

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

de acordo,

Dalton Soares Arantes

17 / Agosto 1987

"CÓDIGOS CONVOLUCIONAIS PARA MODULAÇÃO CODIFICADA"

JOÃO BATISTA BEZERRA
Orientador : PROF. DR. DALTON SOARES ARANTES

Tese apresentada à Faculdade de Engenharia Elétrica, da Universidade Estadual de Campinas - UNICAMP - como parte dos requisitos exigidos para a obtenção do título de DOUTOR EM ENGENHARIA ELÉTRICA.

MARÇO 1987

UNICAMP

BIBLIOTECA CENTRAL

A minha esposa, Luzia;
Aos meus filhos,
Cláudio Emanuel,
Ana Cláudia e
João Luiz

R E S U M O

Os sistemas de modulação codificada que usam códigos convolucionais para controlar a transmissão dos símbolos associados à modulação, são estudados utilizando uma abordagem que se fundamenta no modelamento do codificador associado ao código convolucional como uma máquina sequencial de Mealy.

O modelo desenvolvido é utilizado na otimização de códigos convolucionais a serem usados em conjunção com uma dada constelação de sinais, cujos símbolos são supostos pertencerem a um espaço multidimensional. Este mesmo modelo é utilizado para avaliar as propriedades espectrais de sinais codimodulados e, desta forma, determinar o efeito da dimensionalidade na faixa de transmissão.

O desempenho dos códigos seleccionados é obtido por meio de simulação do sistema, como também por meio de aproximações para as probabilidades de erro de evento e de bit.

AGRADECIMENTOS

Ao meu Orientador, Prof. Dr. Dalton Soares Arantes, pelo interesse demonstrado neste trabalho, pela orientação e pela confiança em mim depositada;

Ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Norte - UFRN, pela oportunidade proporcionada;

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, pelo apoio financeiro parcial através da concessão de Bolsa de Pós-Graduação;

Ao Centro de Pesquisas e Desenvolvimento - CPqD, da Telebrás pelo suporte colocado a minha disposição através do Convênio UNICAMP/TELEBRÁS 024A/83;

Aos meus familiares, pelo apoio e incentivo que sempre recebemos;

Aos meus colegas.

ÍNDICE

CAPÍTULO 1 - Considerações preliminares e objetivos	1
1.1 - Considerações preliminares	1
1.2 - Objetivos	3
CAPÍTULO 2 - Estrutura e modelo para o codificador de um código convolucional	4
2.1 - Introdução	4
2.2 - Caracterização de um código convolucional.	4
2.3 - Modelagem do codificador.	5
2.3.1 - Caracterização do vetor de estado.	6
2.3.2 - Caracterização da interconexão de memórias.	9
2.3.4 - O codificador como uma máquina de Mealy.	10
2.3.5 - Representação matricial das funções do codificador.	11
2.4 - Flexibilidade do modelo	13
2.5 - Complexidade computacional do modelo.	17
2.6 - Generalidade do modelo.	17
APÊNDICE 2A	19
APÊNDICE 2B	20
CAPÍTULO 3 - Estrutura e modelo para um código de treliça associado a um código convolucional	22
3.1 - Introdução	22
3.2 - Mapeamento entre e palavras código e constelação	22

3.3 - O mapeamento por partição de conjunto	25
3.4 - Caracterização de um código de treliça	28
3.5 - Desempenho de um código de treliça	36
3.6 - Estrutura do decodificador	36
3.6.1 - Princípios básicos do algoritmo de Viterbi	36
3.6.2 - O algoritmo de Viterbi para códigos de treliça	39
CAPÍTULO 4 - Otimização de códigos convolucionais associados a códigos de treliça	43
4.1 - Introdução	43
4.2 - A complexidade da seleção exaustiva sem regras desejeção	43
4.3 - O MPC na redução do universo da busca exaustiva	45
4.4 - O procedimento de busca	48
4.5 - Cálculo da distância livre	48
4.6 - Simplificações no cálculo da distância livre	52
4.7 - Otimização de códigos com máxima distância livre	57
APÊNDICE 4A	60
CAPÍTULO 5 - Propriedades espectrais de sinais codimodulados	63
5.1 - Introdução	63
5.2 - Caracterização temporal do sinal codimodulado	64
5.3 - Espectro do sinal codimodulado	66

5.4 - Espectro para sinais sem codificação	68
APÊNDICE 5A	70
APÊNDICE 5B	72
APÊNDICE 5C	73
CAPÍTULO 6 - Códigos de treliça para constelações tri- dimensionais	75
6.1 - Introdução	75
6.2 - Capacidade de Canal para constelações multi- dimensionais	76
6.3 - Avaliação da margem de ganho teórico	79
6.4 - Margem de ganho para o sistema OCTAL CÚBICO relativo ao sistema BPSK	80
6.5 - Margem de ganho para o sistema OCTAL CÚBICO relativo ao sistema OCTAL PRISMÁTICO	81
6.6 - Caracterização temporal e espectral para constelações em até tres dimensões	83
6.6.1 - Constelações bidimensionais	83
6.6.2 - Constelações tridimensionais	86
6.7 - Espectro para as constelações OCTAL CÚBICA, OCTAL PRISMÁTICA e BPSK	89
6.8 - Códigos com máxima distância livre para as constelações BPSK e OCTAL CÚBICO	90
6.9 - Otimização de códigos com máxima distância livre para 4PSK codificado em BPSK e OCTAL CÚBICO	95

CAPÍTULO 7 - Simulação de sistemas com modulação codificada	103
7.1 - Introdução	103
7.2 - Modelo para um sistema de transmissão com modulação codificada	103
7.2.1 - Fonte q-ária e conversor S/P	106
7.2.2 - Codificação	107
7.2.3 - Codimodulação	108
7.2.4 - Caracterização do canal	109
7.2.5 - Receptor	110
7.2.6 - Contagem de erros de evento e de bit	111
7.3 - Estimativa das probabilidades de erro	112
7.4 - Teste do modelo da simulação	114
7.5 - Simulação do código (46 612) para codificação do sistema 4PSK no 8PSK	121
7.6 - Simulação do código (24 654) para codificação do sistema 4PSK no OCTAL CÚBICO	125
CAPÍTULO 8 - Conclusões e comentários	128
REFERENCIAS BIBLIOGRÁFICAS	130

CAPÍTULO 1

CONSIDERAÇÕES PRELIMINARES E OBJETIVOS

1.1 - Considerações preliminares

Um aspecto fundamental no desenvolvimento de qualquer meio de comunicações é assegurar a confiabilidade do sistema ao usuário, porém utilizando de forma eficiente os recursos físicos disponíveis (energia, espectro de frequências, meio de transmissão, memória, etc.) com os mínimos custos possíveis.

Em geral, infelizmente, para cada aplicação particular, a disponibilidade de recursos se apresenta de forma desigual e sem correlação, dependendo do tipo de transmissão (banda base ou rádio frequência), características físicas do canal (cabo, fibra óptica, meios de armazenamento em memória, enlaces de rádio, enlace satélite), das condições de operação (ruído de canal, interferência co-canal e de canais adjacentes, interferência proposital), da aplicação a que se destina (militar, espacial, controle de outros sistemas, prestação de serviços ao cidadão comum e a empresas), além da obediência às normas técnicas vigentes e que estão sujeitas a acordos internacionais. Um outro aspecto importante, é que a integração entre povos, nações e mesmo entre regiões de um mesmo país seja assegurada. Esse aspecto implica na necessidade da interligação entre sistemas com características diferentes, tanto no aspecto do processamento quanto das condições tecnológicas (rede local, rede regional e rede internacional). Uma situação nítida desse fenômeno se cristaliza com a implantação dos sistemas de transmissão digital que, por pertencerem a uma realidade tecnológica relativamente recente, têm que assegurar a compatibilidade com os sistemas analógicos já existentes.

Este trabalho de dissertação está relacionado, de maneira geral, com o desenvolvimento de sistemas de comunicações com transmissão digital e, especificamente, com o problema do uso eficiente da energia disponível e da disponibilidade de faixa de transmissão no espectro de frequências.

A colocação desse problema pode ser inserida ao longo do tempo a partir do instante em que se procurou utilizar códigos para proteger dados de escrita/leitura armazenados ou retirados das memórias de massa dos computadores. Esse problema de escrita/leitura em memória de massa é um caso típico de transmissão digital em banda base. Como a codificação associa a cada bloco de k bits um outro bloco de n bits, $n > k$, o aumento da taxa de transmissão, num processamento serial, implica numa expansão da faixa de transmissão de um fator n/k . O próximo passo foi a utilização de sistemas de transmissão digital com codificação para a aquisição de dados gerados por sondas espaciais. Aqui, a codificação seria utilizada para compensar as difíceis condições de operação das sondas (limitação na disponibilidade de energia

e era implementada em banda base, seguida de modulação digital. Assim, a expansão da faixa em banda base era transferida para o espectro de rádio-freqüências. Entretanto, nessas aplicações, em geral não há congestionamento de espectro, e portanto o problema da expansão de faixa não tinha tanta importância.

As dificuldades começariam a surgir no momento em que se passou a usar transmissão digital via rádio a taxas de transmissão sempre crescentes. Nesse caso, a faixa de transmissão é restrita e a disponibilidade de energia em princípio não é problema. Entretanto, para evitar interferência inter-canal, o uso da energia está limitado a níveis estabelecidos por normas técnicas. Assim, com a potência usada no limite, a limitação na faixa de transmissão e a necessidade crescente do aumento da taxa de transmissão, a solução seria o uso de esquemas mais complexos de modulação digital multinível (possibilitando economia de faixa). Neste caso, a maior necessidade de potência do esquema de modulação seria compensada pelo uso de uma codificação adequada em banda base. Para taxas de transmissão sempre crescentes, a solução seria um esquema de transmissão onde a codificação não mais seria efetivada em banda base, mas simultaneamente com a modulação. Um sistema com essas características foi introduzido por Anderson (1978) e pertence a uma classe mais genérica denominada de CPM (Continuous Phase Modulation) que assegura uso eficiente de potência e conformação espectral (Anderson e outros, 1981; Aulin, Sundberg e Rydbeck, 1981). Uma outra classe de sistemas foi introduzida por Ungerboeck (1982) que assegura uso eficiente de potência com conservação de faixa. Essa classe de sistemas usa um código convolucional para controlar a transmissão dos elementos de uma constelação de sinais, de forma redundante e de acordo com regras bem estabelecidas. Aqui, o principal objetivo é escolher o código convolucional que assegura ao sistema o melhor rendimento para uma dada complexidade de decodificação. Esse problema foi tratado por Ungerboeck (1982) para códigos de taxa $k/(k+1)$, para constelações de sinais em uma e duas dimensões e por Borelli (1983) para códigos de taxa k/n e constelações com múltiplas fases/amplitudes. Entretanto, existem aplicações onde o maior problema é a limitação na disponibilidade de energia. Esse é o caso dos sistemas de transmissão via satélite cujos amplificadores de potência são levados a operar próximo da saturação. O uso, o mais eficiente possível, da potência disponível é fundamental, mesmo que tenha que ser pago algum preço como, por exemplo, uma certa expansão da faixa de transmissão. Nesse sentido, este trabalho procura investigar a possibilidade do uso de constelações multidimensionais, tratado por Gersho e Lawrence (1984) para códigos de bloco, em sistemas usando códigos convolucionais com a concepção proposta por Ungerboeck.

1.2 - Objetivos

O principal objetivo deste trabalho, como descrito anteriormente, consiste em estudar a viabilidade do uso de constelações multidimensionais em conjunção com códigos de Ungerboeck. A ênfase será dada para constelações tridimensionais e em particular àquelas que possam ser usadas em canais não lineares, tais como os "transponders" dos sistemas de transmissão via satélite. Entretanto, para se atingir esse objetivo é necessário que sejam providos os meios que possibilitem: 1) Averiguar o efeito sobre as margens de ganho teóricas do esquema de codificação para constelações tridimensionais relativamente às constelações em uma e/ou duas dimensões; 2) Desenvolver procedimentos sistemáticos que possibilitem a seleção de códigos ótimos para uma constelação específica e uma dada complexidade de decodificação; 3) Averiguar os efeitos do aumento da dimensionalidade na faixa de transmissão do sinal codimodulado; 4) Prover meios que possibilitem a avaliação do desempenho do sistema para um código selecionado previamente.

No intuito de atingir as metas acima especificadas, o trabalho está organizado da seguinte forma: No capítulo 2 será desenvolvido um modelo para a caracterização do codificador de um código convolucional fundamentado na teoria das máquinas autônomas com número finito de estados (Kohavi, 1970). No capítulo 3, o modelo desenvolvido no capítulo 2, juntamente com as regras de codificação estabelecidas por Ungerboeck, serão utilizados no desenvolvimento de um modelo para caracterização da codificação e decodificação dos sistemas com modulação codificada. O capítulo 4 trata do problema da otimização de códigos convolucionais para uma dada constelação e uma complexidade de decodificação previamente estabelecidas. O capítulo 5 trata do problema da caracterização espectral de sinais codimodulados. O capítulo 6 trata da obtenção de códigos ótimos para algumas constelações específicas. O capítulo 7 trata do desenvolvimento de um modelo de simulação para sistemas com modulação codificada, e se fundamenta nos modelos desenvolvidos nos capítulos 2, 3 e 4. Aqui serão apresentados resultados para alguns dos códigos obtidos no capítulo 6. Finalmente, no capítulo 8 são apresentadas algumas conclusões e comentários relativos ao trabalho.

CAPÍTULO 2

ESTRUTURA E MODELO PARA O CODIFICADOR DE UM CÓDIGO CONVOLUCIONAL

2.1 - INTRODUÇÃO

Neste capítulo serão estruturados os códigos convolucionais e seus codificadores de modo adequado ao desenvolvimento deste trabalho de dissertação. Nesse sentido, será abandonado o tratamento matemático fundamentado na álgebra de polinômios, geralmente usada na caracterização dos códigos convolucionais (Forney, 1970; Peterson e Weldon, 1972). O codificador será modelado como uma máquina sequencial com um conjunto finito de estados, segundo o modelo de Mealy (Kohavi, 1970) cujas funções transição de estado e saída serão especificadas de acordo com a aplicação em que esse modelo será utilizado. Também será desenvolvido um modelo matemático simples que permitirá um acoplamento direto entre a caracterização de um código e seu codificador e vice versa. A flexibilidade desse modelo será melhor avaliada através de exemplos apresentados no final do capítulo. Em resumo, o principal objetivo deste capítulo consiste em estabelecer a fundamentação matemática a ser utilizada nos capítulos subsequentes.

2.2 - Caracterização de um Código Convolucional

Considere um sinal digital com símbolos pertencentes a um alfabeto q -ário a ser codificado usando-se um código convolucional de taxa k/n e restrição de memória m (Constraintlength em Peterson e Weldon, 1972). Considere ainda que os símbolos do sinal codificado pertencem a um alfabeto p -ário. Com o fim de assegurar a possibilidade de se enviar símbolos de informação sem qualquer processamento, como no caso extremo de códigos sistemáticos, assume-se que o alfabeto q -ário seja um subconjunto do alfabeto p -ário, e mais, que sobre eles possam ser definidos corpos de Galois de q e p elementos, $CG(q)$ e $CG(p)$, respectivamente (Peterson e Weldon, 1972). Após estas considerações, caracteriza-se um código convolucional, como usualmente, através de sua matriz geradora G

$$G = [G_0, G_1, \dots, G_{m-1}], \quad (2.1)$$

onde G_j , ($j=0,1,\dots,m-1$), são submatrizes de dimensões $k \times n$ com elementos pertencentes a $CG(p)$.

2.3 - Modelagem do Codificador

O codificador associado ao código definido por (2.1) é suposto paralelo e sem realimentação, cujo esquema genérico está mostrado na Fig. 2.1. A cada intervalo de tempo r , um vetor x^r tem acesso às entradas do codificador gerando um vetor de saída z^r , enquanto ocorre uma transição do vetor de estados atual y^r para o o vetor dos próximos estados, y^{r+1} . Abandonando por conveniência os índices de tempo, esses vetores são definidos como vetores coluna da seguinte forma:

$$\begin{aligned} x &= (x_1, x_2, \dots, x_k)^t \\ y &= (y_1, y_2, \dots, y_\mu)^t \\ z &= (z_1, z_2, \dots, z_n)^t, \end{aligned} \quad (2.2)$$

onde as componentes de x e y são q -árias, as de z são p -árias, μ é o número de células de memória q -árias do codificador e t significa transposição. Para os vetores em (2.2) convencionou-se que a significância das componentes decresce da esquerda para a direita. Desse modo, procura-se estabelecer uma estrutura de ordem para o codificador de tal modo que as entradas conectadas diretamente às saídas correspondem às componentes mais significativas dos vetores x e z , respectivamente, e estas não contribuem à formação do estado do codificador. Também, com essa convenção, se estabelece uma forma de caracterização do processo de codificação como um processamento de números inteiros correspondentes à representação decimal de x , y e z ,

$$\begin{aligned} (x)_{10} &= \sum_{i=1}^k x_i q^{k-i} \\ (y)_{10} &= \sum_{i=1}^{\mu} y_i q^{\mu-i} \\ (z)_{10} &= \sum_{i=1}^n z_i p^{n-i}, \end{aligned} \quad (2.3)$$

notação que será usada ao longo deste trabalho.

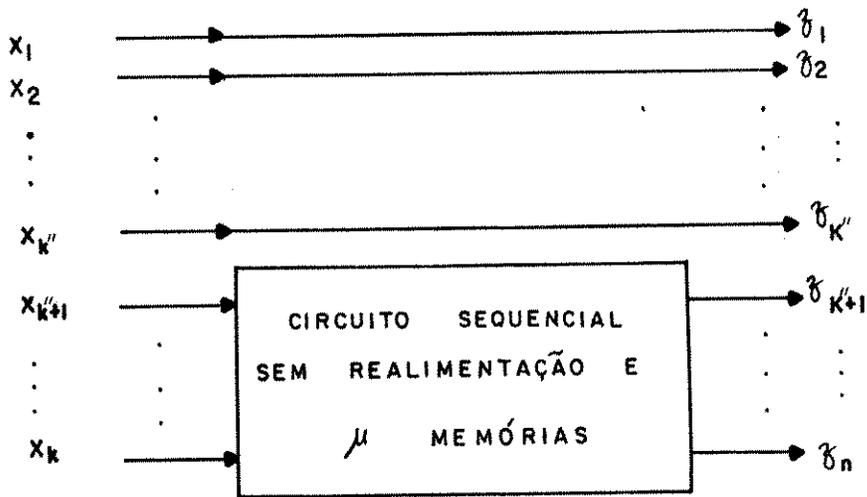


Fig. 2.1: Esquema genérico para o codificador de um código convolucional de taxa k/n , restrição de memória m e memórias q -árias, $k' \leq \mu \leq k'(m-1)$, sendo $k' = k - k''$.

2.3.1 - Caracterização do vetor de estado

A inexistência de realimentação no codificador assegura que o vetor de estados y depende somente das k' componentes menos significativas de x . Para uma mesma seqüência de entradas $\{x^r\}$, a correspondente seqüência de vetores de estados $\{y^r\}$ depende de dois aspectos. O primeiro é o modo como as μ memórias estão dispostas ao longo dos caminhos estabelecidos pelas k' entradas menos significativas do vetor x . O segundo é o modo como é definido o vetor de estados. Note que, de acordo com a Fig. 2.1, somente as $k'' = k - k'$ entradas mais significativas estão conectadas diretamente à saída. Assim, o número de memórias deve satisfazer à relação

$$k' \leq \mu \leq k'(m-1). \quad (2.4)$$

A distribuição das memórias do codificador obedece a um arranjo cuja disposição consiste de $m-1$ bancos de memórias conectados em cascata. Essa distribuição é ilustrada na Fig. 2.2 para o caso em que $\mu = k'(m-1)$.

