


Figura 5.6(a) - Custo Ótimo no Segundo Estágio do Exemplo 5.1 para SNR = 9dB.

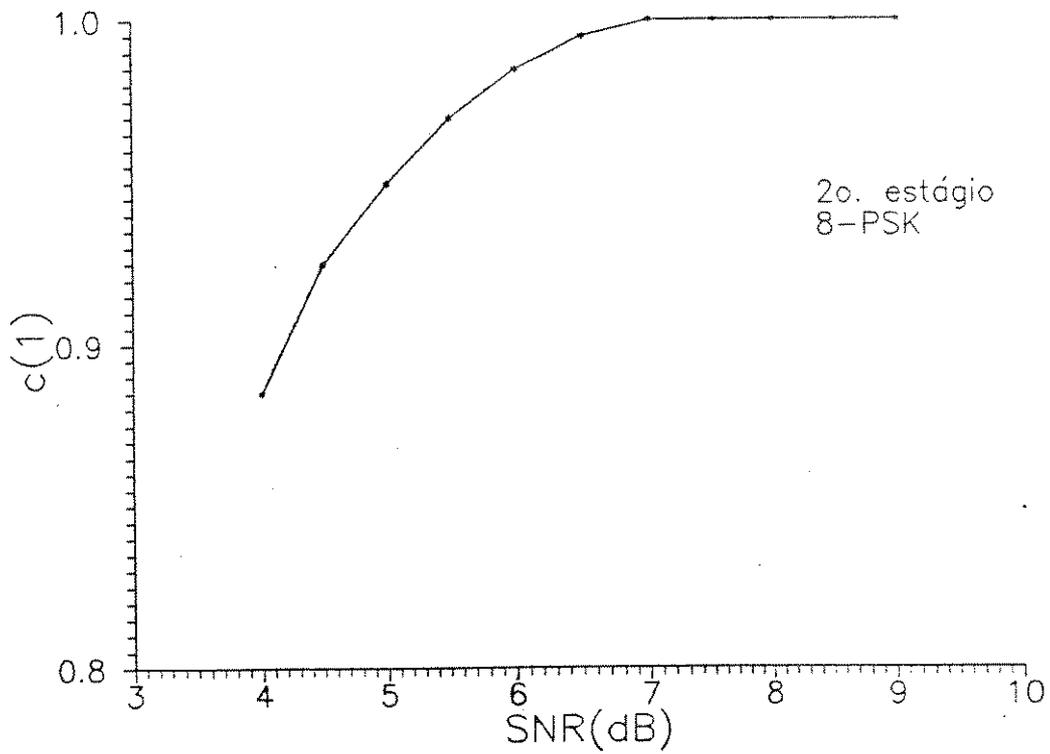


Figura 5.6(b) - Custo Ótimo x SNR no Segundo Estágio do Exemplo 5.1.

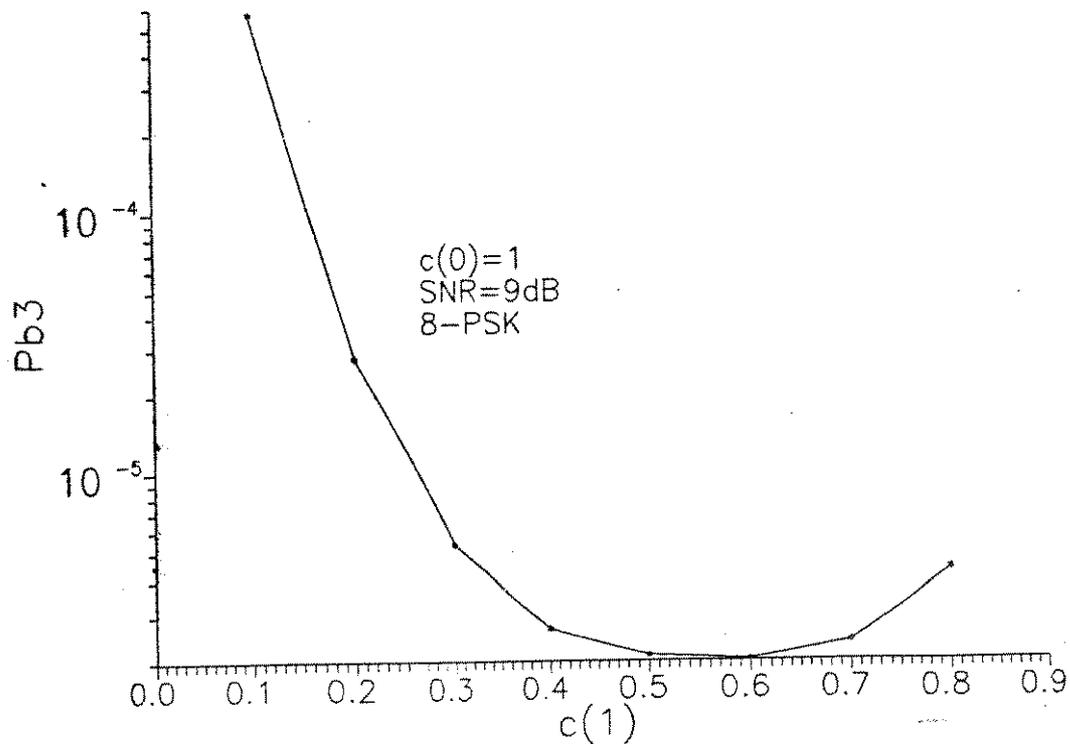


Figura 5.7(a) - Custo Ótimo no Terceiro Estágio do Exemplo 5.1 para SNR = 9dB.