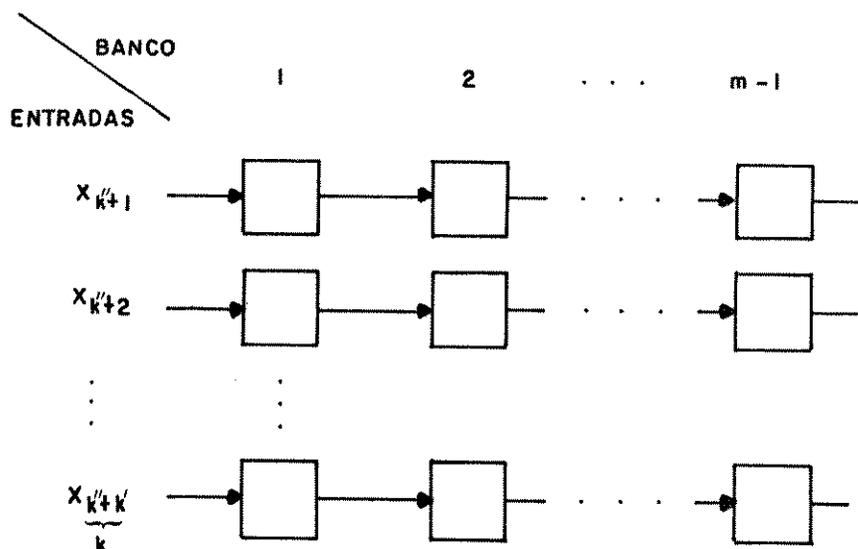


Fig. 2.2: Distribuição de memórias no circuito sequencial esquematizado na Fig. 2.1 para o caso em que $\mu = k'(m-1)$.

Sejam $s(j)$ o número de memórias presentes no j -ésimo banco de memórias e y^r_j um vetor coluna cujos componentes representam o conteúdo dessas memórias no r -ésimo intervalo de tempo. Isto é,

$$y^r_j = (y^{r_{a(j)+1}}, y^{r_{a(j)+2}}, \dots, y^{r_{a(j)+s(j)}})^t \quad (2.5)$$

onde $a(j)=0$ para $j=1$ e $a(j) = a(j-1) + s(j-1)$ para $j = 2, 3, \dots, m-1$.

O conteúdo (estado) de cada memória do banco j pode influenciar apenas o próximo estado de uma memória no banco $j+1$ ou apenas a saída do codificador ou ambos simultaneamente. Nesse sentido, é conveniente classificar as memórias do codificador em dois tipos distintos:

Memória "tipo 1" - tem influência apenas na saída ou no próximo estado e saída simultaneamente.

Memória "tipo 2" - tem influência apenas no próximo estado de uma outra memória.

Feita a distinção entre esses dois tipos de memórias, a estrutura de memórias do codificador é melhor caracterizada através da matriz ternária

$$M = [m_{ij}], \quad (i=1, 2, \dots, k'; j=1, 2, \dots, m-1), \quad (2.6)$$

onde $m_{i,j} = 1$ se a memória associada à entrada $x_{i,u}$, $u = k^n + i$, e pertencente ao j -ésimo banco é do "tipo 1", $m_{i,j} = -1$ caso a memória seja do "tipo 2" e $m_{i,j} = 0$ nas posições de cada banco em que não haja a presença de memórias.

Os elementos da matriz M apresentam as seguintes propriedades:

- 1) Em cada linha, os elementos que antecedem a um elemento de valor 1 são todos de valores ± 1 ;
- 2) Ao primeiro elemento de valor 0 (zero) de uma linha seguem-se apenas elementos de valor 0 (zero);

São verdadeiras as relações

$$\left\{ \begin{array}{l} s(j) = \sum_{i=1}^{k'} |m_{i,j}|, \\ \sum_{j=1}^{m-1} s(j) = \mu \end{array} \right. \quad (2.7a)$$

$$\left\{ \begin{array}{l} \sum_{j=1}^{m-1} s(j) = \mu \end{array} \right. \quad (2.7b)$$

$$k' = s(1) > s(2) > \dots > s(m-1);$$

A verificação dessas propriedades é imediata tendo em vista os tipos de memórias definidos e se for acrescentada a restrição inerente aos codificadores sem realimentação, nos quais só é possível a conexão entre memórias situadas ao longo de uma mesma entrada $x_{i,u}$ ($u = k^n + 1, \dots, k^n + k' - 1, k$).

O vetor definido em (2.5) corresponde ao estado do j -ésimo banco de memórias e, em consequência, o estado do codificador pode ser definido por

$$y^n = [(y_{11}^n)^k, (y_{22}^n)^k, \dots, (y_{m-1, m-1}^n)^k]^k \quad (2.8a)$$

cujos componentes, por construção, estão ordenados de tal forma que suas significâncias decrescem num mesmo banco segundo a ordem crescente dos índices das entradas e, ao longo dos bancos, no sentido da entrada para a saída do codificador. Similarmente, definindo-se $Y_j = y_j^{n+k}$ como o próximo estado associado ao j -ésimo banco de memórias, o próximo estado é expresso por

$$Y = [(Y_{11})^k, (Y_{22})^k, \dots, (Y_{m-1, m-1})^k]^k \quad (2.8b)$$

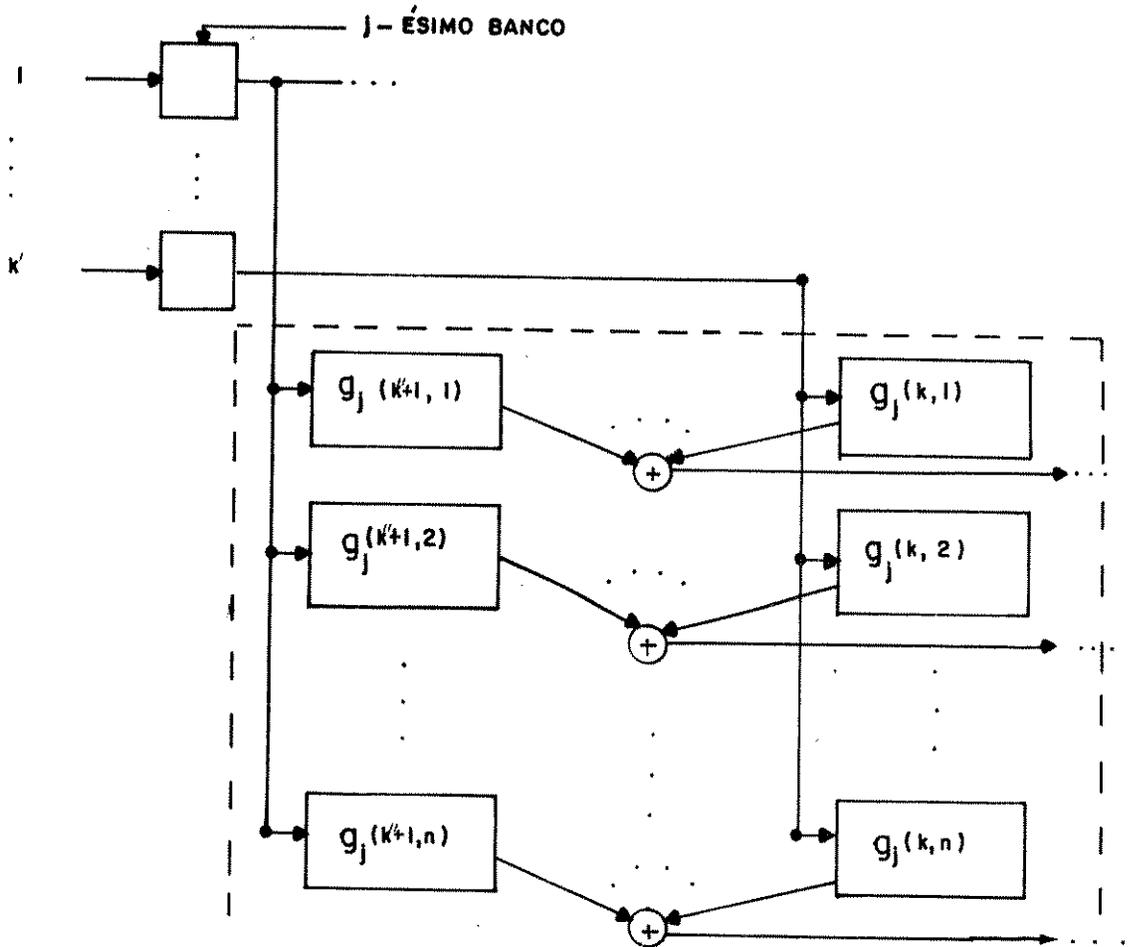


Fig. 2.3: Conexões entre as memórias do j-ésimo banco de memórias e as saídas menos significativas do codificador.

2.3.2 - Caracterização da interconexão entre memórias

A conexão das memórias às saídas é estabelecida através das submatrizes G_j ($j=1,2,\dots,m-1$) na Eq. (2.1). A Fig. 2.3 mostra as conexões correspondentes ao j-ésimo banco de memórias e às saídas menos significativas do codificador.

Através da Fig. 2.3 pode-se observar que, se $m_{i,j}=0$ ou $m_{i,j}=-1$, então a linha de índice $k''+1$ na submatriz G_j deve ser nula. Evidentemente, todas as linhas de G_j cujos índices estejam compreendidos entre 1 e k'' , inclusive, são nulas. Assim, cada submatriz G_j apresenta a forma genérica

$$G_j = \left[\begin{array}{c|c} 0_{k'' \times k''} & 0_{k'' \times n} \\ \hline 0_{k'' \times k''} & G^j_{k'' \times n} \end{array} \right], \quad j=1,2, \dots, m-1, \quad (2.9)$$

onde $O_{i,j}$ são submatrizes nulas de dimensões $i \times j$ e $G^{j,k,n}$ é uma submatriz de dimensões $k' \times n'$, $n'=n-k''$, cujas linhas nulas são conhecidas a priori através de (2.6). Quanto à submatriz G_ϕ , sua forma genérica é dada por

$$G_\phi = \left[\begin{array}{c|c} U_{k''} & \\ \hline & G^{k',n'} \end{array} \right] \quad (2.10a)$$

onde $U_{k''}$ é uma matriz identidade de ordem k'' , $O_{k',k''}$ como definido em (2.9) e $G^{k',n'}$ de dimensões $k' \times n'$. No caso particular em que as k'' entradas mais significativas não afetam as n' saídas menos significativas, (2.10a) se reduz a

$$G_\phi = \left[\begin{array}{c|c} U_{k''} & O_{k'',n'} \\ \hline O_{k',k''} & G^{k',n'} \end{array} \right] . \quad (2.10b)$$

Como última observação, note que, conhecida a matriz geradora do código (Eq. 2.1), por inspeção da mesma se obtém (2.6). Por outro lado, no problema de busca de códigos cujo codificador apresenta uma estrutura de memórias especificada previamente por meio de (2.6), todas as linhas nulas das submatrizes G_j ($j=1, 2, \dots, m-1$) são conhecidas a priori, sendo desconhecidas apenas as linhas correspondentes aos $m_{i,j}=1$.

2.3.4 - O codificador modelado como uma máquina de Mealy.

A característica de uma máquina de Mealy é que a cada instante a saída é função do estado e da entrada (Kohavi, 1970). Não é difícil mostrar, a partir da estrutura de conexões mostrada na Fig. 2.3, que a função transição de estado e a função de saída associadas ao codificador podem ser expressas, respectivamente, como

$$y^{r+1} = A \cdot y^r + B \cdot x^r, \quad \text{sobre } CG(q), \quad (2.11a)$$

$$z^r = C \cdot y^r + D \cdot x^r, \quad \text{sobre } CG(p), \quad (2.11b)$$

onde A , B , C e D são as matrizes constituintes do codificador cujas dimensões são $\mu \times \mu$, $\mu \times k$, $n \times \mu$ e $n \times k$, respectivamente. As matrizes A e B são binárias e dependem somente da estrutura de memórias do codificador, enquanto que C e D são p-árias e dependem apenas da matriz geradora, como especificado a

A seguir apresenta-se este modelo.

Seja X uma matriz cujas linhas são as possíveis entradas $x_{i,u}$ pertinentes ao processo de codificação, $u=1,2,\dots,K$, onde $K=q^\mu$ é o número de vetores de k componentes q -árias. Por conveniência, assume-se que as linhas $x_{i,u}$ sejam organizadas segundo a ordem crescente de suas representações decimais. Isto é,

$$X = [x_{i,u}], \quad (u = 1,2,\dots,K), \quad (2.14)$$

onde $x_{i,u} = (x_{i,u,1}, x_{i,u,2}, \dots, x_{i,u,k})$ e $(x_{i,u})_{i \in \mathbb{N}} = u-1$.

Num codificador com μ memórias q -árias, existe um total de $L=q^\mu$ vetores de estado y_i com representação decimal $(y_i)_{i \in \mathbb{N}} = i-1$, $i=1,2,\dots,L$.

Seja $x_{i,u}$ a entrada do codificador e y_i seu estado atual. Substituindo-se em (2.11a) x^r por $(x_{i,u})^u$ e y^r por y_i , obtém-se como próximo estado $y^{r+1} = y_j$ para algum j satisfazendo à condição $1 \leq j \leq L$. O vetor linha $e_i(u)$ com componentes $e_{i,j} = 1$ se a entrada $x_{i,u}$ possibilita a transição de estado $y_i \rightarrow y_j$ e $e_{i,j} = 0$ em caso contrário, possibilita a definição da matriz binária

$$E_{i,u} = [e_{i,j}(u)], \quad i=1,2,\dots,L, \quad (2.15)$$

de dimensões $L \times L$ que caracteriza todas as transições de estado permitidas para a entrada $x_{i,u}$. Assim, a matriz

$$E = \begin{bmatrix} E_{1,u} \\ E_{2,u} \\ \vdots \\ E_{L,u} \end{bmatrix} \quad (2.16)$$

caracteriza completamente a função transição de estado do codificador.

Agora, repetindo-se a substituição feita na Eq. 2.11a para a Eq. 2.11b, obtém-se a correspondente saída z^r definida aqui como um vetor linha $z_i(u) = (z_{i,1}, z_{i,2}, \dots, z_{i,n})$. Assim, a matriz p -ária

$$A_{i,u} = [z_i(u)], \quad i=1,2,\dots,L, \quad (2.17)$$

de dimensões $L \times n$ caracteriza todas as saídas geradas pela entrada $x_{i,u}$ quando o codificador passa por todos os estados possíveis. Portanto,

$$A = \begin{bmatrix} A_{1u} \\ A_{2u} \\ \vdots \\ A_{Lu} \end{bmatrix} \quad (2.18)$$

caracteriza completamente a função de saída do codificador.

Uma forma mais conveniente para representar (2.18) é definir

$$A_{i,u} = [a(i,u)] \quad (2.19)$$

onde $a(i,u)$ é a representação decimal de $z_i(u)$. Desse modo, cada linha i em (2.19) corresponde às saídas do codificador quando o estado é y , e as entradas são $0, 1, \dots, K-1$, nesta ordem.

2.4 - Flexibilidade do modelo

A flexibilidade do modelo é melhor analisada através de exemplos.

Exemplo 2.1 : Seja o código convolucional de parâmetros $k=2$, $n=3$, $m=3$, $q=p=2$, $k''=0$ e matriz geradora

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

Deseja-se determinar as funções transição de estado e de saída usando o modelo de Mealy e a representação matricial.

Por inspeção da matriz G , através das submatrizes G_1 e G_2 , conclui-se que a estrutura de memórias do codificador é especificada através de

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

Usando-se a estrutura de estados especificada pelas Eq. 2.7 e 2.8 e as conexões mostradas na Fig. 2.3, se obtém o esquema do codificador mostrado na Fig. 2.4, a partir do qual a função transição de estado é determinada e expressa por

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

A função de saída tanto pode ser obtida a partir da Fig. 2.4 quanto a partir das Eq. 2.11a e 2.11b. Observe que na matriz M o elemento $m_{1,2}=0$. Assim, para a obtenção de C, a primeira linha da submatriz G_{22} deve ser suprimida. A função de saída é expressa por

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

A representação matricial para as funções do codificador pode ser obtida através do algoritmo apresentado no apêndice 2B. As matrizes que caracterizam a função transição de estado são:

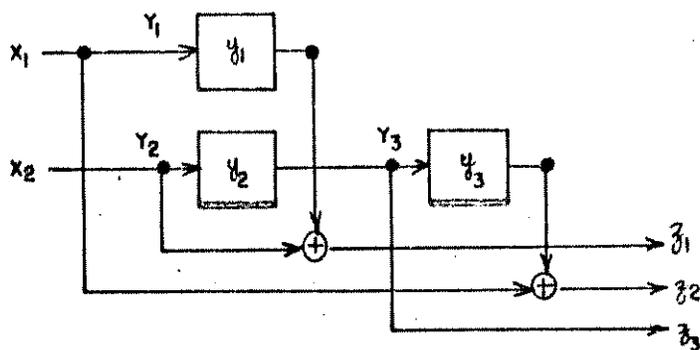


Fig. 2.4: Codificador para o código do Ex. 2.1.

$$E_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$E_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$E_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(2.20a)

As matrizes correspondentes à função de saída são:

$$A_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad A_{21} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad A_{22} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

(2.20b)

Utilizando a representação caracterizada por (2.19),

$$A_{2,1,22} = \begin{bmatrix} 0 & 4 & 2 & 6 \\ 2 & 6 & 0 & 4 \\ 1 & 5 & 3 & 7 \\ 3 & 7 & 1 & 5 \\ 4 & 0 & 6 & 2 \\ 6 & 2 & 4 & 0 \\ 5 & 1 & 7 & 3 \\ 7 & 3 & 5 & 1 \end{bmatrix}$$

(2.20c)

Exemplo 2.2 : Considere um codificador com estrutura de memórias especificada pela matriz M do Ex. 2.1 e cuja função de saída é caracterizada através das matrizes

$$C = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

Deseja-se determinar a matriz geradora do código e o respectivo codificador.

Por inspeção de M , C e D conclui-se que os parâmetros do código são: $k=3$, $n=4$, $m=3$, $q=p=2$, $k''=1$ e $\mu=3$. Isto é, o codificador tem uma entrada conectada diretamente à saída e a matriz geradora é da forma

$$G = [G_0, G_1, G_2].$$

De acordo com a Eq. 2.12b,

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

Como $k''=1$, as primeiras linhas de G_1 e G_2 são nulas como também o é a segunda linha de G_2 . Já que $m_{1,2}=0$ (ver Eq. 2.9). Assim,

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} .$$

O codificador correspondente está mostrado na Fig. 2.5 abaixo.

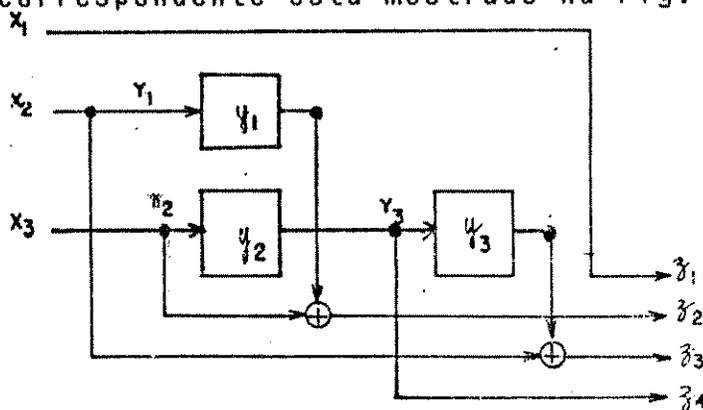


Fig. 2.5: Codificador para o código do Ex. 2.2

No capítulo 4 será tratado o problema da otimização de códigos convolucionais associados a códigos de treliça. O procedimento de busca dos códigos será baseado nas matrizes C e D e na estrutura de memórias do codificador. O Ex. 2.2 ilustra o procedimento de obtenção da matriz geradora do código.

2.5 - Complexidade computacional do modelo

A grande dificuldade da implementação do modelo desenvolvido nas seções antecedentes, diz respeito às necessidades de armazenamento, sendo que este aspecto é mais crítico quando se utiliza o modelo matricial da subseção 2.3.5, onde as dimensões das matrizes envolvidas dependem do número de estados L , que cresce exponencialmente com o número de memórias do codificador. Felizmente, nas aplicações onde o modelo matricial é utilizado (Bezerra e Arantes, 1986; capítulo 5 desta dissertação), as expressões envolvidas dependem de forma indireta de (2.16) e (2.17), o que reduz de forma considerável a necessidade de armazenamento (um fator de K). Note também que o armazenamento das matrizes constituintes A, B, C e D em (2.11), é suficiente para a geração, quando necessário, de (2.15) e/ou (2.17). Entretanto, como A e B dependem fundamentalmente de M em (2.6), é suficiente o armazenamento dessa última. No apêndice 2A é apresentado um algoritmo que implementa a função transição de estado a partir de M, sem depender das operações matriciais envolvidas em (2.11a). No apêndice 2B é apresentado um algoritmo para a obtenção de (2.15) e (2.17). Uma outra característica do modelo, é que todas as operações envolvidas são de natureza inteira e, pela amplitude dos valores numéricos envolvidos, todas as variáveis podem ser tratadas como inteiros simples (2 bytes por variável).

2.6 - Generalidade do modelo e suas restrições

As únicas restrições impostas ao modelo são que o codificador seja paralelo, sem realimentação e que sobre os símbolos q-ários e p-ários possam ser definidos pelo menos anéis comutativos. Como já foi dito, de acordo com Forney (1970), um codificador com realimentação é equivalente a um outro sem realimentação precedido de um embaralhador (scrambler). Entretanto, nessa equivalência é necessária a realização de operações matriciais de inversão que impõem a existência dos corpos $CG(q)$ e $CG(p)$. Assim, caso se deseje trabalhar com todo o universo de códigos convolucionais, é obrigatória a existência dos referidos corpos. No caso particular de códigos cujo codificador é sem realimentação e se os procedimentos que venham a utilizar o modelo não necessitem de efetuar inversões matriciais, é suficiente que sobre os alfabetos q-ários e p-ários sejam definidos anéis comutativos. Essa será a condição presente ao longo desta dissertação.

Finalmente, note que a caracterização da estrutura de memórias do codificador por meio de (2.6) possibilita a utilização de códigos convolucionais com matrizes geradoras esparsas, já que não é feita qualquer restrição quanto à distribuição das memórias ao longo de cada entrada que influa no estado. Entretanto, deve ser ressaltado que no caso de serem consideradas matrizes geradoras esparsas, necessariamente, em (2.6) existirão elementos $m_{i,j} = -1$ e isto, em geral, acarreta um acréscimo no número de estados do codificador quando comparado com um código de parâmetros idênticos mas com uma matriz M cujos elementos assumem apenas os valores 1 ou 0. Este é um aspecto importante que deve ser considerado na escolha de códigos convolucionais aplicados a sistemas com modulação codificada, supondo que o algoritmo de Viterbi é utilizado na decodificação.

APENDICE 2A

Algoritmo para a implementação da função transição de estado do codificador de um código convolucional, Eq. 2.11a, sem a realização das operações matriciais envolvidas.

1 - Armazenamento.

1.1 - Parâmetros do código/codificador: k, n, m, q, p, k', k'' .

1.2 - Estrutura de memórias do codificador: M (Eq. 2.6).

1.3 - Entrada do codificador: x'' .

1.4 - Estado atual do codificador: y'' (Eq. 2.8a).

1.5 - Vetor s de componentes $s(j)$ segundo as Eqs. 2.7.

1.6 - Próximo estado Y .

1.7 - Variáveis auxiliares.

1.7.1 - Índice de referência $a(j)$ para as componentes do estado y_j na Eq. 2.8a: LJ .

1.7.2 - Idem para o próximo estado Y_j

1.7.3 - Índices de linha e coluna para M : i, j, l .

1.7.4 - Variáveis de controle: $IA, l1$.

2 - Inicialização.

2.1 - $AJ \leftarrow \mu - s(m-1)$

2.2 - $J \leftarrow m-1$

2.3 - $LJ \leftarrow AJ$

2.4 - $IJ \leftarrow AJ$

3 - Recursão.

3.1 - Determinar, ao longo da J -ésima coluna de M , a i -ésima linha tal que $|m(i, j)|=1$ e estabelecer o correspondente índice para a componente do próximo estado: $IJ \leftarrow IJ + 1$;

- 3.2 - Para a linha i do passo anterior fazer $LJ \leftarrow LJ - s(j-1)$;
- 3.3 - Enquanto $j > 1$ fazer $LJ \leftarrow LJ + m(i1, j-1)$ e $Y_j(ij) \leftarrow y_1(LJ)$.
- 3.4 - Fazer $LJ \leftarrow AJ$, $i \leftarrow i + 1$ e repetir os passos 3.1 a 3.4 até que $i > k'$.
- 3.5 - Fazer $j \leftarrow j - 1$, $Aj \leftarrow Aj - s(j)$, $i j \leftarrow Aj$, $Lj \leftarrow Aj$ e repetir os passos 3.1 a 3.5 até que $j=2$, inclusive.
- 3.6 - Para $i=1,2, \dots, k'$ fazer $Y_1(i) \leftarrow x(i+k'')$.

APENDICE 2B

Algoritmo para a obtenção de $E_{i,j}$ e $A_{i,j}$ segundo as Eqs. 2.15 e 2.17.

1 - Armazenamento.

- 1.1 - Matrizes $A_{i,j}$ e $E_{i,j}$.
- 1.2 - Entrada $x_{i,j}$.
- 1.3 - Estado atual y_i .
- 1.4 - Próximo estado Y_j .
- 1.5 - Matrizes C e D conforme Eq. 2.11.
- 1.6 - Parâmetros do código/codificador: k , n , m , q , p , k' , k'' .
- 1.7 - Parâmetros auxiliares.
- 1.7.1 - Índices de linha e coluna: i , j .
- 1.7.2 - Número de estados do codificador: L .

2 - Inicialização:

- 2.1 - Fazer $e_{i,j}(i,j) \leftarrow 0$ para $i, j = 1, 2, \dots, L$.
- 2.2 - Fazer $i \leftarrow 1$.

3 - Recorrência para se obter $E_{i,j}$ e $A_{i,j}$.

- 3.1 - Gerar o vetor y_i tal que $(y_i)_{1 \leq j \leq L} = i-1$.
- 3.2 - Aplicar o algoritmo do apêndice 2A a $x_{i,j}$ e y_i para obter o próximo estado Y_j .
- 3.3 - Fazer $j \leftarrow (Y_j)_{1 \leq j \leq L} + 1$ e assim $e_{i,j} \leftarrow 1$.
- 3.4 - Efetuar as operações envolvidas na Eq. 2.11b, obtendo-se a i -ésima linha de $A_{i,j}$.
- 3.5 - Fazer $i \leftarrow i + 1$ e repetir os passos 3.1 a 3.5 até que $i > L$.

CAPÍTULO 3

ESTRUTURA E MODELO PARA UM CÓDIGO DE TRELIÇA ASSOCIADO A UM CÓDIGO CONVOLUCIONAL

3.1 - Introdução

O princípio básico para a transmissão de um sinal digital em sistemas com modulação codificada, consiste na transmissão dos símbolos de uma constelação com N pontos, obedecendo a uma seqüência de estados estabelecida de acordo com a estrutura da treliça associada às transições de estado do codificador. Os símbolos dessa constelação são transmitidos com a mesma duração T que seria usada para transmitir os símbolos de uma outra constelação de um sistema de referência com K pontos ($K < N$). Desse modo, é introduzida uma redundância entre os símbolos do sinal codimodulado a qual pode assegurar uma maior eficiência no uso da potência transmitida, quando comparada com o sistema de referência. Dependendo do esquema de codificação utilizado, o ganho de potência pode vir associado a uma conformação espectral (Anderson e outros, 1981; Aulin, Sundberg e Rydbeck, 1981), ou à conservação da faixa de transmissão (Ungerboeck, 1982). Em ambos os casos, é a forma como é definida e utilizada a treliça que assegura o resultado desejado.

Neste capítulo será formulado um modelo para a caracterização de um tipo particular de código, apresentado por Ungerboeck (1982), onde as seqüências de símbolos da modulação são geradas de acordo com a estrutura da treliça associada a um código convolucional binário com parâmetros compatíveis com a constelação de referência, com a constelação do código e com a complexidade (número de estados) especificada para o sistema. Mesmo sendo um caso particular na classe dos chamados códigos de treliça, ao longo deste trabalho, os códigos do tipo Ungerboeck serão denominados de códigos de treliça. O modelo a ser formulado tem como base a estrutura desenvolvida no capítulo 2 para os códigos convolucionais e será usado no procedimento de busca de códigos ótimos, a ser apresentado no capítulo 4.

3.2 - Mapeamento entre palavras-código e constelação

O primeiro passo importante na construção de um código de treliça consiste em se estabelecer o mapeamento entre o conjunto das palavras-código ou sub-blocos do código convolucional e os N símbolos da constelação usada pelo modulador. Nos esquemas de modulação digital convencionais, esse mapeamento obedece a um código de Gray que, em casos particulares como o 4PSK, assegura um casamento ótimo entre distâncias de Hamming (DH) e distâncias Euclidianas (DE). Calderbank e Mazo (1984), propuseram uma des-

crição para códigos de treliça onde a técnica de construção de códigos proposta por Ungerboeck é realizada em apenas uma etapa. O mapeamento a ser considerado ao longo deste trabalho é denominado de "mapeamento por partição de conjunto" (MPC), proposto inicialmente por Ungerboeck (1982) e apresentado de forma mais estruturada por Forney e outros (1984). Antes de maiores considerações sobre o MPC, é conveniente que se caracterize o código convolucional a ser usado em conjunção com a constelação código.

Para um código de treliça com constelação de referência e constelação código com K e N símbolos, respectivamente, o código convolucional a ser utilizado deve ser capaz de mapear K blocos de informação em N palavras-código. Se o sinal a ser codificado é q -ário e o código convolucional é p -ário, a taxa do código deve ser k/n , $k = \log_q K$ e $n = \log_p N$.

Estabelecida a taxa do código convolucional, restam as especificações da restrição de memória, m , e a estrutura da treliça segundo a qual serão transmitidos os símbolos da constelação código. Essa estrutura depende de três fatores. O primeiro é o número de estados do codificador, L ; o segundo é o modo como as $\mu = \log_q L$ memórias estão distribuídas, e o terceiro é o número de ramos paralelos, $N_p = q^{k'}$, onde k' é o número de entradas do codificador conectadas diretamente à saída (veja Cap. 2). A restrição de memória depende do número de memórias μ e do número de entradas que influenciam o estado do codificador, $k' = k - k''$, satisfazendo à relação

$$\lceil \mu/k' \rceil + 1 \leq m \leq \mu - k' + 2 \quad (3.1)$$

onde $\lceil * \rceil$ significa o menor inteiro que iguala ou excede $*$. O lado esquerdo de (3.1) corresponde à situação em que, exceto o último banco do codificador, todos os outros apresentam k' memórias. Por outro lado, o lado direito de (3.1) corresponde à situação em que ao longo de uma das k' entradas existem $\mu - k' + 1$ memórias. Isto é, o primeiro banco possui k' memórias enquanto os outros apenas uma, mas alinhadas ao longo de uma mesma entrada.

Especificados os parâmetros k , n , m e μ , resta estabelecer a estrutura da treliça a ser utilizada, a qual depende apenas da configuração de memórias do codificador, matriz M , repetida abaixo por conveniência (Eq. 3.2).

$$M = [m_{i,j}], \quad i=1,2,\dots,k' ; \quad j=1,2,\dots,m-1. \quad (3.2)$$

Note entretanto, que a estrutura do vetor de estado estabelecida em (2.8a) garante que permutações em (3.2) dos elementos $m_{i,j} = -1$ e $m_{i,j} = 1$ não dão origem a estruturas de treliças distintas. Assim, somente as configurações obtidas por manipulações dos $m_{i,j} = 0$, de tal forma que sejam mantidas as propriedades de M (veja Cap. 2), darão origem a estruturas de treliças distintas.

Supondo-se que m seja escolhido com seu valor mínimo (lado esquerdo de (3.1)), o número de estruturas possíveis para as treliças é dado por

$$N_M = \binom{k'}{k' - s(m-1)} \quad (3.3)$$

onde $k' - s(m-1)$ é o número de zeros presentes na última coluna de M . Por outro lado, se m assume seu máximo valor o número de estruturas é expresso por

$$N_M = k'. \quad (3.4)$$

Para m assumindo valores intermediários em (3.1), o número de estruturas é dado por

$$N_M = N_m \cdot \prod_{j=J}^{m-1} \binom{s(j-1)}{s(j-1) - s(j)} \quad (3.5)$$

onde N_m é o número de vetores da forma $[s(1) s(2) \dots s(m-1)]$ cujas componentes satisfazem (2.7b) com $s(1) = k'$, $s(j+1) \leq s(j)$ e J é o índice da primeira componente $s(i)$ que satisfaz à condição $s(i) < k'$. Isto é, J indica a primeira coluna de M na qual existe pelo menos um $m_{i,j} = 0$ ($i=1, 2, \dots, k'$).

Escolhida uma estrutura para a treliça, a atribuição dos elementos $m_{i,j} \neq 0$ com -1 ou $+1$ permitirá a caracterização das linhas nulas ($m_{i,j} = 0$ ou $m_{i,j} = -1$) nas submatrizes G_j ($j=1, 2, \dots, m-1$). Entretanto, como o código convolucional deve ser capaz de gerar $N=q^n$ sub-blocos distintos a serem mapeados aos N símbolos da constelação código, independentemente da matriz geradora ser esparsa ou não, é suficiente que na seleção de códigos ótimos sejam consideradas apenas as matrizes M para as quais os $m_{i,j} \neq 0$ são iguais a 1. Resumindo o que foi apresentado anteriormente, é conveniente introduzir a seguinte definição.

Definição 3.1 : Cada código convolucional que possa ser usado em conjunção com uma constelação de referência com K símbolos e com uma constelação código com N símbolos, transmitidos segundo uma treliça de L estados, é especificado através da sétupla (k, n, m, L, k'', G, M) .

O uso de G e M na Def. 3.1 em princípio é redundante à medida em que G implica em M . Entretanto, o inverso não é verdade pois apenas os elementos $m_{i,j} \neq 0$ fornecem informação precisa a respeito de G (veja Cap. 2). Porém, note que no processo de busca de códigos ótimos, M possui as informações quanto às características desejadas para os códigos.

Determinados os parâmetros do código convolucional, resta estabelecer o mapeamento entre as palavras-código e os símbolos da constelação código. Seja então z_i a palavra código associada, através do MPC, ao símbolo p_i de coordenadas $(p_{i,1}, p_{i,2}, \dots, p_{i,Q})$ em relação a uma base de dimensão Q . Assumindo-se que $(z_i)_{1,0} = i-1$, os índices das linhas da matriz

$$P = [p_i], \quad i=1,2,\dots,N, \quad (3.6)$$

possuem a informação relativa ao mapeamento das palavras-código nos símbolos da constelação código segundo as regras do MPC, a serem apresentadas na próxima seção.

3.3 - O mapeamento por partição de conjunto

Os princípios do MPC foram introduzidos originalmente por Ungerboeck (1982) e posteriormente, Forney e outros (1984) apresentaram procedimentos mais sistematizados para a caracterização desse mapeamento. Aqui, todos aqueles princípios serão seguidos, exceto por um abrandamento da relação a ser observada entre as distâncias Euclidianas mínimas correspondentes a duas subpartições consecutivas. Entretanto, será introduzido um formalismo apropriado ao desenvolvimento deste trabalho.

O princípio básico do MPC é que a constelação código com N símbolos seja subdividida em constelações, cada uma com $N_i = p^{n-1}$ símbolos ($i=1,2,\dots,n$), e que são associadas aos nós de uma árvore p -ária. As N/N_i constelações obtidas na i -ésima subpartição devem ser obtidas de tal forma que a mínima DE, Δ_i , entre os símbolos de cada constelação satisfaça à relação

$$\Delta_i \geq \Delta_{i-1}, \quad i=1,2,\dots,n-1. \quad (3.7)$$

Em relação à Ungerboeck (1982), a inclusão da igualdade em (3.7) representa o abrandamento, referido anteriormente, na relação entre as DE Δ_i 's. Isto possibilitará o uso do MPC para constelações multidimensionais.

Aos ramos da árvore no i -ésimo nível são associados os símbolos p -ários $\{0, 1, \dots, p-1\}$ de acordo com uma convenção convenientemente pré-estabelecida. Ao final da n -ésima subparti-

ção, os caminhos que se originam nos nós terminais e terminam no nó inicial correspondem às N palavras código (sub-blocos) do código convolucional, cada uma associada ao símbolo correspondente ao respectivo nó terminal. A Fig. 3.1 ilustra o MPC para a constelação do esquema BPSK para o caso em que $p=2$.

Seja Z o conjunto das N palavras-código, z , geradas pelo código convolucional, mas representadas segundo seu equivalente decimal $(z)_{10}$. Isto é,

$$Z = \{ 0, 1, \dots, N-1 \} . \quad (3.8)$$

Após o MPC, os elementos de Z estarão associados aos símbolos de tal forma que (3.6) é satisfeita. As constelações da i -ésima subpartição correspondem subconjuntos disjuntos Z_j ($j=1,2,\dots,J$), $J=(N/N_1)$, que satisfazem à condição (veja Fig. 3.1):

$$Z = Z_1 \cup Z_2 \cup \dots \cup Z_J . \quad (3.9)$$

Existem no MPC duas subpartições para as quais se deve atribuir uma importância especial, a subpartição na qual as constelações possuem K símbolos e a subpartição em que as constelações possuem N_p símbolos. A importância dessas duas subpartições reside no fato de que as mesmas servem de base para o estabelecimento das regras de construção de um código de treliça, assunto a ser apresentado na seção 3.4. No caso particular dos subconjuntos associados à partição com constelações de N_p símbolos, os mesmos terão importância fundamental no desenvolvimento da estrutura do decodificador, a ser apresentada na seção 3.6, como também na avaliação do desempenho dos códigos, a ser desenvolvida no capítulo 4.

O número de constelações com $K=q^k$ símbolos é expresso por N/K , um número inteiro. Portanto, como $N=p^n$, para os códigos de treliça em estudo, p tem que ser uma potência de q . Por outro lado, entre duas partições consecutivas, o número de símbolos nas respectivas constelações difere por um fator de p . Assim, só é possível uma partição cujas constelações possuam N_p símbolos se N_p for uma potência de p . Este fato impõe que k só pode assumir valores bem particulares já que $N_p=q^k$. Assim, para que o modelo de codificador da Fig. 2.1 possa ser usado sem restrições quanto aos valores de k , é necessário que $q=p$. Nestas condições, as partições de maior interesse no MPC são as de ordem $n-k$ e $n'-k$ com K e N_p símbolos por constelação, respectivamente.

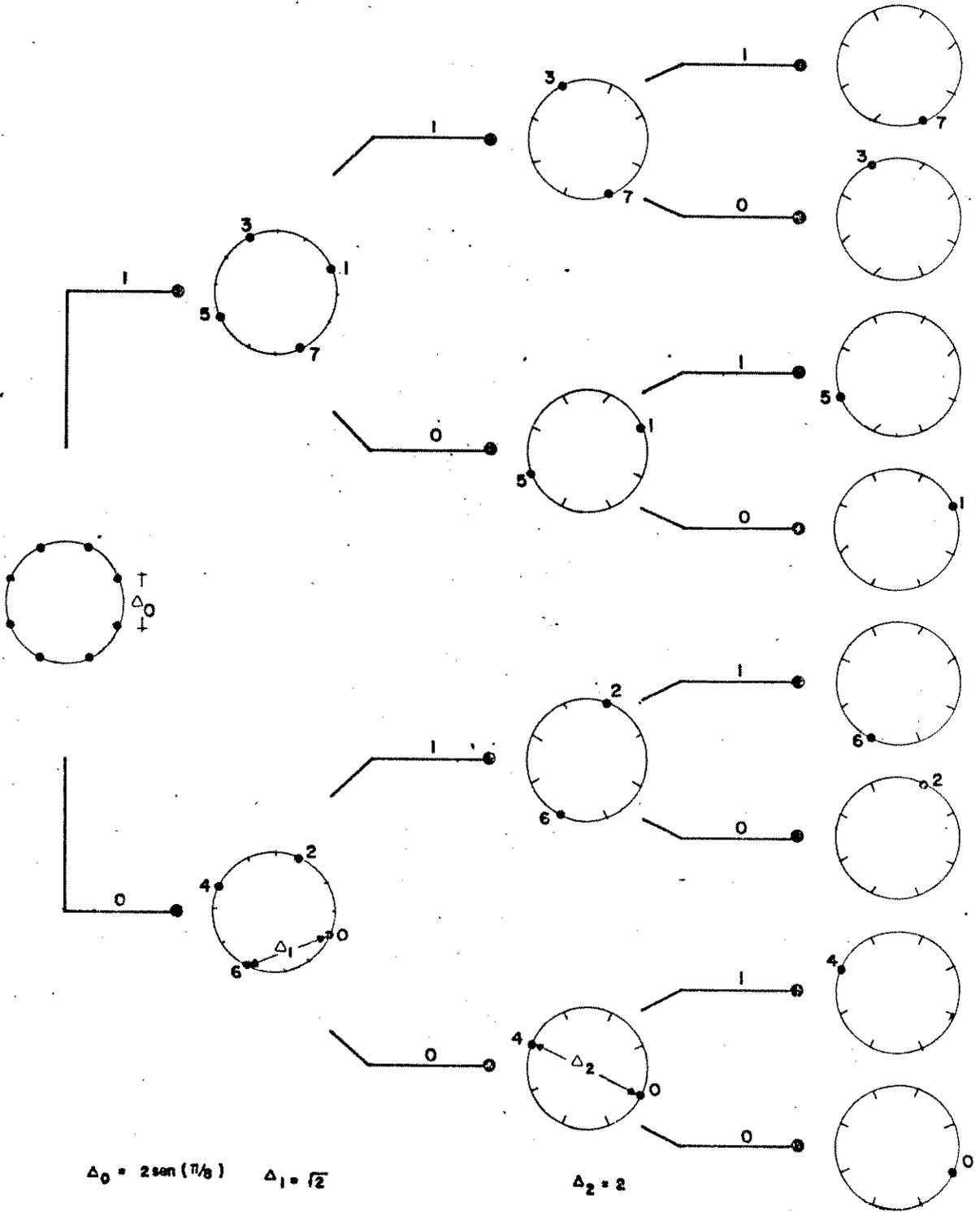


Fig. 3.1: MPC para a constelação do esquema BPSK usando-se código convolucional binário. O bit mais significativo está associado aos ramos dos nós terminais.