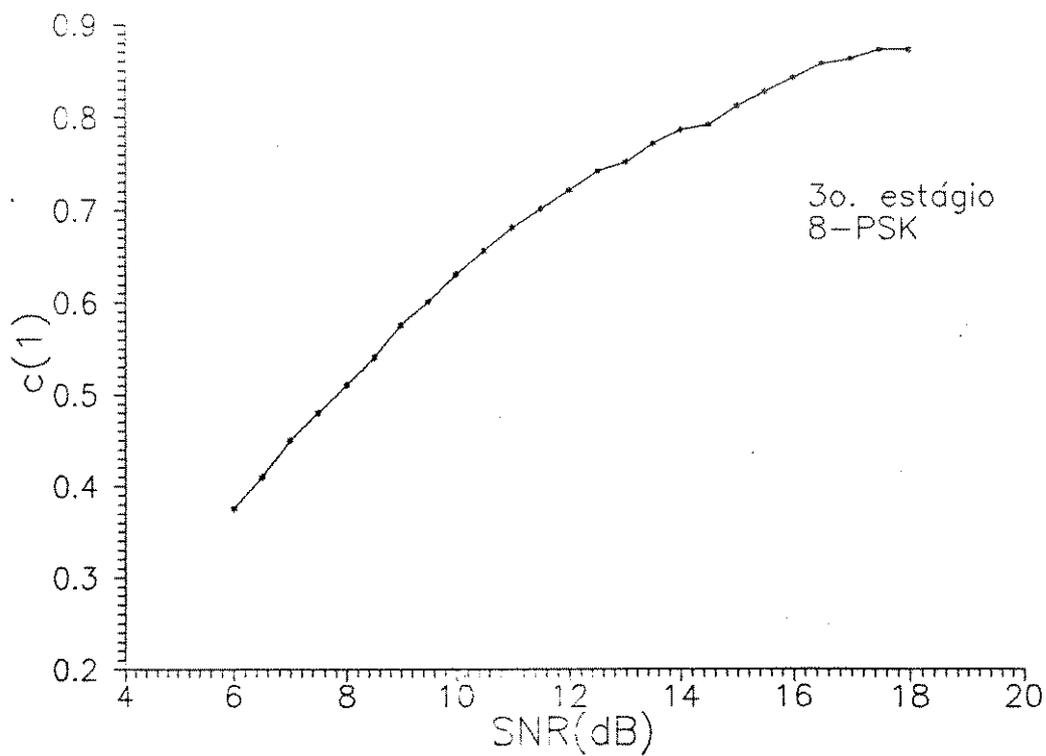


Figura 5.7(b) - Custo Ótimo x SNR no Terceiro Estágio do Exemplo 5.1.

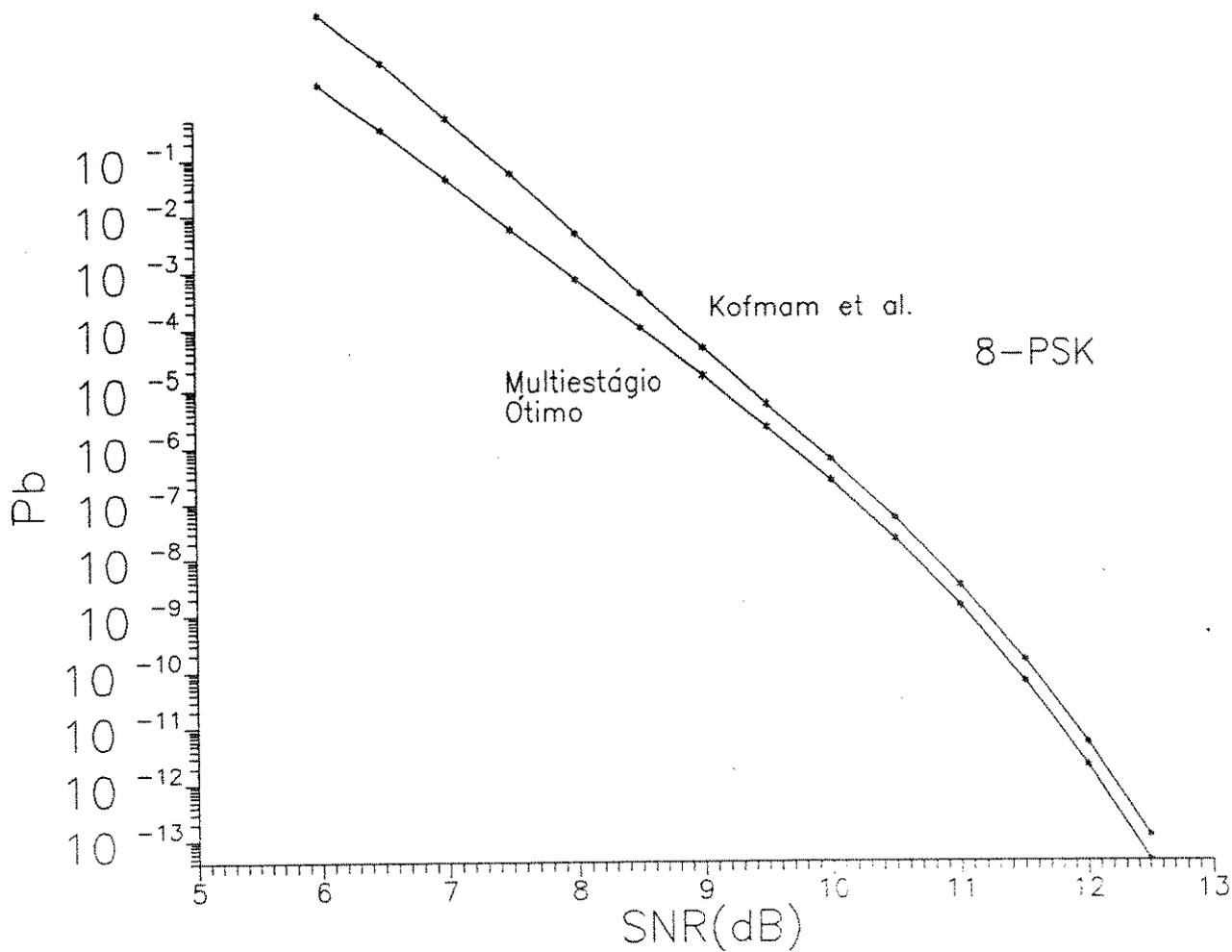


Figura 5.8 - Comparação de Desempenho para a Constelação 8-PSK.