3.4 - Caracterização de um código de treliça

Em princípio parece que, conhecidos o código convolucional e o correspondente mapeamento na constelação código (Eq. 3.6), o respectivo código de treliça está completamente caracterizado. Esta afirmação seria verdadeira se houvesse uma correspondência biunívoca entre DH nas palavras-código e DE nos respectivos símbolos quando, então, um código convolucional ótimo implicaria num código de treliça ótimo. Assim, no intuito de se obter um código de treliça ótimo, isto é, um código com máxima distância Euclidiana livre, o respectivo código convolucional deve assegurar que as seguintes regras, baseadas em Ungerboeck (1982), sejam verdadeiras:

- R1 - Na treliça especificada através de (3.2), os ramos que se originam de, ou se destinam a, um mesmo estado devem ser mapeados com símbolos pertencentes a algum dos subconjuntos da partição de ordem $n-k$.
- R2 - Os ramos paralelos devem ser mapeados com símbolos pertencentes a algum subconjunto da partição de ordem $n'=n-k$, e de modo que a regra R1 seja satisfeita.
- R3 - O código deve ser regular no sentido de que os símbolos da constelação código estejam presentes na estrutura da treliça com a mesma frequência.
- R4 - O código de treliça deve ser não catastrófico.

A regra R1 assegura que a mínima DE entre os símbolos associados aos ramos oriundos ou que se destinam a um mesmo estado seja Δ_{n-k} , enquanto que R2 assegura uma DE mínima Δ_n entre os símbolos associados aos ramos paralelos. A regra R3 garante que, se os símbolos de fonte forem equiprováveis, os símbolos no sinal codimodulado também o serão. Esta condição também é utilizada para avaliar a margem teórica de ganho para um dado esquema de codificação, assunto que será apresentado no capítulo 6.

Quanto à regra R4, ela assegura que um número finito de erros de transmissão não resulte em um número infinito de erros de decodificação. A seguir, será desenvolvido um modelo para caracterizar um código de treliça a partir do qual as regras R1-R4 poderão ser verificadas.

A primeira etapa do modelo consiste em caracterizar a estrutura básica para a treliça associada ao código, isto é, a especificação das transições de estado. Como visto no Cap. 2, as matrizes E_u ($u=1, \dots, K$) caracterizam todas as transições de estado associadas às entradas x_u (veja Eq. 2.15). Assim, a matriz T definida por

$$T = [t(i,j)]$$

$$= \sum_{u=1}^K E_{i,u} \quad , \quad i,j=1,2,\dots,L, \quad (3.10)$$

caracteriza todas as transições de estado permitidas. Isto é, se $t(i,j)=N_r$, a transição de estado $y_i \rightarrow y_j$ é possível, enquanto que se $t(i,j)=0$ a transição não é permitida. Assim, (3.10) além de caracterizar a treliça do código, através de suas linhas e colunas, caracteriza os ramos que se originam de y_i e os que se destinam a um mesmo estado, respectivamente.

Obtida a estrutura da treliça, resta mapear cada um dos ramos com o correspondente par entrada/saída. Seja então o vetor $x_{i,j}$, definido por

$$x_{i,j} = [x(i,j,1), x(i,j,2), \dots, x(i,j,N_r)] \quad (3.11)$$

cujas componentes são a representação decimal das entradas x_u para as quais $e_{i,u}(1,j)=1$ em (2.15). Usando-se (2.11b) para cada uma dessas entradas e o estado y_i , são obtidas as correspondentes saídas especificadas através do vetor

$$z_{i,j} = [z(i,j,1), z(i,j,2), \dots, z(i,j,N_r)] \quad (3.12)$$

onde cada componente $z(i,j,l)$ é a representação decimal da saída cuja entrada é representada por $x(i,j,l)$. Assim, (3.11) e (3.12) caracterizam o mapeamento dos ramos associados à transição de estado $y_i \rightarrow y_j$. Portanto, o mapeamento total da treliça é completamente especificado através dos arranjos tridimensionais

$$X = [x_{i,j}] \quad (3.13a)$$

$$Z = [z_{i,j}] \quad , \quad i,j=1,2, \dots, L, \quad (3.13b)$$

onde $x_{i,j}$ e $z_{i,j}$ são os vetores expressos através de (3.11) e (3.12), respectivamente, para as transições de estado permitidas. Em caso contrário $x_{i,j}=z_{i,j}^*$, o asterisco sendo usado como uma marca de transição de estado não permitida.

A estrutura caracterizada através de (3.13) especifica completamente um código de treliça, desde que as regras R1-R4 sejam satisfeitas. Essa mesma estrutura possibilita a verificação das mencionadas regras para cada código convolucional possível de ser usado como um código de treliça. Essa verificação pode ser

feita da seguinte forma:

R2 - Se os componentes diferentes de x correspondentes a cada $z_{i,j}$ pertencem a um dos subconjuntos da partição de ordem $n-k$, a regra R2 está verificada.

R1 - Considere cada linha (coluna) em (3.13b). Se os componentes de cada linha (coluna) diferentes de x com a mesma terceira coordenada pertencem a um dos subconjuntos da partição de ordem $n-k$, a regra R1 está verificada.

R3 - Uma simples contagem dos elementos distintos presentes em cada $z_{i,j}$ possibilita sua verificação.

Quanto à regra R4, também é possível sua verificação através de (3.13b). Entretanto, a sua implementação depende de outras considerações que serão apresentadas a seguir.

Seja então $\{z^r(t)\}$ a seqüência codimodulada correspondente à seqüência $\{z^r\}$ gerada pelo codificador. O quadrado da distância Euclidiana (DE) entre duas seqüências codimoduladas é definido por

$$d_{EE}^2[\{z^r(t)\}_1; \{z^r(t)\}_2] = \sum_r d_{EE}^2[z^r(t)_1; z^r(t)_2] \quad (3.14)$$

onde $d_{EE}[z^r(t)_1; z^r(t)_2]$ é a DE entre os símbolos das seqüências 1 e 2 correspondentes ao r -ésimo intervalo de tempo. Se z^r_1 e z^r_2 são as respectivas palavras-código, $u=(z^r_1)_{1 \oplus 1}$ e $v=(z^r_2)_{1 \oplus 1}$, a Eq. 3.6 permite escrever

$$d_{EE}^2[z^r(t)_1; z^r(t)_2] = d_{EE}^2(p_u; p_v) \\ = \sum_{j=1}^Q [p(u,j) - p(v,j)]^2 \quad (3.15)$$

Estabelecida a distância entre duas seqüências codimoduladas, a distância livre do código é definida como

$$d_{EE}^2_{1 \neq 2} = \text{Min}_{\{z^r(t)\}_1 \neq \{z^r(t)\}_2} d_{EE}^2[\{z^r(t)\}_1; \{z^r(t)\}_2] \quad (3.16)$$

onde a minimização se estende sobre todos os possíveis pares de seqüências distintas. Esse é um problema complicado que será discutido no capítulo 4. Por enquanto, convém apenas lembrar que a cada seqüência codimodulada corresponde uma seqüência de estados $\{y^n\}$ e assim, no cálculo de (3.14) somente os intervalos de tempo em que as seqüências de estado diferem precisam ser considerados. Esses intervalos de tempo correspondem à formação dos eventos de erro, se uma delas é considerada como a seqüência enviada através do canal (Forney, 1973). Entretanto, como a cada transição de estado estão associados N_r ramos, é possível a existência de mais de uma seqüência codimodulada associada a uma mesma seqüência de estados. Assim, com o objetivo de facilitar a avaliação de (3.16), é conveniente que seja definida a seguinte matriz de distâncias

$$D_{\text{DE}} = [d(Z_i, Z_j)] , \quad i, j = 1, 2, \dots, N' \quad (3.17)$$

onde $N' = N/N_r$ e $d(Z_i, Z_j)$ é a mínima DE entre os símbolos de Z_i e os símbolos de Z_j , onde Z_i e Z_j subconjuntos da partição de ordem $n' = n - k$ no MPC.

Definição 3.2 (Viterbi e Omura, 1979): Um código convolucional é catastrófico se, e somente se, existe pelo menos uma malha fechada no diagrama de estado do codificador que, em relação à seqüência correta, possui distância de Hamming nula.

As Matrizes X e Z em (3.13) possuem todas as informações relativas ao diagrama de estado do codificador e, conseqüentemente, as malhas fechadas com os respectivos mapeamentos, podem ser identificadas a partir das mesmas. Assim, as automalhas são caracterizadas através dos elementos da diagonal principal diferentes de $*$. As malhas de duas transições podem ser identificadas através dos pares de elementos diferentes de $*$ que estejam situados em posições simétricas em relação à diagonal principal. As malhas com mais de duas transições também podem ser identificadas sem contudo existirem regras bem definidas para a caracterização das mesmas.

Proposição 3.1 : Se na diagonal principal em (3.13b), existir pelo menos um par de vetores definidos por (3.12) cujas componentes pertençam a um mesmo subconjunto da partição de ordem n' do MPC, então o código é catastrófico.

Proposição 3.2 : Se em (3.13b) existirem pelo menos dois pares de elementos, simétricos em relação à diagonal principal, e cada par, ordenadamente, é mapeado com vetores idênticos na forma de (3.12), então o código é catastrófico.

A catastroficidade para malhas fechadas com mais de duas transições é difícil de ser caracterizada. Entretanto, devido às regras R1 e R2, a catastroficidade do caso da Prop. 3.2 é provavelmente de difícil ocorrência e com mais razão para os casos de malhas com múltiplas transições. Assim, por essa argumentação e considerando a facilidade da verificação em (3.13b) das condições presentes nas proposições 3.1 e 3.2, a regra R4 poderá ser verificada e, portanto, com grande probabilidade os códigos selecionados nestas condições serão não catastróficos. A favor dessa argumentação são os estudos referenciados por Viterbi e Omura (1979) quanto à fração de códigos catastróficos presente na classe de códigos de taxa $1/n$, fração essa expressa por $1/(2^n - 1)$. Portanto, considerando a regra R4 verificada através das Prop. 3.1 e 3.2, um código de treliça está completamente especificado através da seguinte definição:

Definição 3.3 : Um código de treliça associado ao código convolucional caracterizado pela Def. 3.1 é completamente definido pelas Eqs. 3.13 e pelas regras R1-R4.

Os exemplos que se seguem têm como objetivo ilustrar o modelo desenvolvido ao longo desse capítulo na caracterização de um código de treliça.

Exemplo 3.1 : Considere o código convolucional do Ex. 2.1. De acordo com a Def. 3.1 este código é apropriado para controlar a codificação de uma constelação de $K=4$ símbolos numa outra de $N=8$ símbolos, por exemplo, 4PSK em 8PSK. Como $k=0$, usando-se o MPC mostrado na Fig. 3.1, as partições de interesse são a primeira ($n-k=1$) e a terceira ($n-k=3$). Os subconjuntos da primeira partição são

$$Z_1 = \{ 0, 2, 4, 6 \} \quad \text{e} \quad Z_3 = \{ 1, 3, 5, 7 \}, \quad (3.18a)$$

enquanto que para a terceira partição os subconjuntos são

$$Z_1 = \{0\}, \quad Z_3 = \{1\}, \quad \dots \quad Z_7 = \{7\}. \quad (3.18b)$$

Aplicando-se a Eq. 3.10 às matrizes (2.19) se obtém a matriz de transição

$$T = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (3.19a)$$

que caracteriza a estrutura da treliça associada ao código, conforme ilustrada na Fig. 3.2.

Por inspeção das Eqs. (2.19), (2.20) e (3.19a) se obtém (3.13a) e (3.13b) expressas como

$$X = \begin{bmatrix} 0 & * & 1 & * & 2 & * & 3 & * \\ 0 & * & 1 & * & 2 & * & 3 & * \\ * & 0 & * & 1 & * & 2 & * & 3 \\ * & 0 & * & 1 & * & 2 & * & 3 \\ 0 & * & 1 & * & 2 & * & 3 & * \\ 0 & * & 1 & * & 2 & * & 3 & * \\ * & 0 & * & 1 & * & 2 & * & 3 \\ * & 0 & * & 1 & * & 2 & * & 3 \end{bmatrix} \quad Z = \begin{bmatrix} 0 & * & 4 & * & 2 & * & 6 & * \\ 2 & * & 6 & * & 0 & * & 4 & * \\ * & 1 & * & 5 & * & 3 & * & 7 \\ * & 3 & * & 7 & * & 1 & * & 5 \\ 4 & * & 0 & * & 6 & * & 2 & * \\ 6 & * & 2 & * & 4 & * & 0 & * \\ * & 5 & * & 1 & * & 7 & * & 3 \\ * & 7 & * & 3 & * & 5 & * & 1 \end{bmatrix} \quad (3.19b)$$

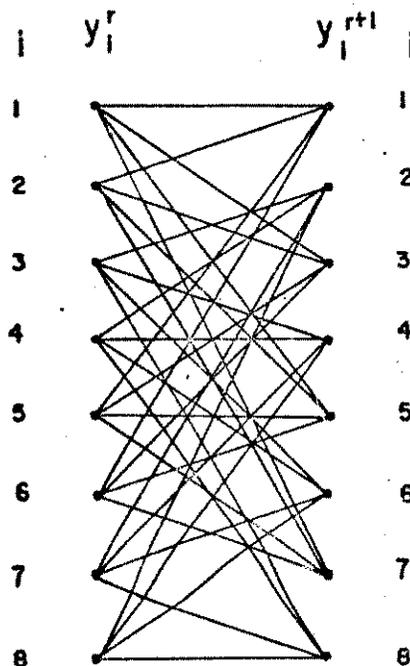


Fig. 3.2 - Estrutura básica para a treliça do codificador associado ao código do Ex. 3.1. ($i = (y_i)_{i \neq 0} + 1$).

Observe que os elementos nas linhas e colunas de Z para os quais é permitida uma transição de estado pertencem a um dos subconjuntos em (3.18a) e portanto R1 é verificada. A regra R2 não se aplica neste caso, pois $N_r=1$. A regra R3 é verificada já que cada símbolo aparece 4 vezes no mapeamento da treliça. Todos os elementos diferentes de * na diagonal de Z como também os pares dos elementos simétricos à diagonal são distintos e assim o código satisfaz à regra R4. Portanto, (3.19b) especifica um código de treliça.

Exemplo 3.2 : Considere-se o código convolucional com matriz geradora

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

Por inspeção de G se obtém os parâmetros $k=2$, $n=3$, $m=3$, $\nu=2$, $k''=1$ e o codificador mostrado na Fig. 3.3. Para este codificador não é difícil verificar que

$$T = \begin{bmatrix} 2 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 2 \end{bmatrix}$$

$$X = \begin{bmatrix} (0 \ 2) & (* \ *) & (1 \ 3) & (* \ *) \\ (0 \ 2) & (* \ *) & (1 \ 3) & (* \ *) \\ (* \ *) & (0 \ 2) & (* \ *) & (1 \ 3) \\ (* \ *) & (0 \ 2) & (* \ *) & (1 \ 3) \end{bmatrix} \quad Z = \begin{bmatrix} (0 \ 4) & (* \ *) & (2 \ 6) & (* \ *) \\ (2 \ 6) & (* \ *) & (0 \ 4) & (* \ *) \\ (* \ *) & (1 \ 5) & (* \ *) & (3 \ 7) \\ (* \ *) & (3 \ 7) & (* \ *) & (1 \ 5) \end{bmatrix}$$

e que Z satisfaz às regras R1-R4. Entretanto, se a matriz geradora fosse

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

a matriz Z resultaria em

$$Z = \begin{bmatrix} (0 \ 4) & (* \ *) & (3 \ 7) & (* \ *) \\ (1 \ 5) & (* \ *) & (2 \ 6) & (* \ *) \\ (* \ *) & (2 \ 6) & (* \ *) & (1 \ 5) \\ (* \ *) & (3 \ 7) & (* \ *) & (0 \ 4) \end{bmatrix}$$

para a qual somente a regra R3 é satisfeita. A Fig. 3.4 mostra a estrutura da treliça para um codificador com estrutura de memórias $M = [1 \ 1]$

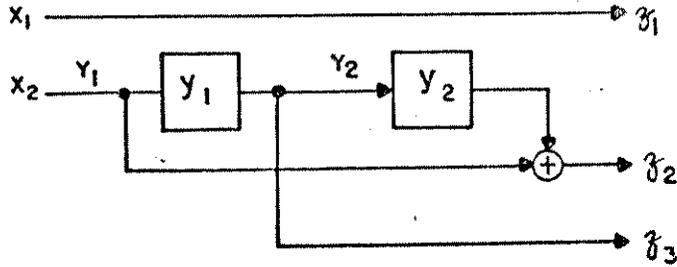


Fig. 3.3: Codificador para o código do Ex. 3.2.

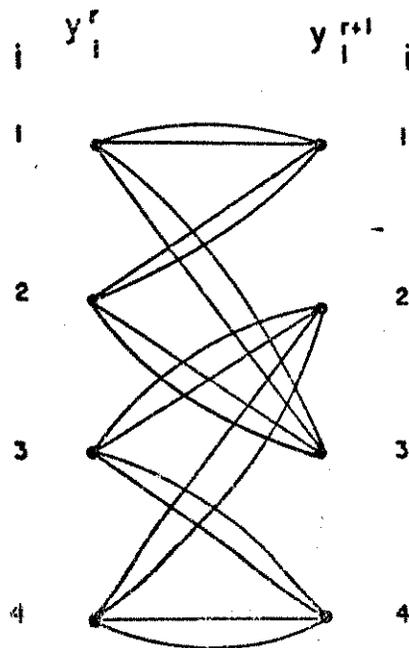


Fig. 3.4: Estrutura da treliça para o código do Ex. 3.2.

3.5 - Desempenho de um código de treliça

Uma das formas de se medir o desempenho de um código de treliça consiste em se determinar o ganho assintótico definido por

$$G_{\infty} = 20 \log(d_{1, \sqrt{E_{\infty}}}/d_{ref}) \quad (\text{dB}) \quad (3.20)$$

onde d_{ref} é a mínima DE entre os símbolos na constelação de referência, e $d_{1, \sqrt{E_{\infty}}}$ é a distância livre do código, ambas obtidas para constelações com energia média $E_M=1$ (Ungerboeck, 1982 ; Forney e outros, 1984).

O ganho assintótico em geral é usado apenas como um referencial para comparação do desempenho de códigos distintos. O desempenho real é obtido por meio do cálculo das probabilidades de evento de erro e erro de bit de informação que, infelizmente, só podem ser avaliadas através da utilização de limites superiores e inferiores ou por meio de simulação. No capítulo 4 este problema será discutido com maiores detalhes. Uma outra utilidade do uso do ganho assintótico consiste em se medir o quão próximo o ganho do código está do ganho máximo permitido pelo Teorema da Codificação de Shannon (Gallager, 1965). O problema do cálculo do máximo ganho teórico será tratado no capítulo 6.

3.6 - Estrutura do decodificador

O ganho de potência máximo proporcionado por um código de treliça pressupõe que o receptor seja um estimador de seqüências de máxima verossimilhança. Em geral, a implementação prática desse estimador se baseia no algoritmo de Viterbi (AV) (Forney, 1973). A seguir, a fundamentação teórica do AV apresentada por Forney (1973) será especializada e modelada de forma apropriada ao uso em sistemas com modulação codificada.

3.6.1 - Princípios básicos do algoritmo de Viterbi

Seja $\{y^r\}$ a seqüência de estados associada à seqüência codimodulada $\{z^r(t)\}$ que, ao ser transmitida através do canal, é corrompida por ruído e outras interferências. Esse sinal distorcido é processado pelo receptor com o fim de estimar uma seqüência $\{\hat{y}^r\}$ que minimize a probabilidade de erro em relação à seqüência verdadeira, $\{y^r\}$. Para um sistema com L estados e supondo-se que uma decisão deva ser tomada no instante $r=R$, o AV decide pela seqüência ótima a partir de um conjunto de L seqüências

$$\{ \tilde{y}^r \}_1, \{ \tilde{y}^r \}_2, \dots, \{ \tilde{y}^r \}_L \quad (3.21)$$

denominadas de sobreviventes. A decisão é tomada com base na métrica acumulada até o instante R , para cada sobrevivente, definida como

$$\lambda_i(R) = \sum_{r=0}^R \Delta\lambda_i(r), \quad i=1,2, \dots, L, \quad (3.22)$$

onde $\Delta\lambda_i(r)$ é o incremento de métrica no instante r . Desse modo, em $r=R$, ao conjunto de sobreviventes expresso por (3.21) está associado o conjunto de métricas

$$\{ \lambda_1(R), \lambda_2(R), \dots, \lambda_L(R) \}. \quad (3.23)$$

Observe que uma decisão final só é possível quando o instante R corresponde ao último estado da seqüência transmitida. Esse fato resulta em inconvenientes tais como atraso de decodificação inaceitável e necessidade de uma ilimitada capacidade de armazenamento no decodificador. Assim, em termos realistas, deve-se considerar um receptor subótimo no qual o atraso de decodificação é fixado em δ unidades de tempo, cujo valor é escolhido de forma apropriada (Viterbi e Omura, 1979). Uma escolha padrão atribuí para o atraso de decodificação $\delta = 6\mu$, onde μ é o número de memórias do codificador. Assumidas essas limitações, para $r > \delta$, cada sobrevivente passa a ser da forma

$$\{ \tilde{y}^{r-\delta}, \tilde{y}^{r-\delta+1}, \dots, \tilde{y}^r \}_i, \quad i=1,2,\dots,L \quad (3.24)$$

a partir dos quais é estimada a transição de estado

$$T_j = \tilde{y}^{r-\delta} \text{ ----> } \tilde{y}^{r-\delta+1} \quad (3.25)$$

onde j é o índice do sobrevivente com métrica mínima $\lambda_j(r)$, no instante r , pertencente ao conjunto

$$\{ \lambda_1(r), \lambda_2(r), \dots, \lambda_L(r) \} \quad (3.26)$$

onde

$$\lambda_i(r) = \sum_{s=0}^r \Delta\lambda_i(s), \quad i=1,2, \dots, L. \quad (3.27)$$

Na hipótese da existência em (3.26) de mais de um elemento igual a $\lambda_j(r)$, é necessário que se estabeleça algum critério de desempate. O critério mais imediato consiste em escolher uma dessas métricas aleatoriamente. Um critério mais elaborado, denominado de majoritário, é sugerido por Forney (1973) e se fundamenta no argumento de que, com grande probabilidade, a cada decisão todos os sobreviventes apresentam em sua cauda a transição T_j expressa em (3.25). Quando esse evento não ocorrer e há vários sobreviventes com a mesma métrica mínima, é possível o agrupamento dos sobreviventes com caudas idênticas, independentemente de suas métricas. Após esse agrupamento, se os sobreviventes com métrica mínima pertencem a um mesmo grupo, a decisão é pela transição de estado correspondente à cauda desse grupo. Em caso contrário, isto é, se os sobreviventes com métrica mínima pertencem a grupos distintos, a decisão será pela transição correspondente à cauda do grupo com maior número de sobreviventes.

O próximo passo do algoritmo consiste em estender os sobreviventes ao instante $r+1$. Isto é, determinar \tilde{y}^{r+1} para cada sobrevivente com o conseqüente descarte de $\tilde{y}^{r-\delta}$.

Sejam então y_i o estado do sobrevivente $\{\tilde{y}^r\}$, no instante r e $\Delta\lambda_{i,1}$ o mínimo incremento na métrica correspondente à transição de estado $y_i \rightarrow y_1$ ($i=1,2, \dots, L$). Se y_j é o estado tal que

$$\Delta\lambda_{i,j} = \min_l \Delta\lambda_{i,l}, \quad (3.28)$$

então, no instante $r+1$ o sobrevivente $\{\tilde{y}^r\}$, será especificado como $\{\tilde{y}^{r+1}\}_j$ com métrica

$$\lambda_j(r+1) = \lambda_i(r) + \Delta\lambda_{i,j}. \quad (3.29)$$

Isto é, no instante r

$$\{\tilde{y}^r\}_i = \{\tilde{y}^{r-\delta}, \tilde{y}^{r-\delta+1}, \dots, \tilde{y}^r = y_i\} \quad (3.30)$$

enquanto que no instante $r'=r+1$

$$\begin{aligned}
 (\bar{y}^{r+\delta})_i &= (\bar{y}^{r'})_j \\
 &= (\bar{y}^{r'-\delta}, \bar{y}^{r'-\delta+1}, \dots, \bar{y}^{r'}=y_j) .
 \end{aligned} \tag{3.31}$$

Em geral, $\Delta\lambda_{i,j} > 0$ e portanto a métrica acumulada (Eq. 3.29) é uma função não decrescente ao longo do tempo. Assim, com o fim de se evitar problemas de "overflow", é necessário uma normalização periódica da métrica dos sobreviventes, por exemplo, a cada δ unidades de tempo.

3.6.2 - O algoritmo de Viterbi para códigos de treliça

Na seção anterior foram estabelecidos os princípios básicos que norteiam o AV. Entretanto, é necessário que sejam providos meios que possibilitem sua implementação utilizando o modelo desenvolvido ao longo deste trabalho para os códigos de treliça. Alguns aspectos ainda precisam ser considerados. O primeiro deles consiste em determinar o modo como, a cada instante, é possível a obtenção dos sobreviventes já que, de acordo com (3.30) e (3.31), a seqüência de estados associada a cada sobrevivente muda no tempo de uma forma muito dinâmica e imprevisível. O segundo é a atualização da métrica dos sobreviventes e, finalmente, o aspecto mais importante que é a recuperação da informação correspondente à decisão fornecida pelo algoritmo. A seguir será desenvolvida uma formulação que permitirá a implementação desses detalhes no AV.

Considere inicialmente o intervalo de tempo de $r=0$ até $r=\delta$ quando é tomada a primeira decisão. Ao longo desse intervalo pode ser construída a matriz

$$S = [s(j,l)] , \quad j=1,2, \dots, L ; l=1,2, \dots, \delta, \tag{3.32}$$

onde $s(j,l)=i$, com $i=(y_i)_{1,\delta+1}$, e y_i é o estado que satisfaz à Eq. 3.28 no instante $r=l$. Simultaneamente à construção de S , se constrói o vetor das métricas acumuladas

$$\lambda_1 = [\lambda_1(1), \lambda_2(1), \dots, \lambda_L(1)] , \quad l=1,2, \dots, \delta \tag{3.33}$$

onde $\lambda_j(1)$ ($j=1,2, \dots, L$) é a métrica do sobrevivente que, no instante l , termina no estado j . Assim, a cada instante $r=l$, o elemento $s(j,l)$ fornece a seguinte informação: O sobrevivente que termina no estado j no instante l se encontrava no estado i no

instante $i-1$ e sua métrica atual é $\lambda_j(i)$. Portanto, em $r = \delta$ a primeira decisão pode ser tomada escolhendo-se o sobrevivente j para o qual

$$\lambda_j(\delta) = \text{Min.}(\lambda_1(\delta), \lambda_2(\delta), \dots, \lambda_L(\delta)) \quad (3.34)$$

e cuja seqüência de estados pode ser obtida através da j -ésima linha da matriz

$$S^* = [s^*(j, i)] \quad , \quad j=1, 2, \dots, L \quad ; \quad i=1, 2, \dots, \delta+1 \quad (3.35a)$$

onde

$$s^*(j, \delta-i+1) = \begin{cases} j & ; \quad i=0 \\ s[s^*(j, \delta-i+2), \delta-i+1] & ; \quad i=1, 2, \dots, \delta. \end{cases} \quad (3.35b)$$

A prova de que a j -ésima linha de S^* representa a seqüência de estados associada ao sobrevivente que termina no estado j no instante δ é como se segue:

Sejam $s^*(j, i)$ e $s^*(j, i+1)$ dois elementos consecutivos na j -ésima linha de S^* , com i escolhido de modo que em (3.35b) $i > 1$. Assim, de acordo com (3.35b),

$$s^*(j, i) = s(s^*(j, i+1), i).$$

Suponha que $s^*(j, i) = v$ e que $s(v, i) = u$, u e v inteiros pertencentes ao conjunto $\{1, 2, \dots, L\}$. Como $s(v, i) = u$ significa que o sobrevivente que termina no estado v no instante i terminava no estado u no instante $i-1$, e notando que o índice de coluna de S^* excede o índice de tempo por uma unidade, os elementos $s^*(j, i)$ e $s^*(j, i+1)$ indicam a transição de estados $u \rightarrow v$ entre os instantes $i-1$ e i . A comprovação de que esta transição pertence ao sobrevivente que termina no estado j no instante δ é imediata pois, de acordo com (3.35b), $s^*(j, \delta+1) = j$. Portanto, a construção da j -ésima linha de S^* , segundo a ordem decrescente do índice de coluna, assegura que a mesma representa a seqüência de estados associada ao sobrevivente que termina no estado j no instante δ .

C.Q.D.

Obtida a seqüência de estados associada ao j -ésimo sobrevivente, é tomada a decisão pela transição

$$T_j = (y_{u,v} \rightarrow y_{v,u}), \quad u = s^*(j,1) \text{ e } v = s^*(j,2) \quad (3.36)$$

que muito provavelmente é a mesma para $j=1,2, \dots, L$ (sobreviventes com cauda comum). No caso de empate ao se avaliar (3.34), é necessário se obter as seqüências de estados de todos os sobreviventes e se aplicar o critério majoritário com base na informação fornecida pelas duas primeiras colunas de (3.35b).

Agora, para $r > \delta$, tudo que deve ser feito é descartar a primeira coluna da matriz S em (3.32), após um deslocamento de suas colunas para a esquerda, e repetir o procedimento como na primeira decisão. Note que, no procedimento descrito, somente (3.32) e (3.33) precisam ser armazenadas permanentemente.

A informação associada à transição de estado resultante do processo de decisão pode ser obtida simplesmente como sendo o elemento de coordenadas (u,v) das matrizes X e Z em (3.13). Para códigos em que $N_r=1$, a informação decodificada é simplesmente $x_{u,v}$. Entretanto, para $N_r > 1$, existe mais de uma possibilidade de decisão e assim um procedimento preliminar deve ser implementado antes de ser tomada a decisão final. Um procedimento possível consiste em armazenar permanentemente, nos mesmos moldes do armazenamento de (3.32), as coordenadas do sinal que chega ao receptor. O símbolo em $z_{u,v}$ que estiver mais próximo do sinal recebido no instante $r-\delta$ será considerado, juntamente com o respectivo valor de $x_{u,v}$, como a decisão final do algoritmo no instante r (veja (3.11)-(3.13)).

Estabelecidos os procedimentos que viabilizam implementar a recuperação da informação, resta esclarecer a forma como é efetuado o cálculo dos incrementos de métrica que permitem estabelecer as métricas dos sobreviventes a cada instante. Assumindo-se que o ruído de canal é gaussiano branco e aditivo, é possível se verificar (Forney, 1973) que os incrementos de métrica correspondem ao quadrado da DE entre o sinal recebido no receptor e os símbolos da constelação código. Isto é, se o sinal recebido é especificado pelo vetor $r = (r_1, r_2, \dots, r_Q)$ e $v = (v_1, v_2, \dots, v_Q)$ é o símbolo associado a um dos ramos de uma transição de estado, o incremento de métrica proporcionado pelo ramo é dado por

$$\Delta\lambda = \sum_{i=1}^Q (v_i - r_i)^2 \quad (3.37)$$

Como a cada transição de estados correspondem N_r ramos, (3.37) é repetida N_r vezes, sendo retido o menor valor obtido, $\Delta\lambda_{i,1}$, que será utilizado em (3.28). A implementação desses cálculos impõe

que a matriz P expressa por (3.6) também seja armazenada permanentemente no receptor.

Finalizando este capítulo, note-se que no procedimento descrito acima para a recuperação da informação, quando $N_r > 1$, (3.37) também precisa ser avaliada N_r vezes, provocando atraso de decodificação. Esses cálculos poderão ser evitados se durante a obtenção de $\Delta\lambda_{i,j}$ é guardado de alguma forma conveniente o ramo que o originou. Assim, durante a obtenção dos sobreviventes é possível identificar o ramo associado a cada transição de estado e por conseguinte obter a informação contida em cada transição de estado estimada pelo AV. É evidente que com este último procedimento é necessário o armazenamento permanente de mais $L \times \delta$ valores.

CAPÍTULO 4

OTIMIZAÇÃO DE CÓDIGOS CONVOLUCIONAIS ASSOCIADOS A CÓDIGOS DE TRELIÇA

4.1 - Introdução

Em geral, nos procedimentos de otimização de códigos convolucionais aplicados a sistemas com modulação codificada, o objetivo final é selecionar o código ou códigos com máxima distância livre (Ungerboeck, 1982; Borelli, 1983). Neste trabalho, o procedimento de otimização será dividido em duas etapas. Na primeira fase são selecionados os códigos com máxima distância livre enquanto que na segunda, a partir desses códigos, serão selecionados os códigos que minimizam os coeficientes dos limitantes superiores para as probabilidades de erro de evento e de bit, segundo a ordem crescente das métricas dos eventos de erro e de acordo com critérios a serem estabelecidos. Neste capítulo, tendo em vista a primeira etapa de otimização, será discutida a complexidade envolvida na busca exaustiva de códigos quando não se utiliza regras de seleção. Demonstra-se que, mesmo sem o uso de regras de seleção, pode-se obter uma considerável simplificação na busca exaustiva se esta for realizada sobre o universo das matrizes C e D da função de saída do codificador (veja Cap. 2). Maior simplificação no procedimento de busca será obtida ao se utilizar algumas propriedades peculiares ao MPC, apresentadas no Cap. 3. Com base nessas simplificações, será esboçado um esquema de busca que tem como maior complexidade o cálculo da distância livre do código. No intuito de minimizar a complexidade desse problema, será discutido um procedimento que permite o cálculo da distância livre do código além de permitir, com pequenas alterações, a avaliação das métricas dos eventos de erro presentes nas expressões dos limitantes superiores das probabilidades de erro de evento e de bit.

4.2 - A complexidade da seleção exaustiva sem regras de seleção

Considere-se o problema da busca exaustiva de códigos convolucionais aplicados a códigos de treliça sobre o universo das matrizes geradoras ou das matrizes de verificação de paridade e que não sejam usadas regras de seleção estabelecidas de forma apropriada. Nestas condições, a complexidade do problema é muito grande, se não impraticável, mesmo para as características dos códigos geralmente usados em sistemas com modulação codificada. Para um código convolucional de taxa k/n e restrição de memória m , o número de possibilidades para a matriz geradora G na Eq. 2.1 é dado por

$$N_G = N^{km} \quad (4.1)$$

onde $N=q^n$ é o número de possibilidades para cada linha das submatrizes G_j ($j=0,1, \dots, m-1$). Se G for interpretada como um vetor de $n \cdot m$ componentes K -árias, $K=q^k$, (4.1) passa a ser expressa por

$$N_G = K^{nm}. \quad (4.2)$$

Evidentemente, (4.1) ou (4.2) englobam algumas possibilidades triviais que podem ser eliminadas previamente. Mesmo assim, o esforço computacional envolvido na geração das matrizes G é muito grande, mesmo para os pequenos valores de k, n, m e q que ocorrem para os códigos convolucionais aplicados a códigos de treliça.

Suponha então que a geração do universo dos códigos convolucionais seja transferida da matriz geradora para as matrizes constituintes da função de saída do codificador, matrizes C e D na Eq. 2.11b. Considerando C e D como vetores N -ários de e e k componentes, respectivamente, (4.1) e (4.2) passam a ser expressas por

$$N_G = N_C N_D \quad (4.3a)$$

onde, N_C e N_D representam, respectivamente, o número de possibilidades para C e D , sendo expressos por

$$N_C = (N-1)^e \quad (4.3b)$$

$$N_D = N^k, \quad (4.3c)$$

onde em (4.3b) se usa o fato de que a matriz C , para códigos em que G não é esparsa, não possui como colunas o vetor nulo de n componentes. Além disso, para o caso especial em que $k \neq 0$, as k primeiras linhas de C são nulas (veja Eqs. 2.9 e 2.12b), sendo portanto suficiente considerar C como um vetor de componentes M -árias, $M=q^{n-k}$. Portanto, N_C passa a ser expresso por (4.3d) abaixo, o que assegura uma significativa redução no esforço computacional.

$$N_C = (M-1)^e \quad (4.3d)$$

Apenas com o fim de se estabelecerem comparações mais concretas, considere os exemplos a seguir.

Exemplo 4.1: Seja um código de parâmetros $k=2$, $n=3$, $m=3$, $k''=0$, $q=2$ e $\mu=3$. Para esses valores a Eq. 4.1 fornece $N_{\text{C}}=262.141$, enquanto que através de (4.3b) e (4.3c) se obtém 21.952, o que corresponde a uma redução de um fator de aproximadamente 12.

Exemplo 4.2: Para um código de parâmetros $k=3$, $n=4$, $m=3$, $k''=1$, $q=2$ e $\mu=3$, (4.1) fornece um valor de aproximadamente $7.0 \text{ E}10$, enquanto que usando-se $M=8$ e (4.3d) se obtém $N_{\text{C}}=87.808$, o que representa uma redução de um fator de aproximadamente 800.000.

Os exemplos acima mostram que mesmo numa implementação de busca exaustiva baseada somente na força bruta, a busca sobre os universos das matrizes G e D pode oferecer simplificações significativas. A seguir, as regras do MPC estabelecidas no Cap. 3 serão manuseadas de forma apropriada, no sentido de se obter outras reduções no universo de busca.