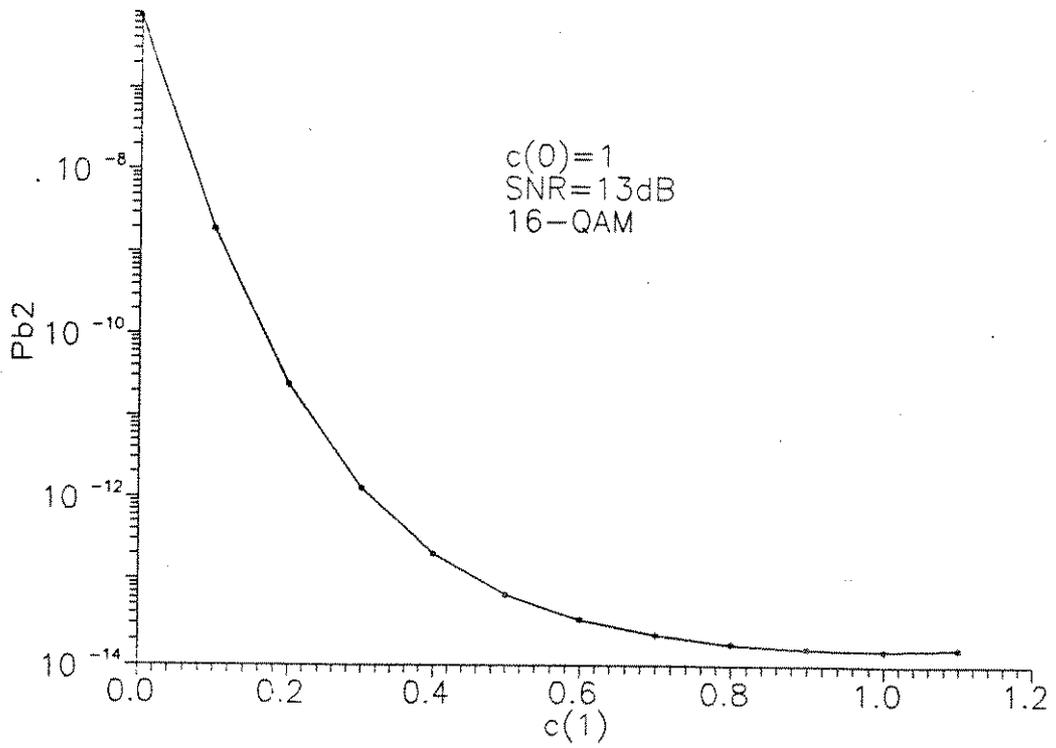


Figura 5.9(a) - Custo Ótimo no Segundo Estágio do Exemplo 5.2 para SNR = 13 dB.

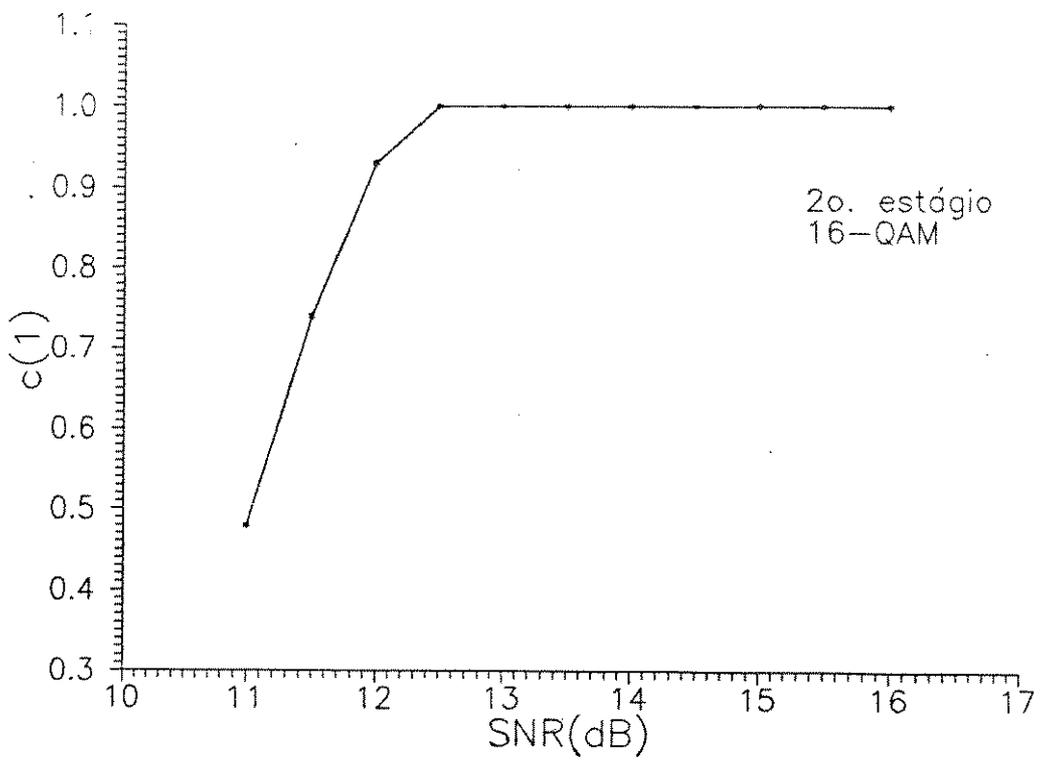


Figura 5.9(b) - Custo Ótimo x SNR no Segundo Estágio do Exemplo 5.2.