4.3 - O MPC na redução do universo da busca exaustiva

Retornando às regras do MPC (Cap. 3, Sec. 3.3), tanto na partição de ordem $n-k$ quanto na partição de ordem $n-k''$, existe somente um subconjunto (ou constelação) ao qual pertence o símbolo mapeado à palavra-código $z=0$. Seja Z_1 esse subconjunto na partição de ordem $n-k$. Isto é,

$$Z_1 = \{0, a_{22}, a_{33}, \dots, a_{K}\} \quad (4.4)$$

onde os a_i 's são a representação decimal das palavras código associadas aos símbolos desse conjunto. Sejam $Z_{1,j}$, ($j=1,2, \dots, K'=K/N_{\mu}$), os subconjuntos na partição de ordem $n-k''$ resultantes da partição de Z_1 . Nominalmente,

$$Z_{1,1} = \{0, b_{22}, \dots, b_{N_{\mu}}\} ; \quad Z_{1,2} = \{c_1, c_2, \dots, c_{N_{\mu}}\} ; \quad \text{etc...} \quad (4.5)$$

Segundo as regras R1 e R3 do MPC, combinações dos elementos em (4.4) devem ser assinaladas aos ramos originários de algum estado associado ao codificador e para os quais as respectivas entradas 0, 1, $\dots, K-1$ (representação decimal) já estão assinaladas através de (3.13a). Observe que para $x''=0$ e $y''=0$, as Eqs. 2.11a e 2.11b fornecem $y''+1=0$ e $z''=0$. Portanto, conclui-se que as combinações dos elementos do conjunto em (4.4) só podem ser alocadas aos ramos que se originem do estado $y=0$ (primeira linha da matriz Z em (3.13b)), e além disso, um desses ramos tem que ser mapeado com o par entrada/saída correspondente aos res-

pectivos vetores (entrada e saída) nulos. Assim, o número de possibilidades de mapeamento desses ramos é dado por

$$(N_p - 1)!(K' - 1)!(N_p - 1) \ll (K - 1)! \quad (4.6)$$

Seja então $(0 v_{22} v_{33} v_{kk})$ a combinação de inteiros correspondente à representação decimal de uma dessas possibilidades e $(0 1 2 \dots K-1)$ a representação decimal das respectivas entradas. Para $y=0$, (2.11b) se reduz a

$$z^r = D \cdot x^r, \quad \text{sobre } CG(q). \quad (4.7)$$

Se x_i e z_i são a entrada e saída com representação decimal u_i e v_i , respectivamente, suas substituições em (4.7) resultam no sistema de equações em $CG(q)$ expresso por

$$z_i = D \cdot x_i, \quad i=2,3, \dots, K, \quad (4.8)$$

com $(K-1)n$ equações e kn incógnitas que correspondem aos elementos da matriz D . Evidentemente esse sistema é redundante e somente kn equações são independentes. A seleção de um desses conjuntos de kn equações independentes permite determinar completamente a matriz D . Assim, sejam as k entradas x com representação decimal $q^{k-1}, q^{k-2}, \dots, 1$. Isto é,

$$\begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \dots \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}. \quad (4.9)$$

S sejam z_j ($j=1,2, \dots, k$) as respectivas saídas presentes na combinação $(0 v_{22} v_{33} \dots v_{kk})$. Assim, a j -ésima coluna de D corresponde à saída z_j . Isto é,

$$D = [z_1 \ z_2 \ \dots \ z_k]. \quad (4.10)$$

Portanto, para cada uma das possíveis combinações entre os elementos do conjunto especificado em (4.4) e (4.5) existe uma correspondente matriz D expressa através de (4.10). Assim, em (4.3) N_p passa a ser expresso por

$$N_{D^*} = (N_{r^*}-1)!(K'-1)!(N_{r^*})^{(K'-1)} \ll (K-1)! \quad (4.11)$$

e o procedimento de busca agora pode ser realizado apenas sobre as matrizes C, para cada D expressa por (4.10).

Exemplo 4.3: Considere os parâmetros do código do Ex. 4.1 a ser usado para codificar 4PSK em 8PSK com o MPC apresentado na Fig. 3.1. Para aqueles valores, $N_{r^*}=1$, $N_{D^*}=(K-1)!=3!$ e $N_{C^*}=343$. Assim, $N_{C^*}=2.058$ o que representa uma redução de um fator em torno de 10 em relação ao valor obtido no Ex. 4.1. Entretanto, agora a busca poderá ser particionada em 6 etapas, cada etapa sobre um universo de 343 possibilidades. Para o mapeamento da Fig. 3.1 $Z_1=\{0, 2, 4, 6\}$. Assim, para o estado $y=0$, às entradas (0 1 2 3) podem ser associadas as seguintes combinações para as respectivas saídas: (0 2 4 6), (0 4 2 6), (0 6 2 4), (0 2 6 4), (0 4 6 2) e (0 6 4 2). As entradas que satisfazem (4.9) são (2 1), nesta ordem, que correspondem ao terceiro e quarto elementos das combinações de saídas. Isto é, (4 2), (2 4), (2 6), (6 2), (6 4) e (4 6), respectivamente. Assim, de acordo com (4.10), se obtém as 6 matrizes D possíveis e expressas, respectivamente, por

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} . \quad (4.12)$$

O aspecto a ser novamente ressaltado é que, mesmo com a considerável redução possível de ser obtida, a busca sobre o universo das matrizes C implica na geração de todo o seu universo, para cada uma das matrizes D previamente conhecidas. Entretanto, simplificações poderão ocorrer se forem aplicados os conceitos de equivalência de códigos (Peterson e Weldon, 1972) usados por Borelli (1983). Aplicando estes conceitos às matrizes D do exemplo 4.3 somente precisam ser consideradas as matrizes $D=[4\ 2]$, $D=[2\ 6]$ e $D=[6\ 4]$. Nestas condições, o universo de busca é reduzido para $3 \times 343 = 1029$ possibilidades. Maior simplificação pode ser obtida se a equivalência for aplicada às matrizes C. Assim, no caso do Ex. 4.3, aplicando a equivalência de códigos apenas ao terceiro componente de C, o universo é reduzido para $3 \times (3 \times 49) = 441$ possibilidades. Entretanto, mesmo com as consideráveis simplificações possíveis pelo uso da equivalência de códigos à matriz C, pela natureza do algoritmo de busca apresentado a seguir, a equivalência de códigos será aplicada somente à matriz D.

4.4 - O procedimento de busca

O objetivo do procedimento de busca é selecionar os códigos convolucionais que maximizam a distância livre do respectivo código de treliça, cujo codificador possui uma estrutura de memórias especificada através de (3.2). Evidentemente, para que haja ganho $d_{1,10r_{max}}$ é maior que $d_{r_{max},r}$ (veja Eq. 3.20). A Fig. 4.1 ilustra o procedimento que possibilita a seleção dos códigos convolucionais ótimos para uma dada constelação código e uma configuração de memórias pré-estabelecida. Como se pode observar, exceto o bloco correspondente ao cálculo da distância livre, todos os outros podem ser implementados usando-se o modelo desenvolvido nos capítulos 2 e 3. O cálculo da distância livre é um dos pontos cruciais de qualquer procedimento de busca, em razão da natureza do problema envolvido (Eq. 3.14 - 3.16). A seguir descreve-se o método adotado para o cálculo da distância livre.

4.5 - Cálculo da distância livre

É conveniente que sejam repetidas aqui as expressões envolvidas no cálculo da distância livre.

$$d_{1,10r_{max}}^{2k} = \text{Min}_{\{z^r(t)\}_1 \neq \{z^r(t)\}_{2k}} d_{10r_{max}}^{2k}[\{z^r(t)\}_1; \{z^r(t)\}_{2k}] \quad (4.13a)$$

$$d_{10r_{max}}^{2k}[\{z^r(t)\}_1; \{z^r(t)\}_{2k}] = \sum_r d_{10r_{max}}^{2k}[z^r(t)_1; z^r(t)_{2k}] \quad (4.13b)$$

Considere inicialmente os códigos para os quais $N_r=1$. Neste caso, existe uma correspondência biunívoca entre seqüências codimoduladas e seqüências de estado. Isto é, para cada par distinto $\{z^r(t)\}_1$, $\{z^r(t)\}_{2k}$ corresponde um par distinto $\{y^r\}_1$, $\{y^r\}_{2k}$. Somente os símbolos associados aos ramos em que as seqüências de estado diferem contribuem para a avaliação do termo entre colchetes em (4.13b), o que corresponde aos intervalos de tempo associados aos eventos de erro (supondo-se que uma das seqüências é a verdadeira e a outra a versão da primeira corrompida pelo ruído de canal). A Fig. 4.2 ilustra a ocorrência desses eventos. Entretanto, a natureza não decrescente de (4.13b) permite que o problema de minimização seja posto de outra forma.

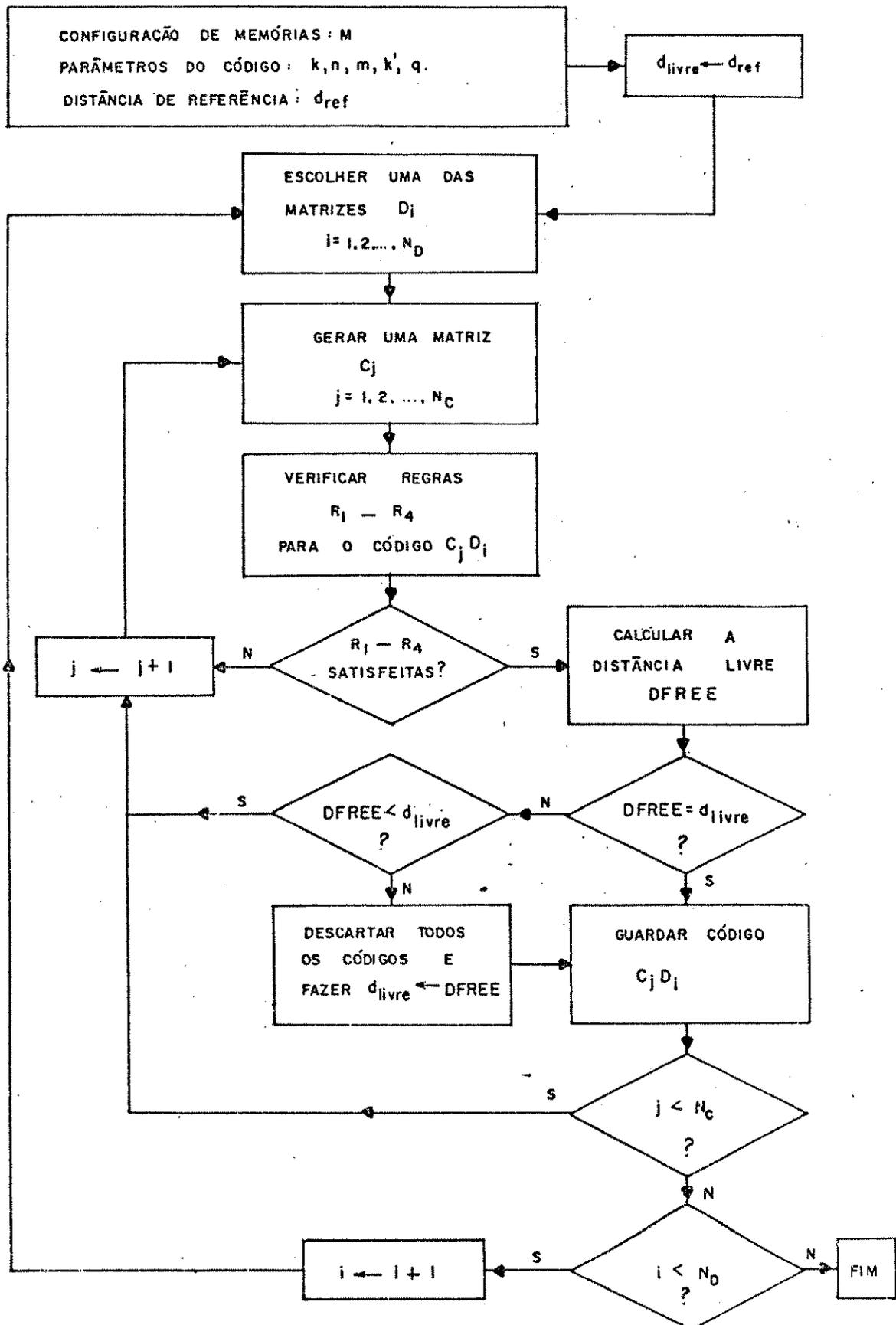


Fig. 4.1: Esquema para implementação do procedimento de busca de códigos ótimos.

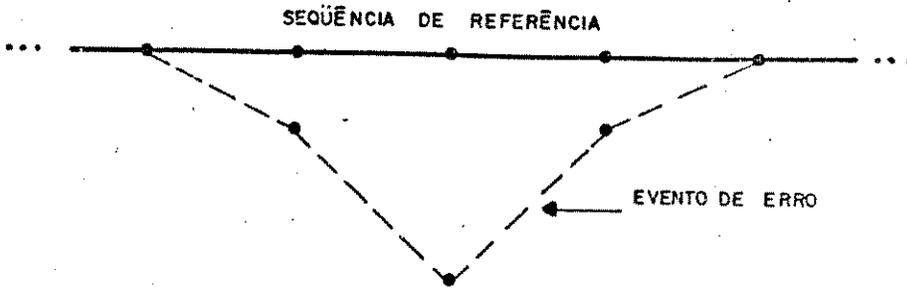


Fig. 4.2: Caracterização da ocorrência de um evento de erro.

Inicialmente, observe que para cada par de seqüências de estado o início de cada evento de erro pode ocorrer em qualquer dos L estados possíveis. Assim, determinando-se as métricas dos eventos originados em cada estado, é possível o cálculo de (4.13b) para cada par de seqüências distintas. Entretanto, no problema de minimização em (4.13a) é suficiente que seja determinada a menor métrica $d_{min}(i)$ dos eventos que se originam do estado i ($i=1,2, \dots, L$). Isto posto, o cálculo da distância livre pode ser efetuado através de

$$d_{free} = \text{Min.}\{d_{min}(1), d_{min}(2), \dots, d_{min}(L)\} .$$

(4.14)

Agora, a dificuldade do problema está reduzida ao cálculo de $d_{min}(i)$, existindo na literatura formas distintas de resolvê-lo como por exemplo, Bahi e outros (1972). Aqui, o procedimento de resolução está baseado numa versão modificada do Algoritmo de Viterbi (AV) e que será descrito a seguir.

Considere todas as seqüências de estado que iniciam no estado i e que são constituídas por δ transições de estado permitidas para o codificador. Cada seqüência pode ser vista como um vetor de $\delta + 1$ componentes L -árias da forma

$$\{y\}_j = [i_0=i, i_1, \dots, i_\delta] ; \quad j=1,2, \dots, K^\delta . \quad (4.15)$$

Cada uma dessas seqüências, desde que válida para o código, será considerada como a seqüência verdadeira, em relação a qual serão formados os eventos de erro cuja seqüência de símbolos correspondente é

$$\{z\}_i = [z_1, z_2, \dots, z_\delta] . \quad (4.16)$$

O AV pode ser usado no cálculo de $d_{\min}(i)$ desde que sejam introduzidas as seguintes considerações:

- 1 - No instante $r=0$ existe apenas um sobrevivente que termina no estado i cuja métrica acumulada é zero e dá origem a todos os outros sobreviventes nos instantes subsequentes.
- 2 - Ao se estenderem os sobreviventes, verificar se ocorre a formação de algum evento de erro. Se isto ocorrer, a métrica do respectivo sobrevivente é a métrica do evento e , no próximo passo do algoritmo, não serão consideradas as transições de estado que se originem desse sobrevivente.
- 3 - O AV deve ser aplicado até que, após a ocorrência de um número conveniente de eventos de erro, as métricas acumuladas dos sobreviventes igualem ou excedam a menor métrica de evento obtida, $d_{\min}(j)$.
- 4 - O AV é aplicado para cada seqüência em (4.15), ao final obtendo-se

$$d_{\min}(i) = \underset{j}{\text{Min}}\{d_{\min}(j)\} . \quad (4.17)$$

- 5 - Os incrementos de métrica a serem calculados para a obtenção de cada sobrevivente correspondem às DE entre os símbolos da constelação código e , portanto, ao contrário do que ocorre na implementação prática do AV, essas distâncias podem ser calculadas e armazenadas previamente.

Tudo que foi exposto acima pressupõe $N_r=1$. Entretanto, o mesmo procedimento se aplica para $N_r > 1$. O que deve ser feito é considerar os ramos paralelos como "super-ramos", mapeados com "super-símbolos" correspondentes aos subconjuntos obtidos na partição de ordem $n-k$ do MPC (veja Eq. 3.11 - 3.13) e considerar como métrica para o AV a matriz de distâncias definida em (3.17), que se reduz à matriz de distâncias entre os símbolos da constelação código quando $N_r=1$.

4.6 - Simplificações no cálculo da distância livre

Apesar da considerável simplificação ocorrida ao avaliar-se (4.13a) através de (4.14) usando-se o AV, ainda assim o problema resulta bastante complexo, já que o algoritmo deve ser aplicado LK^δ vezes. Este número, mesmo para valores pequenos de L, K e δ é bastante significativo, sendo este o motivo do posicionamento do bloco de cálculo da distância livre no algoritmo ilustrado na Fig. 4.1. Entretanto, é possível a obtenção de alguns procedimentos de simplificação do problema. O primeiro deles se fundamenta no conceito de superlinearidade introduzido por Marsen e outros (1984), segundo o qual se um código for superlinear é suficiente serem considerados no cálculo da distância livre apenas os eventos relativos somente a uma única das seqüências em (4.15). O formalismo para a caracterização de um código superlinear, segundo o modelo que vem sendo desenvolvido neste trabalho, é descrito a seguir.

Sejam Z_i ($i=1,2, \dots, N'=N/N_p$) os subconjuntos cujos elementos são os símbolos presentes nas respectivas constelações de ordem $n'=n-k$ do MPC. Estes subconjuntos podem ser vistos como super-símbolos de uma constelação com N' pontos, aos quais, pelo MPC, estão associados N' vetores q-ários com n' componentes. O conjunto dos N' vetores forma um grupo abeliano, $A^{n'}$, em relação à operação soma módulo q, componente a componente, denotada aqui por \oplus . Seja então H a matriz soma correspondente a esse grupo. Isto é,

$$H = [h(i,j)], \quad i,j=1,2, \dots, N', \quad (4.18)$$

onde $h(i,j)=Z_i \oplus Z_j$.

A constelação dos símbolos Z_i também está associada uma matriz de distâncias especificada por (3.17) e repetida aqui por conveniência.

$$D_{\text{MC}} = [d(Z_i, Z_j)], \quad i,j=1,2, \dots, N'. \quad (4.19)$$

A definição para códigos superlineares, segundo Marsen e outros (1984), é dada a seguir.

Definição 4.1 : Um código é superlinear se satisfaz às propriedades seguintes:

1 - Existe uma operação denominada soma e representada por "+", em relação a qual os símbolos Z_i ($i=1,2, \dots, N'$) formam um grupo abeliano.

2 - O código caracterizado pelo mapeamento entre os conjuntos $A^{n'}$ e $\{Z_1, Z_2, \dots, Z_N\}$ é um homomorfismo. Isto é, se z_i, z_j e z_k são elementos de $A^{n'}$ associados pelo mapeamento a Z_i, Z_j e Z_k , respectivamente, então $z_i \oplus z_j = z_k$ implica em $Z_i + Z_j = Z_k$.

3 - É possível associar a cada símbolo Z_i um número real $w(Z_i)$, denominado peso de Z_i , tal que $d(Z_i, Z_j) = w(Z_i + Z_j)$, onde estes pesos são uma extensão dos pesos de Hamming.

Definidos os códigos superlineares, é possível estabelecer o seguinte teorema (Marsen e outros, 1984):

Teorema 4.1 - Um código é superlinear se elementos iguais na matriz de adição do conjunto $\{Z_1, Z_2, \dots, Z_N\}$ corresponde a distâncias iguais nas mesmas posições da matriz de distâncias associada a esse conjunto. Isto é, se para $Z_i + Z_j = Z_k + Z_l$, $d(Z_i, Z_j) = d(Z_k, Z_l)$, o código é superlinear.

Prova : A cada símbolo Z_i está associado um vetor z_i em $A^{n'}$ e este é um grupo abeliano em relação à operação $+$. Agora, se $Z_i + Z_j = Z_k + Z_l$, significa que $z_i \oplus z_j = z_k \oplus z_l$ e, portanto, o conjunto $\{Z_1, Z_2, \dots, Z_N\}$ forma um grupo abeliano em relação à operação $+$. Assim, as duas primeiras condições da Def. 4.1 estão verificadas. Se na matriz de adição $Z_i + Z_j = Z_k + Z_l$ e $d(Z_i, Z_j) = d(Z_k, Z_l)$, então a terceira condição da Def. 4.1 também é verificada e portanto o código é superlinear.

Uma consequência do teorema anterior e do homomorfismo é que um código superlinear pode ser caracterizado pela seguinte definição (Marsen e outros, 1984):

Definição 4.2 : Um código é superlinear se elementos iguais em (4.18) correspondem a elementos iguais em (4.19).