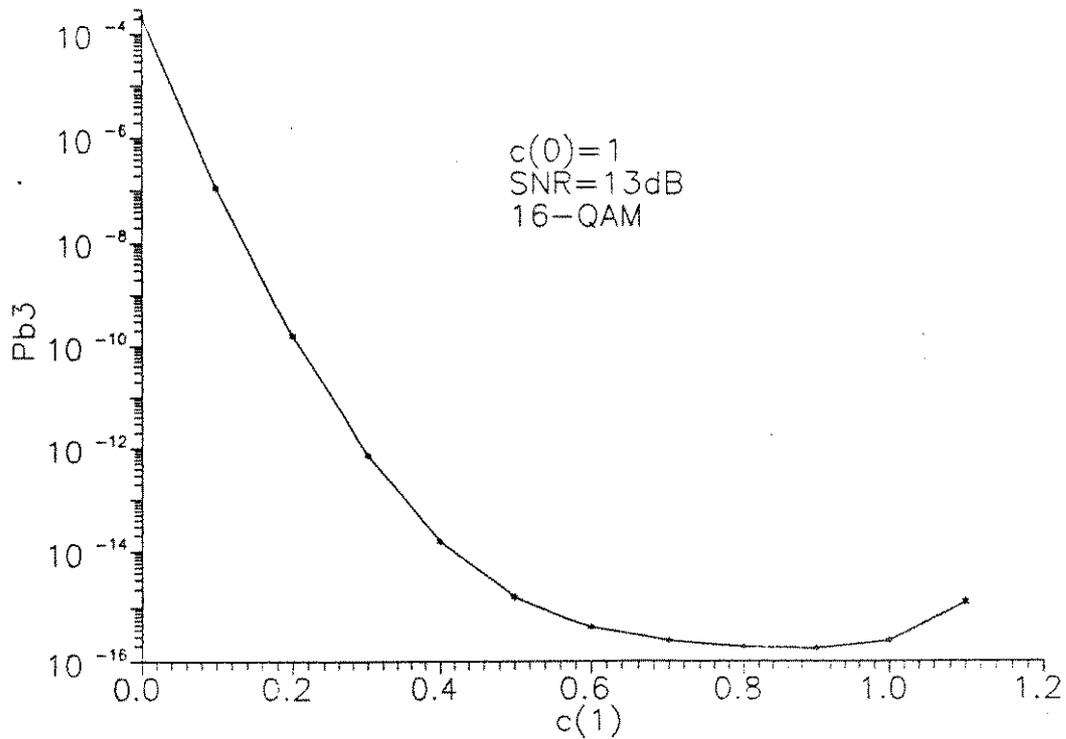


Figura 5.10(a) - Custo Ótimo no Terceiro Estágio do Exemplo 5.2 para SNR=13dB.

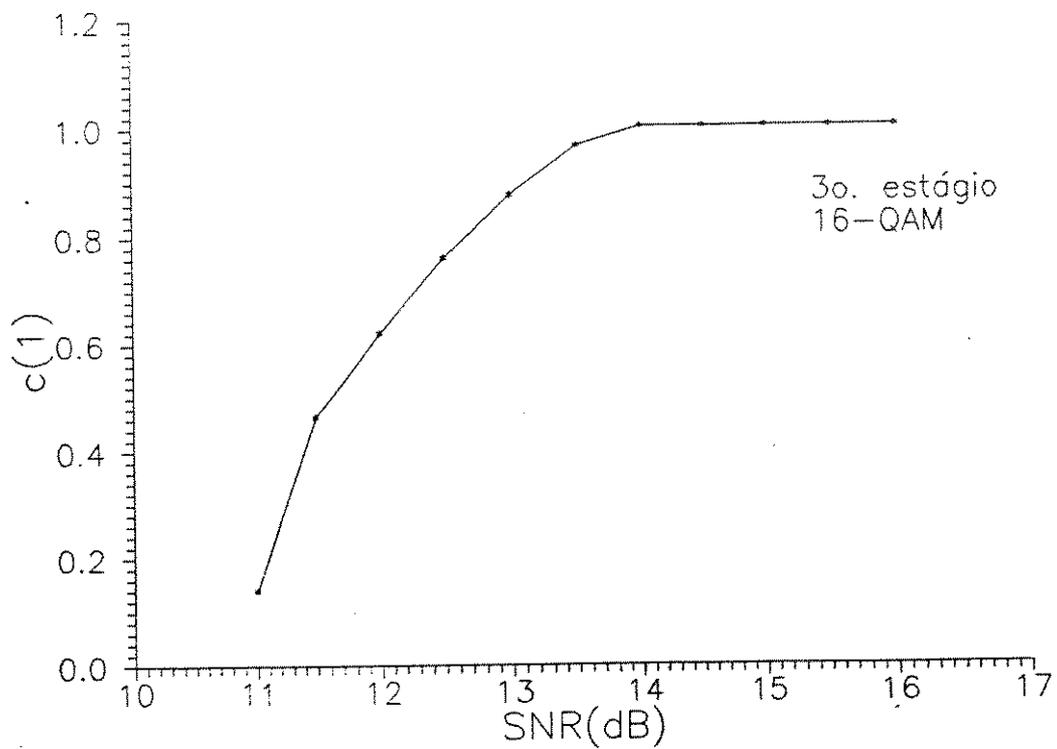


Figura 5.10(b) - Custo Ótimo x SNR no Terceiro Estágio do Exemplo 5.2.

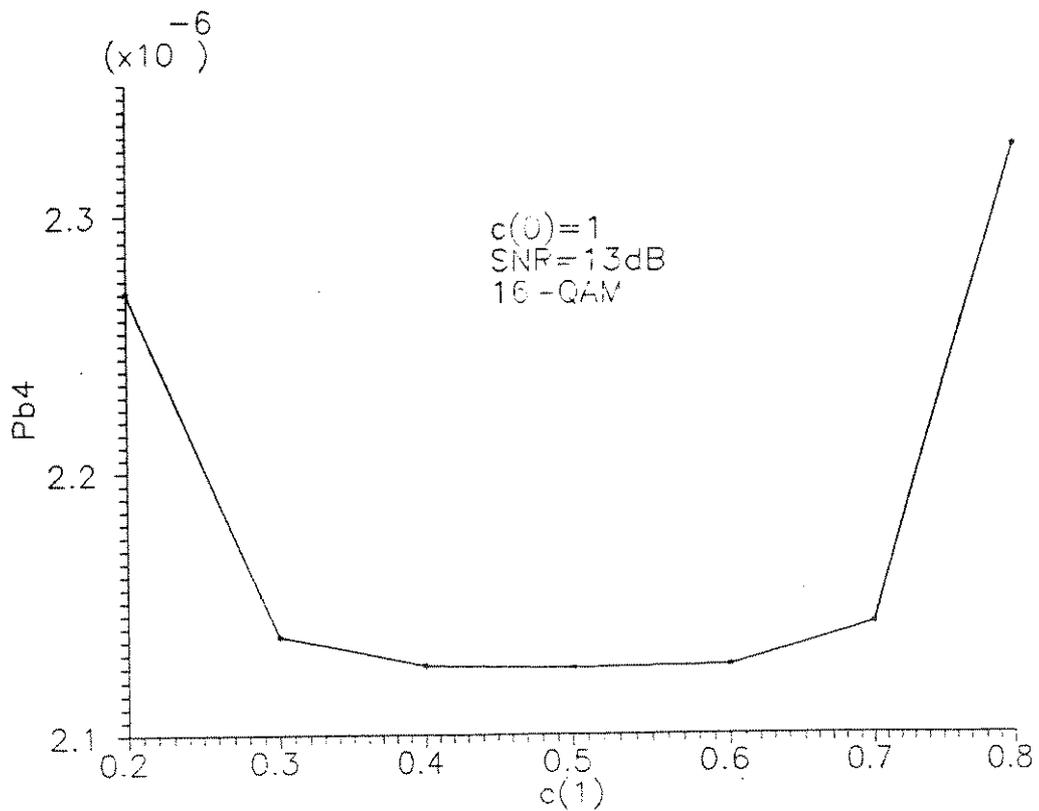


Figura 5.11(a) - Custo Ótimo no Quarto Estágio do Exemplo 5.2 para SNR = 13dB.

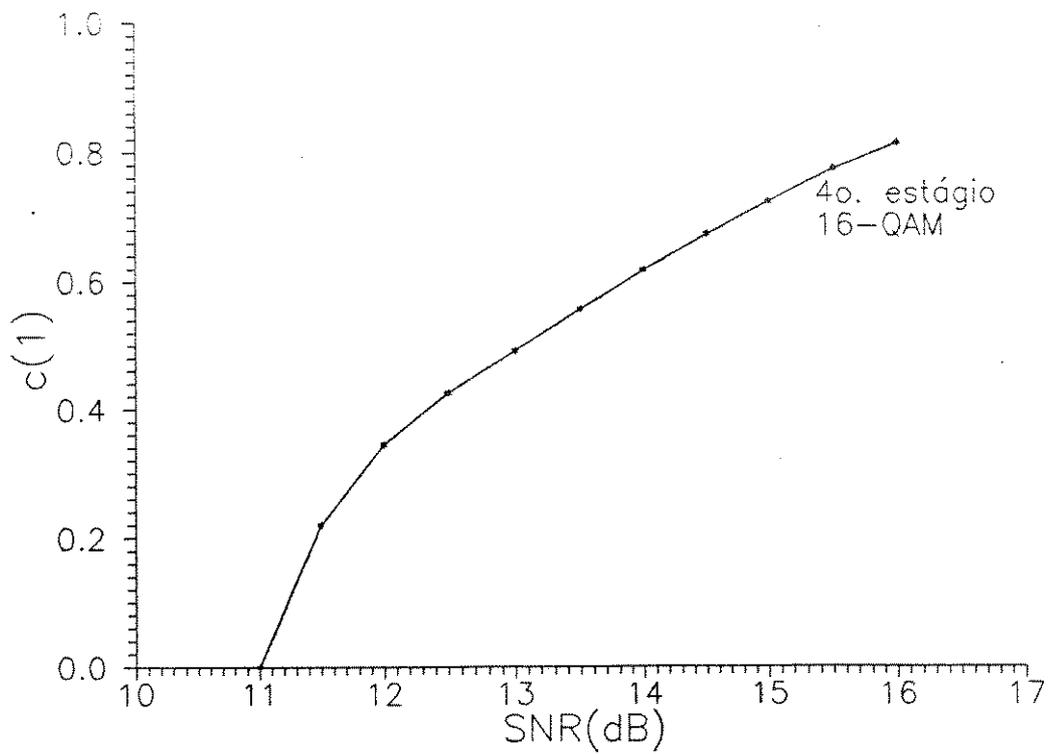


Figura 5.11(b) - Custo Ótimo x SNR no Quarto Estágio do Exemplo 5.2.