Note que quando $N_p = 1$ a matriz (4.18) se reduz à matriz de adição do corpo de Galois de extensão formado pelos sub-blocos do código convolucional, enquanto (4.19) corresponde às DE entre os símbolos da constelação código. Ungerboeck (1982), definiu pesos para os símbolos, através dos sub-blocos, com o intuito de obter um limite inferior para a distância livre de um código no caso específico em que $q=2$. Esses pesos são definidos por

$$w(z_i) = \begin{cases} \Delta_0 & ; z_i = 0 \\ \Delta_j & ; z_i \neq 0 \end{cases}, \quad \begin{matrix} j = 0, 1, \dots, n-1 \\ i = 1, 2, \dots, N \end{matrix}$$

(4.20)

onde J é o número de zeros antes da ocorrência do primeiro 1 menos significativo em z_i , e Δ_j as mínimas DE definidas no MPC. Usando-se esses pesos, a distância entre dois símbolos da constelação código associadas pelo MPC às palavras código z_i e z_j é expressa por

$$d[z_i(t), z_j(t)] = w(z_i \oplus z_j), \quad i, j = 1, 2, \dots, N. \quad (4.21)$$

Assim, a matriz de distâncias em (4.19) pode ser obtida usando-se (4.21) como definição para a distância entre símbolos. Nestas condições, palavras código idênticas em (4.18) correspondem a pesos idênticos em (4.19) e em consequência todos os códigos para os quais (4.20) possa ser verificada são superlineares. Entretanto, o uso dos pesos definidos em (4.20) pressupõe que no MPC

$$\Delta_i > \Delta_{i-1}, \quad i = 1, 2, \dots, n-1 \quad (4.22)$$

que é uma condição que nem sempre pode ser verificada para constelações tridimensionais. Como neste trabalho a condição (4.22) é substituída pela (3.7), é praticamente impossível a obtenção da condição de superlinearidade para o caso em que $N_p = 1$. Os exemplos a seguir esclarecem melhor o problema da superlinearidade.

Exemplo 4.4 : Seja o MPC ilustrado na Fig. 3.1 a ser usado com um código covolucional de parâmetros $k=2$, $n=3$, $m=3$, $q=p=2$ e $k''=0$.

Nestas condições, $N_p=1$ e os vetores Z_i correspondem às próprias palavras-código (sub-blocos). Assim, a matriz de adição associada ao grupo abeliano é expressa por

$$H = \begin{bmatrix} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ 001 & 000 & 011 & 010 & 101 & 100 & 111 & 110 \\ 010 & 011 & 000 & 001 & 110 & 111 & 100 & 101 \\ 011 & 010 & 001 & 000 & 111 & 110 & 101 & 100 \\ 100 & 101 & 110 & 111 & 000 & 001 & 010 & 011 \\ 101 & 100 & 111 & 110 & 001 & 000 & 011 & 010 \\ 110 & 111 & 100 & 101 & 010 & 011 & 000 & 001 \\ 111 & 110 & 101 & 100 & 011 & 010 & 001 & 000 \end{bmatrix} \quad (4.23)$$

onde considera-se uma ordenação crescente para a representação decimal dos sub-blocos. A matriz de distâncias correspondente, segundo (3.17), é expressa por

$$D_{III} = \begin{bmatrix} 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{1'} & \Delta_{2.} & \Delta_{2'} & \Delta_{3.} & \Delta_{\emptyset} \\ \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{1'} & \Delta_{2.} & \Delta_{2'} & \Delta_{3.} \\ \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{1'} & \Delta_{2.} & \Delta_{2'} \\ \Delta_{1'} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{1'} & \Delta_{2.} \\ \Delta_{2.} & \Delta_{1'} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{1'} \\ \Delta_{2'} & \Delta_{2.} & \Delta_{1'} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} \\ \Delta_{3.} & \Delta_{1'} & \Delta_{2.} & \Delta_{1.} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} \\ \Delta_{\emptyset} & \Delta_{1.} & \Delta_{1'} & \Delta_{2.} & \Delta_{1'} & \Delta_{1.} & \Delta_{\emptyset} & 0 \end{bmatrix}$$

(4.24)

onde $\Delta_{1'}^{1'2} = \Delta_{2.}^{22} - \Delta_{\emptyset}^{22}$. Entretanto, usando-se (4.21) a matriz de distâncias é expressa por

$$D_{III} = \begin{bmatrix} 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & \Delta_{2.} & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} \\ \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & \Delta_{2.} & \Delta_{\emptyset} & \Delta_{1.} \\ \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & \Delta_{2.} & \Delta_{\emptyset} \\ \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & \Delta_{2.} \\ \Delta_{2.} & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} \\ \Delta_{\emptyset} & \Delta_{2.} & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} & \Delta_{1.} \\ \Delta_{1.} & \Delta_{\emptyset} & \Delta_{2.} & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & 0 & \Delta_{\emptyset} \\ \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & \Delta_{2.} & \Delta_{\emptyset} & \Delta_{1.} & \Delta_{\emptyset} & 0 \end{bmatrix}$$

(4.25)

Uma inspeção nas diagonais secundárias de (4.23) e (4.24) é suficiente para se verificar que as mesmas não satisfazem à condição da superlinearidade. Por outro lado, uma inspeção em (4.23) e (4.25) mostra que as mesmas satisfazem à condição da superlinearidade. Portanto, se no procedimento de busca de códigos ótimos, para a constelação de que trata este exemplo, (4.25) for usada como métrica, somente uma seqüência de referência precisa ser considerada para o cálculo da distância livre.

Exemplo 4.5 : Considere novamente o Ex. 4.4, porém com $k''=1$. Neste caso, $N_{r'}=2$ e os subconjuntos envolvidos são:

$$Z_1 = \{0, 4\} \quad Z_2 = \{2, 6\} \quad Z_3 = \{1, 5\} \quad Z_4 = \{3, 7\}$$

correspondendo aos vetores $Z_1 = (0 \ 0)$, $Z_2 = (0 \ 1)$, $Z_3 = (1 \ 0)$ e $Z_4 = (1 \ 1)$. A matriz soma para o grupo abeliano é expressa por

$$H = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 01 & 00 & 11 & 10 \\ 10 & 11 & 00 & 01 \\ 11 & 10 & 01 & 00 \end{bmatrix} .$$

(4.26)

Agora, usando tanto (3.17) quanto (4.21) se obtém a mesma matriz de distâncias, expressa por

$$D_{\text{int}} = \begin{bmatrix} 0 & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{12} & 0 & \Delta_{23} & \Delta_{24} \\ \Delta_{13} & \Delta_{23} & 0 & \Delta_{34} \\ \Delta_{14} & \Delta_{24} & \Delta_{34} & 0 \end{bmatrix}$$

(4.27)

Assim, para este exemplo, qualquer das duas definições (Eq. 3.17 ou 4.21) de métrica, os códigos serão superlineares.

Infelizmente, para códigos em que $N_{\text{p}} > 1$, a distância livre é limitada superiormente pela mínima DE entre os símbolos associados aos ramos paralelos. Isto é,

$$d_{\text{livre}} \leq \Delta_{n'} \quad , \quad n' = n - k \quad \text{e} \quad N_{\text{p}} > 1. \quad (4.28)$$

Portanto, se o máximo ganho possível (veja 3.20) para códigos com $N_{\text{p}} > 1$ não for aceitável para uma aplicação específica, a otimização do código terá que ser realizada sobre o universo de códigos em que $N_{\text{p}} = 1$. Nestas condições, a superlinearidade só será verificada se (4.22) for verdadeira para a constelação código. Em caso contrário, a simplificação do cálculo da distância livre terá que ser implementada, se possível, através de outros procedimentos. Um desses procedimentos será descrito a seguir.

Suponha que no MPC as constelações obtidas em cada subpartição possuam o mesmo conjunto de distâncias Euclidianas entre seus símbolos. Isto é, através de rotações e translações é possível fazer coincidir todas as constelações de uma mesma subpartição. As regras R1 e R2 introduzem na treliça associada ao código essas mesmas propriedades, exceto pela ordem em que as distâncias aparecem para cada conjunto de ramos que se origina de, ou que se destina a, um mesmo estado. Assim, nestas condições, o cálculo da distância livre pode ser realizado aplicando-se o AV apenas para as seqüências de referência que se originam de apenas um dos estados, o que reduz a complexidade do problema de um fator L. É evidente que essa simplificação só se aplica a constelações código simétricas em relação aos eixos do sistema de coordenadas do espaço de sinais que, felizmente, é o caso mais freqüente.

Uma observação final no que se refere ao cálculo da distância livre. Note que os pesos definidos em (4.20) apenas asseguram um limite inferior para a distância livre (Ungerboeck, 1982). Assim, no procedimento de otimização de códigos, quando $N_{\text{p}} = 1$, a métrica obtida através de (4.21), caso seja possível sua utilização, deve ser usada para uma seleção preliminar dos códigos. A seleção final terá que ser realizada usando-se a métrica definida em (3.17).

4.7 - Otimização de códigos com máxima distância livre

O procedimento de otimização ilustrado na Fig.4.1 assegura a seleção dos códigos com máxima distância livre e portanto todos apresentam o mesmo ganho assintótico (veja Eq. 3.20). Entretanto, o verdadeiro desempenho de um código é determinado através da avaliação das probabilidades de erro de evento e de erro de bit, P_{ev} e P_b , respectivamente. Infelizmente, o cálculo dessas probabilidades é bastante difícil e, em geral, apenas podem ser estimadas através de limitantes superiores e inferiores (Forney, 1972 e 1973) ou então por simulação em computador. No caso específico do cálculo dos limitantes das probabilidades, os seguintes fatores são fundamentais: As diferentes métricas d_i ($i=0,1,2, \dots$) dos eventos de erro possíveis de ocorrer, o número de eventos N_i com métrica d_i e W_i , o número total W_i de erros de bit presente nestes N_i eventos.

Supondo-se que o ruído de canal é gaussiano branco e aditivo, com variância σ^2 por dimensão, o limite superior para P_{ev} pode ser escrito como (Forney, 1972)

$$P_{ev} < Q(d_\phi/2\sigma) \exp(d_\phi^2/8\sigma^2) \sum_{i=0}^{\infty} N_i \exp(-d_i^2/8\sigma^2) \quad (4.29)$$

onde d_ϕ é a distância livre do código, (Eq.4.13), e $Q(x)$ é a integral, a partir de x , da cauda da função gaussiana normalizada. O correspondente limite para P_b é expresso por

$$P_b < Q(d_\phi/2\sigma) \exp(d_\phi^2/8\sigma^2) \sum_{i=0}^{\infty} W_i \exp(-d_i^2/8\sigma^2) \quad (4.30)$$

Para altas relações sinal/ruído, (4.29) e (4.30) são dominadas pelos termos envolvendo d_ϕ e são aproximadas por

$$P_{ev} \cong N_\phi Q(d_\phi/2\sigma) \quad (4.31a)$$

$$P_b \cong W_\phi Q(d_\phi/2\sigma) \quad (4.31b)$$

e que correspondem aos limites inferiores para P_{ev} e P_b , respectivamente (Forney, 1972).

Considerando as parcelas em (4.29) e (4.30) ordenadas segundo a ordem crescente das métricas d_i , somente os primeiros termos são significativos a partir de médias relações sinal/ruído ($P_{e\downarrow} < 10^{-4}$). Como na implementação do AV (versão subótima) cada decisão é tomada com base em observações para cada sobrevivente, versões truncadas de (4.29) e (4.30) podem ser obtidas considerando-se apenas as métricas dos eventos que possam ocorrer para uma profundidade na treliça associada ao código. Assim, se N é o número dessas métricas, pode-se obter boas estimativas dos limites superiores de $P_{e\downarrow}$ e P_b para altas razões sinal/ruído, os quais podem ser expressos, respectivamente, por

$$P_{e\downarrow} \cong Q(d_\phi/2\sigma) \exp(d_\phi^{2\alpha}/8\sigma^{2\alpha}) \sum_{i=0}^{N_\delta-1} N_i \exp(-d_i^{2\alpha}/8\sigma^{2\alpha}) \quad (4.32a)$$

$$P_b \cong Q(d_\phi/2\sigma) \exp(d_\phi^{2\alpha}/8\sigma^{2\alpha}) \sum_{i=0}^{N_\delta-1} W_i \exp(-d_i^{2\alpha}/8\sigma^{2\alpha}) \quad (4.32b)$$

Essas expressões serão denominadas aqui de limites superiores truncados. Portanto, para um mesmo conjunto ordenado de métricas

$$\{d_\phi, d_1, \dots, d_{N_\delta-1}\} \quad (4.33)$$

o código que tem melhor desempenho para médias e altas relações sinal/ruído deve apresentar, na mesma ordem, os mínimos N_i e W_i possíveis.

O procedimento de otimização que permite a seleção do código ótimo terá que ser implementado em duas etapas. Na primeira são selecionados os códigos que apresentem os menores N_i e por conseguinte o menor $P_{e\downarrow}$, segundo (4.32a). Na segunda etapa, a partir dos códigos obtidos na primeira fase, são selecionados os códigos que apresentem os mínimos W_i e por conseguinte minimizam P_b , segundo (4.32b).

Apesar de conceitualmente simples, o procedimento de otimização acima descrito é de difícil implementação sob o ponto de vista do esforço computacional envolvido, mesmo sabendo-se que o número de códigos envolvidos no problema (os de máxima distância livre) é bem reduzido quando comparado com o número inicial

presente na etapa de busca dos códigos de máxima distância livre. A principal razão é que para se obter o conjunto de métricas em (4.33) é necessário que se apliquem procedimentos semelhantes aos usados no cálculo da distância livre. Se os códigos a serem utilizados forem superlineares é suficiente se obter (4.33) para os eventos relativos a uma única seqüência de referência. Apesar dessa condição nem sempre ser verificada e tendo em vista a complexidade computacional envolvida no problema, neste trabalho a otimização dos códigos de máxima distância livre será considerada determinando-se as métricas dos eventos relativos a uma única seqüência de referência. Assim, os códigos serão ótimos no sentido de apresentarem os mínimos N_i e W_i em relação a uma seqüência de referência que será suposta igual à seqüência de estados $(0, 0, \dots, 0)$. O apêndice 4A apresenta o algoritmo para a implementação desses procedimentos.

APÊNDICE 4A

Algoritmo para implementação da otimização dos códigos de máxima distância livre associados a uma configuração de memórias M e a uma matriz D .

1 - Armazenamento

- 1.1 - Configuração de memórias: M
- 1.2 - Matriz D .
- 1.3 - Número de códigos associados a M e D : N_{cod}
- 1.4 - Matrizes C_i ($i=1,2,\dots,N_{cod}$) armazenadas como linhas de uma matriz CC .
- 1.5 - Profundidade a ser pesquisada na treliça:
- 1.6 - Seqüência de estados de referência: $Y_{r_{ref}}$
- 1.7 - Vetor das métricas distintas dos eventos de erro obtidos em relação a $Y_{r_{ref}}$ até a profundidade d .
- 1.8 - Número de métricas distintas observadas até a profundidade d : N_δ
- 1.9 - Vetor dos pesos N_i ($i=1,2,\dots,N$): N
- 1.10 - Vetor dos pesos W_i ($i=1,2,\dots,N$): W
- 1.11 - Mapeamento da treliça com as entradas: X
- 1.12 - Mapeamento da treliça com as saídas: Z
- 1.13 - Matriz das métricas do AV: D_{AV}
- 1.14 - Vetor auxiliar com N_{cod} componentes: X_{aux}

2 - Inicialização

- 2.1 - $i \leftarrow 0$
- 2.2 - $j \leftarrow 0$

3 - Fase preliminar: Obtenção de d , e N .

- 3.1 - Considerar o código D , C_1 .
- 3.2 - Gerar a matriz das métricas para o AV.

- 3.3 - Gerar o mapeamento da treliça com as entradas e saídas.
- 3.4 - Usando o AV modificado determinar as métricas dos eventos distintos: d_i .
- 3.5 - Ordenar as métricas segundo a ordem crescente.
- 3.6 - O número de métricas distintas obtidas no passo 3.4 representa o valor de N_δ .
- 4 - Primeira fase da otimização: Seleção dos códigos D_i, C_j, j $\{1, 2, \dots, N_{cod}\}$, para os quais N_i é mínimo ($i=1,2, \dots, N_\delta$).
- 4.1 - Fazer $i \leftarrow i + 1$ e considerar a métrica d_i .
- 4.2 - Fazer $j \leftarrow j + 1$.
- 4.3 - Se $i=1$, passar ao passo 4.4. Em caso contrário, repetir o passo 4.2 até que $X_{aux}(j)=N(i-1)$.
- 4.4 - Gerar o mapeamento da treliça com as saídas para o código D_i, C_j .
- 4.5 - Usar o AV modificado, determinar N_i e fazer $X_{aux}(j) \leftarrow N_i$.
- 4.6 - Repetir os passos 4.2 - 4.5 até que $j=N_{cod}$.
- 4.7 - Para $j = 1,2, \dots, N_{cod}$, determinar, para $X_{aux}(j) > 0$, $\text{Min}[X_{aux}(j)]$ e fazer $N(i) \leftarrow \text{Min}[X_{aux}(j)]$.
- 4.8 - Para $j=1,2, \dots, N_{cod}$, fazer $X_{aux}(j) \leftarrow -X_{aux}(j)$ se $X_{aux}(j) > 0$ e $X_{aux}(j) \neq N(i)$.
- 4.9 - Fazer $j \leftarrow 0$ e repetir os passos 4.1 a 4.8 até que $i = N_\delta$.
- 5 - Segunda fase da otimização: Para os códigos seleccionados na primeira fase seleccionar os que apresentarem os mínimos W_i , $i = 1, 2, \dots, N_\delta$.
- 5.1 - Fazer $i \leftarrow 0$ e $j \leftarrow 0$.
- 5.2 - Fazer $i \leftarrow i + 1$ e considerar a métrica d_i .
- 5.3 - Fazer $j \leftarrow j + 1$
- 5.4 - Repetir o passo 5.3 até que $X_{aux}(j) > 0$.

- 5.5 - Se $i=1$, passar ao passo 5.6. Em caso contrário, repetir os passos 5.3 - 5.5 até que $X_{aux}(j) = W(i-1)$
- 5.6 - Obter o mapeamento da treliça com as saídas para o código D, C_j .
- 5.7 - Usar o AV modificado, determinar W_i e fazer $X_{aux}(j) \leftarrow W_i$.
- 5.8 - Repetir os passos 5.3 - 5.7 até que $j = N_{cod}$.
- 5.9 - Para $j = 1, 2, \dots, N_{cod}$, determinar, para $X_{aux}(j) > 0$, $\text{Min}[X_{aux}(j)]$ e fazer $W(i) \leftarrow \text{Min}[X_{aux}(j)]$.
- 5.10 - Para $j = 1, 2, \dots, N_{cod}$, fazer $X_{aux}(j) \leftarrow -X_{aux}(j)$ se $X_{aux}(j) > 0$ e $X_{aux}(j) \neq W(i)$.
- 5.11 - Fazer $j \leftarrow 0$ e repetir os passos 5.2-5.10 até que $i = N_\delta$.

CAPÍTULO 5

PROPRIEDADES ESPECTRAIS DE SINAIS CODIMODULADOS

5.1 - Introdução

A principal característica dos sistemas que usam modulação codificada controlada por um código convolucional é a obtenção de ganhos de potência com uma simultânea conservação da faixa de transmissão. Isto é, as propriedades espectrais do sinal codimodulado são idênticas às de um sinal modulado que usasse como símbolos de transmissão os símbolos da constelação de referência. Assim, em princípio não há a necessidade do cálculo das propriedades espectrais de sinais codimodulados já que as mesmas podem ser obtidas a partir dos sistemas de referência usando-se procedimentos próprios (Prabhu e Rowe, 1974, 1975; Billardi e outros, 1983). Entretanto, a invariança espectral pressupõe que tanto a constelação de referência quanto a constelação código sejam caracterizadas em espaços de sinais de mesma dimensionalidade. Além disso, quase sempre, assume-se no cálculo espectral que os símbolos são transmitidos com regularidade, o que nem sempre poderá ser o caso.