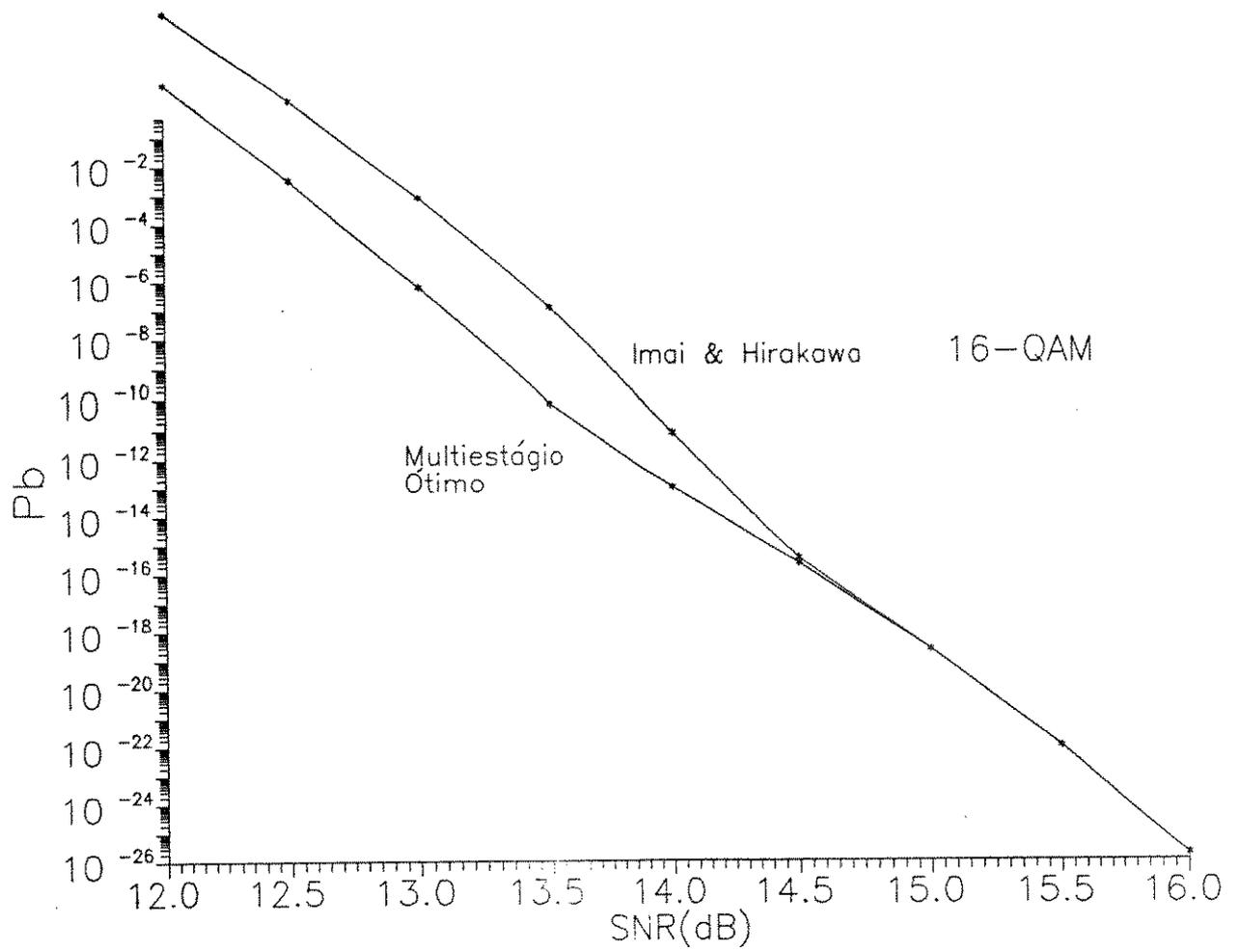


Figura 5.12 - Comparação de Desempenho para a Constelação 16-QAM.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error Correcting Codes", *IEEE Trans. Information Theory*, Vol. IT-23, pp. 371-377, May 1977.
- [2] Y. Kofman, S. Shamai (Shitz) and E. Zehavi, "Analysis of a Multilevel Coded Modulation System", BILKNET'90, Ankara, Turkey, Jul. 1990.
- [3] M.K. Simon, J.K. Omura, R.A. Sholtz and B.K. Levitt, *Spread Spectrum - Communications - Volume I*, (Computer Science Press, 1985).
- [4] J.K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis", *IEEE Trans. Information Theory*, Vol. IT-24, No. 1, Jan. 1978.
- [5] G.J. Pottie and D.P. Taylor, "Multilevel Codes Based on Partitioning", *IEEE Trans. Information Theory*, Vol. IT-35, No. 1, pp. 87-98, Jan. 1989.
- [6] S. Haykin, *Digital Communications* (Wiley, 1988).
- [7] H.L. Van Trees, *Detection, Estimation and Modulation Theory, Part I* (Wiley, 1968).
- [8] A.D. Whalen, *Detection of Signals in Noise* (Academic Press, 1971).
- [9] A. Said, and R. Palazzo, Jr., "Using Combinatorial Optimization to Design Good Unit Memory Convolutional Codes", *IEEE Trans. Information Theory*, submitted for publication.

- [10] Y. Yasuda, K. Kashiki and Y. Hirata, "High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding", *IEEE Trans. Communications*, Vol. 22, No. 3, pp. 315-319, Mar. 1984.
- [11] M.C.P. Young, "Obtenção de Códigos Convolucionais Ótimos de Memória Unitária por Programação Matemática, Tese de Mestrado, UNICAMP, 1989.
- [12] T. Vernoeff, "An Updated Table of Minimum-Distance Bounds for Binary Linear Codes", *IEEE Trans. Information Theory*, Vol. IT-23, No. 5, Sept. 1987.
- [13] B.F. Uchôa Filho and R. Palazzo Jr., "Performance Analysis of a Multistage Decoding Strategy for Multilevel Coded Modulation Systems", aceito no The Third IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Boston, Massachusetts, Oct. 1992.
- [14] B.F. Uchôa Filho and R. Palazzo Jr., "Performance Analysis of a Multistage Decoding Strategy for a Class of Geometrically Uniform Signal Sets", IEEE ITW92 - Information Theory Workshop 1992, Salvador, Bahia, Jun. 1992 e a ser submetido para publicação no *IEEE Trans. Communications*.

CAPÍTULO 6

CONCLUSÕES

6.1. COMENTÁRIOS E CONCLUSÕES

Nesta tese foi proposta uma estratégia de decodificação para Sistemas de Modulação Codificada Multinível, baseada no decodificador de Imai/Hirakawa (convencional). Na verdade, esta pode ser entendida como sendo uma classe de decodificadores multiestágio da qual o decodificador multiestágio convencional (aquele para $c(0) = c(1) = 1$), bem como o decodificador multiestágio ótimo (aquele obtido a partir dos custos otimizados), são um caso especial.

Um procedimento geral para se obter um limitante superior da probabilidade de erro de bit destes esquemas foi estabelecido, onde as constelações M-PSK e M-QAM foram analisadas. A constelação 8-PSK foi usada em um sistema de modulação codificada de 3 níveis com três codificadores convolucionais distintos. A curva da probabilidade de erro de bit versus SNR do sistema total usando a estratégia de decodificação proposta foi então traçada. O ganho para $\text{SNR} \geq 9$ dB foi de 0.2 dB sobre o decodificador modificado de Kofman *et. al.*. Para $\text{SNR} < 9$ dB o ganho foi maior que 0.2 dB. Uma curva teórica para a constelação 16-QAM foi também traçada com uma eficiência espectral igual a 3 bits/sinal. A estratégia de decodificação proposta apresentou um ganho de aproximadamente 0.4 dB sobre o decodificador de Imai/Hirakawa, para baixas relações sinal-ruído.

Notamos, a partir das curvas de desempenho das Figuras 6.8 e 6.12, que a estratégia proposta provê ganhos maiores para baixas relações sinal-ruído. Deste modo, ela oferece uma contribuição substancial em aplicações tais como Comunicações Móveis, onde a operação nesta faixa de potência é requerida.

A estratégia proposta pode ser facilmente incorporada aos sistemas práticos. Nenhum aumento relevante de complexidade ou de tempo de decodificação foi introduzido, uma vez que os mesmos elementos básicos são utilizados tanto na estimação de estado de canal como no cômputo da métrica.

Como sugestões para trabalhos futuros apresentamos o seguinte:

- Medir o desempenho da estratégia proposta através de simulações, e comparar com os "bounds" teóricos obtidos neste trabalho;
- Estabelecer critérios para a escolha das taxas (ex.: via Teoria de Informação (Kofman et al.)) e dos códigos componentes;
- Tratar o problema estudado na tese para canais com desvanecimento, bem como usar códigos convolucionais variantes no tempo para canais variantes no tempo;
- Modificar a Função de Transferência para obter as Probabilidades de Erro de Bit individual (α e β , Seção 3.7.1), para aplicações que necessitam de proteção desigual de erro.

APÊNDICE A

O LIMITANTE DE CHERNOFF

Para se chegar ao limitante de Chernoff [1] devemos considerar as duas funções abaixo:

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (\text{A.1})$$

e

$$e_\lambda(t) = e^{\lambda t}, \quad \text{para todo } t, \quad (\text{A.2})$$

que são mostradas na Figura A.1.

Percebemos claramente que

$$u(t) \leq e_\lambda(t), \quad \text{para } \lambda \geq 0. \quad (\text{A.3})$$

Agora seja X uma variável aleatória com função densidade de probabilidade $p_x(x)$ para todo x . Sendo assim, devemos ter o seguinte:

$$\Pr\{X \geq 0\} = \int_0^{\infty} p_x(t) dt = \int_{-\infty}^{\infty} u(t) p_x(t) dt \leq \int_{-\infty}^{\infty} e_\lambda(t) p_x(t) dt = E\{e^{\lambda X}\}. \quad (\text{A.4})$$

Então,

$$\Pr\{X \geq 0\} \leq E\{e^{\lambda X}\}, \quad \text{para } \lambda \geq 0 \quad (\text{A.5})$$

é a forma final do limitante de Chernoff, onde λ deve ser escolhido de maneira a minimizar a esperança matemática em (A.5).

Similarmente, pode-se mostrar que este limitante também é válido para o caso em que X é uma variável aleatória discreta.

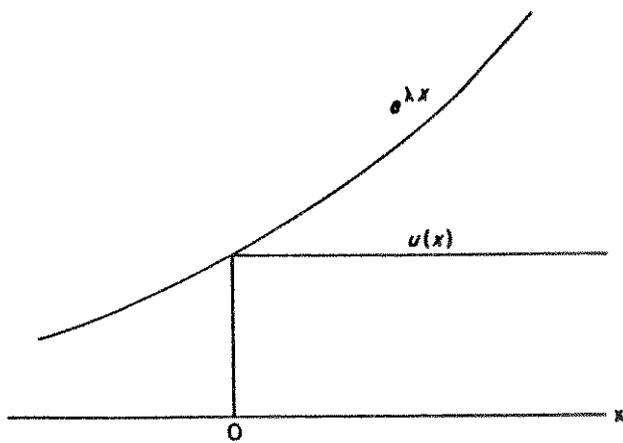


Figura A.1 - Funções $u(x)$ e $e_{\lambda}(x)$.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] M.K. Simon, J.K. Omura, R.A. Sholtz and B.K. Levitt, *Spread Spectrum - Communications - Volume I*, (Computer Science Press, 1985).

APÊNDICE B

PARTIÇÃO DE CONJUNTO

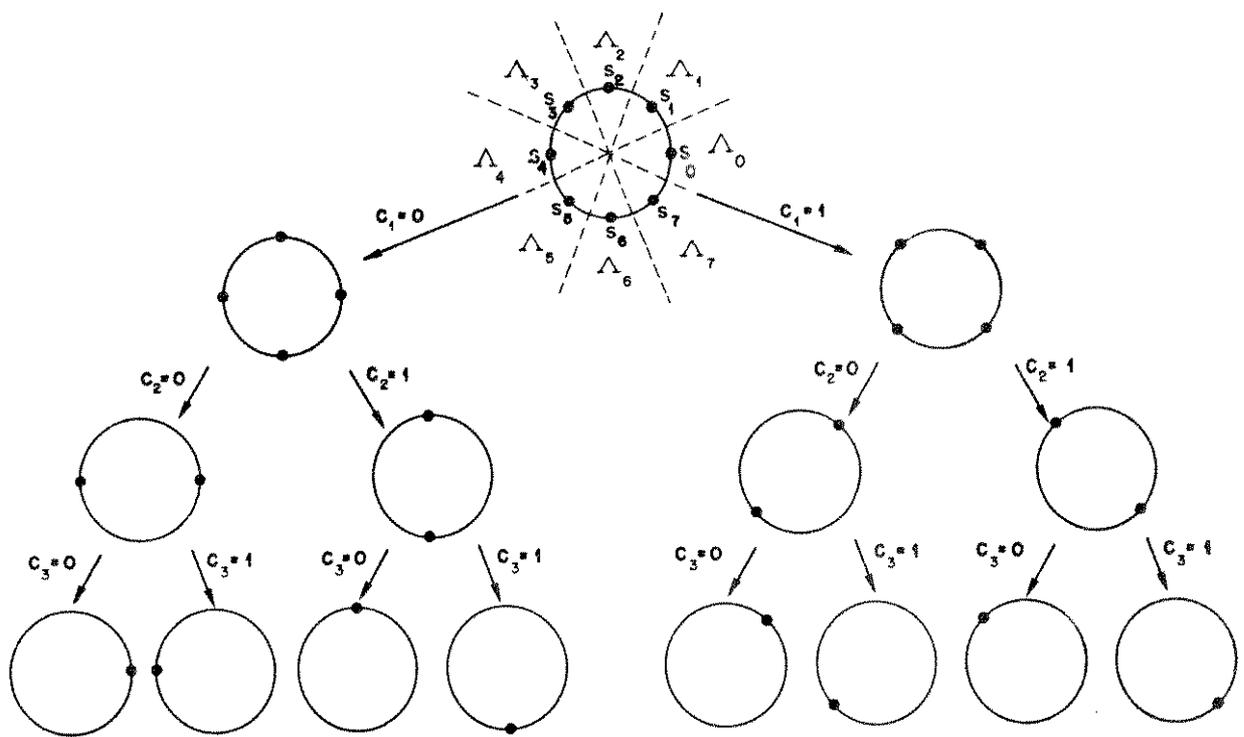


Figura B.1 - Partição da Constelação 8-PSK.

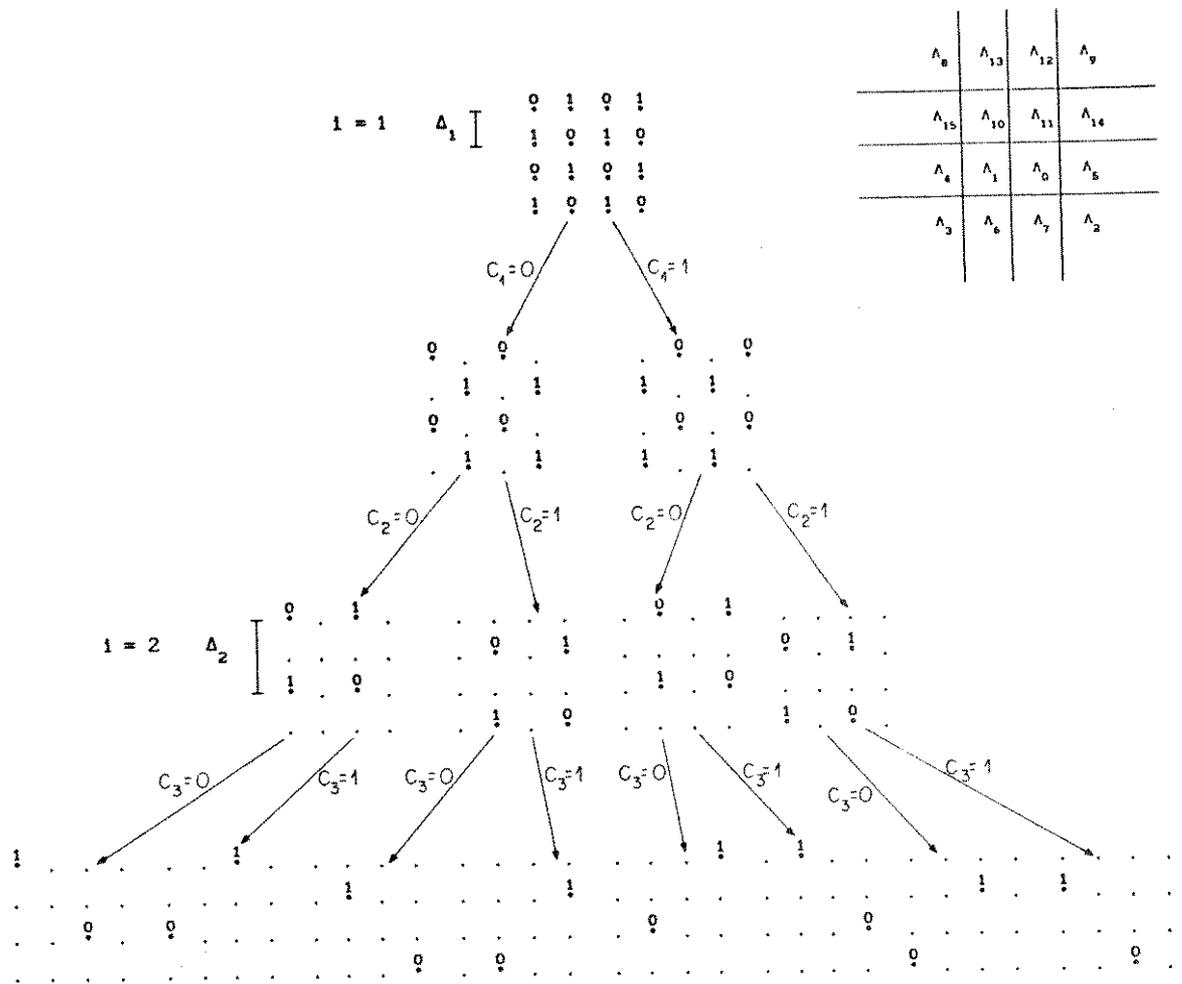


Figura B.2 - Partição da Constelação 16-QAM.