Um dos objetivos deste trabalho é a investigação de códigos ótimos para os quais as constelações acima referidas estão caracterizadas em espaços com dimensionalidades distintas. Como será visto no capítulo 6, essa particularidade pode proporcionar uma melhora na margem teórica de ganho associada a uma constelação código, relativamente a um sistema de referência. É claro que existe um preço a ser pago por essa possibilidade de melhora. Esse preço é o aumento da faixa de transmissão como consequência do aumento da dimensionalidade. Para se analisar esse aumento de faixa, se faz necessário o desenvolvimento de procedimentos que permitam a avaliação das propriedades espectrais de sinais codimodulados com as características acima mencionadas. Além desses aspectos, existe um outro que merece ser investigado, qual seja verificar a influência da correlação introduzida no sinal codimodulado pelo código convolucional de controle já que, segundo observado inicialmente por Divsalar e Simon (1980) e posteriormente por Bezerra e Arantes (1986b), o espectro em banda base de sinais codificados por códigos convolucionais podem apresentar propriedades de compressão de espectro em relação ao espectro do sinal sem codificação. O procedimento a ser desenvolvido neste capítulo se fundamenta no método de cálculo de espectro de sinais codificados por códigos convolucionais desenvolvido por Bezerra e Arantes (1986). Esse método apresenta a vantagem de ser totalmente matricial e conseqüentemente apropriado à implementação em computador. Também apresenta a flexibilidade de ser aplicado a sinais codificados por códigos de bloco, vistos como códigos convolucionais com restrição de memória $m=1$, como também sinais digitais sem codificação se encarados como tendo sido codificados por um código de bloco cuja matriz geradora é a matriz identidade.

5.2 - Caracterização temporal do sinal codimodulado

A matriz P expressa através da Eq. 3.6 possui todas as informações concernentes ao modo como o código convolucional controla a transmissão dos símbolos da constelação código. Isto é, se no r -ésimo intervalo de tempo a saída do codificador é z^r , o símbolo a ser transmitido é especificado através das coordenadas do vetor p_i ,

$$p_i = [p_{i1}, p_{i2}, \dots, p_{iQ}] \quad i = (z^r)_{10} + 1. \quad (5.1)$$

Assim, se $\{\varphi_1(t), \varphi_2(t), \dots, \varphi_Q(t)\}$ é uma base ortonormal para o espaço de sinais de dimensionalidade Q , a representação temporal do respectivo símbolo é expressa por

$$p_i(t) = \sum_{j=1}^Q p_{ij} \varphi_j(t) \quad , \quad 0 \leq t \leq T_m \quad (5.2)$$

$i \in \{1, 2, \dots, N\}$,

onde $T_m = kT_m$ é a duração do intervalo de modulação, T_m é a duração do intervalo dos símbolos q -ários e os componentes da base satisfazem à relação

$$\int_0^{T_m} \varphi_i(t) \varphi_j(t) dt = \begin{cases} 1 & ; \quad i=j \\ 0 & ; \quad i \neq j \end{cases} \quad (5.3)$$

A representação temporal para os símbolos da constelação código pode ser expressa através do vetor coluna

$$p(t) = [p_1(t), p_2(t), \dots, p_N(t)]^t \quad (5.4a)$$

cujos componentes satisfazem à expressão matricial

$$p(t) = P \phi(t) \quad (5.4b)$$

onde $\phi(t) = [\varphi_1(t), \varphi_2(t), \dots, \varphi_Q(t)]^t$.

Como cada elemento da base é de duração finita e bem comportado, a sua transformada de Fourier (TF) existe. Assim, a caracterização espectral da constelação código é expressa através de

$$P(f) = P \Phi(f), \quad (5.5)$$

resultante da TF de (5.4). Note que tanto em (5.4b) quanto em (5.5) o MPC se faz presente através do índice de linha.

O sinal codimodulado pode ser representado no domínio do tempo por

$$z_{cc}(t) = \sum_r z(t-rT_m-t_\phi), \quad (5.6)$$

onde $z(t-rT_m-t_\phi)$ é o símbolo transmitido sob controle da palavra-código z^r e t_ϕ corresponde a um atraso aleatório, suposto uniformemente distribuído no intervalo $[0, T_m]$.

A Eq. (5.6) corresponde, através do MPC, a uma seqüência $\{z^r\}$ gerada pelo codificador do código convolucional segundo sua função de saída (Eq. 2.11b ou Eq. 2.18). Assim, às matrizes $A_{i,u}$ (Eq. 2.17) correspondem vetores coluna temporais

$$A_{i,u}(t) = [z_i(u; t)] , \quad i=1,2, \dots, L, \quad (5.7a)$$

que caracterizam completamente a saída do codimodulador, quando para cada entrada $x_{i,u}$ ($u=1,2, \dots, K$), o codificador do código convolucional passa por todos os estados. Através de (5.4) e (5.5) se obtém a correspondente representação no domínio da freqüência. Isto é,

$$A_{i,u}(f) = [z_i(u; f)] , \quad i=1,2, \dots, L. \quad (5.7b)$$

A caracterização nos domínios do tempo e da freqüência expressa através de (5.1)-(5.7) juntamente com a representação matricial para a função transição de estado do codificador é suficiente para se determinar as propriedades espectrais associadas ao sinal codimodulado.

5.3 - Espectro do sinal codimodulado

O espectro do sinal codimodulado é expresso pela TF de sua função de autocorrelação

$$\begin{aligned} R_{z_c}(t, t+\tau) &= R_{z_c}(\tau) \\ &= E\{z_c(t)z_c(t+\tau)\} \end{aligned} \quad (5.8)$$

onde $E(\cdot)$ significa média estatística. No apêndice 5A são apresentados os passos envolvidos no cálculo de (5.8), os quais resultam em

$$R_{z_c}(\tau) = \sum_r \frac{1}{T_m} \sum_{u,v=1}^K \int_{-\infty}^{\infty} dt' A_{u,c}^T(t-t') P_{r^{u,v}} A_{v,c}(t+\tau-t'-rT_m) \quad (5.9)$$

onde $A_{u,c}(\cdot)$ e $A_{v,c}(\cdot)$ são obtidas a partir de (5.6), T significa transposição e $P_{r^{u,v}}$ é uma matriz expressa por

$$P_{r^{u,v}} = [P_{r^{u,v}}(i,j)] \quad , \quad i,j=1,2, \dots, L \quad (5.10)$$

cujos elementos $P_{r^{u,v}}(i,j)$ representam a probabilidade de que no instante s a entrada e o estado do codificador sejam, respectivamente, $x_{i,c}$ e y_j e que no instante $s+r$ a entrada e o estado sejam $x_{j,c}$ e y_i , respectivamente. Isto é,

$$P_{r^{u,v}} = \text{Prob.}[(y^m=y_j, x^m=x_{i,c}) \cap (y^{m+r}=y_i, x^{m+r}=x_{j,c})] \quad (5.11)$$

Supondo-se que o sinal de entrada $\{x^m\}$ seja estacionário no sentido fraco com independência estatística entre os elementos x^m , os resultados de Carliolaro (1974) podem ser usados. Isto é,

$$P_{r^{u,v}} = \begin{cases} q_{i,c} \delta_{i,j} \overset{\infty}{D} & ; \quad r = 0 \\ q_{i,c} q_{j,c} \overset{\infty}{D} E_{i,c} \Pi^{r-1} & ; \quad r > 0 \end{cases} \quad (5.12a)$$

$$q_{i,c} q_{j,c} \overset{\infty}{D} E_{i,c} \Pi^{r-1} \quad (5.12b)$$

$$\begin{aligned} P_{\infty}^{u,v} &= \lim_{r \rightarrow \infty} P_{r^{u,v}} \\ &= q_{i,c} q_{j,c} \overset{\infty}{D} E_{i,c} \end{aligned} \quad (5.12c)$$

onde

$$\Pi = \sum_{u=1}^K q_{i,u} E_{i,u} \quad (5.13)$$

é a matriz transição de estado do codificador, $q_{i,u} = \text{Prob.}(x_{i,u})$, $q_{i,i} = \text{Prob.}(x_{i,i})$ e \mathbf{U} é uma matriz diagonal com elementos não nulos pertencentes, ordenadamente, ao vetor

$$\mathbf{p}^{\infty} = [p_{11}, p_{22}, \dots, p_{L,L}]^T \quad (5.14)$$

que é a solução para o sistema de equações

$$\left\{ \begin{array}{l} \Pi^T \mathbf{p}^{\infty} = \mathbf{p}^{\infty} \\ \sum_{i=1}^L p_i = 1 \end{array} \right. \quad (5.15)$$

Finalmente, Π_{∞} é a matriz limite da matriz transição de estado e que tem como linhas o transposto do vetor \mathbf{p}^{∞} .

Agora, tomando-se a TF de (5.9) e usando-se (5.7) se obtém

$$W_{\infty}(f) = \frac{1}{T_{\text{m}}} \sum_{u,v=1}^K A_{i,u}^{*} (f) \sum_{r=1}^{\infty} P_{r-1}^{i,u} \exp(-j2\pi fr T_{\text{m}}) A_{i,v}(f) \quad (5.16)$$

onde * significa o transposto do conjugado. Os detalhes da obtenção de (5.16) são apresentados no apêndice 5B. Seguindo Cariolaro (1974), com pequenas diferenças, (5.16) pode ser expressa por meio de

$$W_{\infty}(f) = \frac{1}{T_{\text{m}}} [W_{\text{c}}(f) + W_{\text{d}}(f)] \quad (5.17)$$

onde a primeira parcela corresponde à parte contínua do espectro enquanto a segunda parcela corresponde à parte discreta e são expressas, respectivamente, por

$$\begin{aligned}
 W_{cl}(f) = & \sum_{u=1}^K q_{i,u} A_{i,u}^{**}(f) \overset{\infty}{D} A_{i,u}(f) + \\
 & + G_{i,1}^{**}(f) [2\text{Real} \{ (U - \Pi_{\infty}) [wU - (\Pi - \Pi_{\infty})]^{-1} \} - \Pi_{\infty}] G_{i,2}(f)
 \end{aligned}
 \tag{5.18}$$

$$W_{cl}(f) = \frac{1}{T_{\mu}} G_{i,1}^{**}(f) \Pi_{\infty} G_{i,2}(f) \sum_{r=-\infty}^{\infty} \delta(f - r/T_{\mu}) ,
 \tag{5.19}$$

onde U é uma matriz identidade de ordem L , $w = \exp(j2\pi f T_{\mu})$ e $G_{i,1}^{**}(f)$, $G_{i,2}(f)$ são vetores de L componentes expressos por

$$G_{i,1}^{**}(f) = \sum_{u=1}^K q_{i,u} A_{i,u}^{**}(f) \overset{\infty}{D} E_{i,u}
 \tag{5.20}$$

$$G_{i,2}(f) = \sum_{u=1}^K q_{i,u} A_{i,u}(f) .
 \tag{5.21}$$

No apêndice 5C são apresentados os detalhes para a obtenção das Eqs. (5.18)-(5.21).

5.4 - Espectro para sinais sem codificação

Como foi dito anteriormente, um sinal sem codimodulação é equivalente a um sinal codimodulado que usa na codificação um código de bloco cuja matriz geradora é a matriz identidade de ordem k . Para um código de bloco, $\mu = 0$ e portanto existe somente um estado. Nestas condições, $\Pi = \Pi_{\infty} = \overset{\infty}{D} = E_{i,i} = 1$, além dos vetores expressos por (5.7) se reduzirem às TF dos símbolos da constelação. Assim, (5.18) e (5.19) se reduzem, respectivamente, a

$$W_{cc}(f) = \sum_{u=1}^K q_{i,u} |X_{i,u}(f)|^2 - \sum_{u,v=1}^K q_{i,u} q_{i,v} X_{i,u}^{**}(f) X_{i,v}(f) \quad (5.22)$$

$$W_{cl}(f) = \frac{1}{T_m} \sum_{u,v=1}^K q_{i,u} q_{i,v} X_{i,u}^{**}(f) X_{i,v}(f) \sum_{r=-\infty}^{\infty} \delta(f-r/T_m) \quad (5.23)$$

APÊNDICE 5A

A autocorrelação para $z_m(t)$ em (5.6) é expressa por

$$\begin{aligned} R_{zz}(\tau) &= E\{z_m(t)z_m(t+\tau)\} = \\ &= E\left\{\sum_p z_m(t-pT_m-t_\phi) \sum_q z_m(t+\tau-qT_m-t_\phi)\right\} = \\ &= \sum_{p,q} E\{z_m(t-pT_m-t_\phi)z_m(t+\tau-qT_m-t_\phi)\} \end{aligned} \quad (5A.1)$$

Introduzindo-se uma notação semelhante àquela presente em (5.6) com o objetivo de se evidenciar a dependência do sinal codimodulado com as entradas e estados do codificador do código convolucional, (5A.1) passa a ser expressa como

$$\begin{aligned} R_{zz}(\tau) &= \sum_{p,q} E\{z[t-pT_m-t_\phi; (x^p, y^p)]z[t+\tau-qT_m-t_\phi; (x^q, y^q)]\} = \\ &= \sum_{p,r} E\{z[t-pT_m-t_\phi; (x^p, y^p)]z[t+\tau-(p+r)T_m-t_\phi; (x^{p+r}, y^{p+r})]\} = \end{aligned} \quad (5A.2)$$

onde $r=q-p$ é usado para referenciar os símbolos a serem correlacionados nos instantes p e q , com o símbolo no instante p tomado como referência.

A média a ser avaliada em (5A.2) se estende sobre as variáveis aleatórias x^p , y^p , x^{p+r} , y^{p+r} e t_ϕ para p e r fixos. Assim,

$$\begin{aligned} R_{zz}(\tau) &= \sum_{p,r} \sum_{u,v}^K \sum_{i,j}^L \frac{1}{T_m} \int_0^{T_m} dt_\phi z[t-pT_m-t_\phi; (x_u, y_i)] \cdot \\ &\quad \cdot z[t+\tau-(p+r)T_m-t_\phi; (x_v, y_j)] \text{Prob}\{ (x^m=x_u, y^m=y_i) \\ &\quad \cap (x^{p+r}=x_v, y^{p+r}=y_j) \} \end{aligned} \quad (5A.3)$$

onde a integração corresponde à média da variável aleatória contínua t_ϕ , suposta uniformemente distribuída entre $[0, T_m]$ e a probabilidade entre colchetes é como está expressa em (5.10). Assim,

$$\begin{aligned}
 R_{xx}(\tau) &= \sum_{p,r} \sum_{u,v}^K \sum_{i,j}^L \frac{1}{T_m} \int_0^{T_m} dt_\phi z[t-pT_m-t_\phi; (x_{i,u}, y_{i,v})] \\
 &\quad \cdot P_{r,u,v}(i,j) z[t+\tau-(p+r)T_m-t_\phi; (x_{i,j}, y_{i,v})] = \\
 &= \sum_{p,r} \frac{1}{T_m} \sum_{u,v=1}^K \int_0^{T_m} dt_\phi \left[\sum_{i,j=1}^L z[t-pT_m-t_\phi; (x_{i,u}, y_{i,v})] \right. \\
 &\quad \left. \cdot P_{r,u,v}(i,j) z[t+\tau-(p+r)T_m-t_\phi; (x_{i,j}, y_{i,v})] \right] = \\
 &= \sum_{p,r} \frac{1}{T_m} \sum_{u,v=1}^K \int_0^{T_m} dt_\phi A_{i,u}^T(t-pT_m-t_\phi) P_{r,u,v} A_{i,v}(t+\tau-(p+r)T_m-t_\phi).
 \end{aligned}
 \tag{5A.4}$$

Fazendo $t' = pT_m + t_\phi$, se obtém

$$\begin{aligned}
 R_{xx}(\tau) &= \sum_{p,r} \frac{1}{T_m} \sum_{u,v=1}^K \int_{pT_m}^{(p+1)T_m} dt_\phi A_{i,u}^T(t-t') P_{r,u,v} A_{i,v}(t+\tau-t'-rT_m) = \\
 &= \sum_r \frac{1}{T_m} \sum_{u,v=1}^K \int_{-\infty}^{\infty} dt_\phi A_{i,u}^T(t-t') P_{r,u,v} A_{i,v}(t+\tau-t'-rT_m).
 \end{aligned}$$

Este é o resultado desejado e que corresponde à Eq.

5.9.

APENDICE 5B

A transformada de Fourier da Eq. 5.9 é expressa por

$$\begin{aligned}
 W_{xx}(f) &= \int_{-\infty}^{\infty} r_{xx}(\tau) \exp(-j2\pi f\tau) d\tau = \\
 &= \int_{-\infty}^{\infty} d\tau \exp(-j2\pi f\tau) \left[\sum_r \frac{1}{T_{ms}} \sum_{u,v=1}^K \int_{-\infty}^{\infty} dt' A_{u,v}^T(t-t') P_{r,u,v} A_v(t+\tau-t'-rT_{ms}) \right].
 \end{aligned}
 \tag{5B.1}$$

Trocando-se a ordem de integração, (5B.1) passa a ser expressa por

$$\begin{aligned}
 &\int_{-\infty}^{\infty} dt' \left[\sum_r \frac{1}{T_{ms}} \sum_{u,v=1}^K A_{u,v}^T(t-t') P_{r,u,v} \left[\int_{-\infty}^{\infty} d\tau A_v(t+\tau-t'-rT_{ms}) \exp(-j2\pi f\tau) \right] \right] \\
 &= \int_{-\infty}^{\infty} dt' \left[\sum_r \frac{1}{T_{ms}} \sum_{u,v=1}^K A_{u,v}^T(t-t') P_{r,u,v} A_v(f) \exp[-2\pi f(t-t'+rT_{ms})] \right]
 \end{aligned}
 \tag{5B.2}$$

onde foi usado o par transformado (5.6) e (5.7).

Ordenando-se os termos em (5B.2), trocando-se a ordem dos somatórios e de integração, se obtém

$$\begin{aligned}
 W_{xx}(f) &= \frac{1}{T_{ms}} \sum_{u,v=1}^K \left[\int_{-\infty}^{\infty} dt' A_{u,v}^T(t-t') \exp[-2\pi f(t-t')] \right] \\
 &\cdot \left[\sum_r P_{r,u,v} \exp(-2\pi f r T_{ms}) \right] A_v(f) =
 \end{aligned}$$

$$= \frac{1}{T_m} \sum_{u,v=1}^K A_{u,v}^{**}(f) \left[\sum_r P_r^{u,v} \exp(-2\pi f r T_m) \right] A_v(f),$$

(5B.3)

que é o resultado desejado para a Eq. (5.16).

APENDICE 5C

A Eq. 5.16 pode ser escrita como

$$\begin{aligned} &= \frac{1}{T_m} \sum_{u,v=1}^K A_{u,v}^{**}(f) \left[\sum_r P_{\infty}^{u,v} \exp(-2\pi f r T_m) \right] A_v(f) + \\ &+ \frac{1}{T_m} \sum_{u,v=1}^K A_{u,v}^{**}(f) \left[\sum_r (P_r^{u,v} - P_{\infty}^{u,v}) \exp(-2\pi f r T_m) \right] A_v(f) = \\ &= \frac{1}{T_m} [W_{cl}(f) + W_{cr}(f)] \end{aligned} \quad (5C.1)$$

A matriz de correlação entre as palavras-código geradas por um codificador modelado segundo uma máquina de Mealy, de acordo com Cariolaro (1974), é expressa por

$$R(r) = \sum_{u,v=1}^K A_{u,v}^T P_r^{u,v} A_v \quad (5C.2)$$

onde $A_{u,v}$ e A_v são definidas de acordo com (2.17). Como $R(r) = R(-r)$, $P_r^{u,v} = P_{-r}^{u,v}$. Assim,

$$\begin{aligned}
 W_{cc}(f) = & \sum_{u,v=1}^K A_{u,v}^{**}(f) [P_{\phi}^{u,v} - P_{\infty}^{u,v} + \\
 & + 2\text{Re} \left[\sum_{r=1}^{\infty} (P_r^{u,v} - P_{\infty}^{u,v}) w^{-r} \right]] A_{\downarrow}(f)
 \end{aligned}
 \tag{5C.3}$$

onde $w = \exp(j2\pi f T_m)$. Substituindo (5.12) em (5C.3),

$$\begin{aligned}
 W_{cc}(f) = & \sum_{u,v=1}^K A_{u,v}^{**}(f) [q_{u,v} \delta_{u,v} \overset{\infty}{D} - q_{u,v} q_{\downarrow} \overset{\infty}{D} E_{u,v} + \\
 & + 2\text{Re} \left[q_{u,v} q_{\downarrow} \overset{\infty}{D} E_{u,v} \sum_{r=1}^{\infty} (\Pi^{r-1} - \Pi_{\infty}) w^{-r} \right]] A_{\downarrow}(f)
 \end{aligned}
 \tag{5C.4}$$

A suposição de que o sinal de entrada do codificador é estacionário no sentido fraco com independência entre vetores permite que o somatório sobre r presente em (5C.4), segundo Cariolaro (1974), resulte em

$$(U - \Pi_{\infty}) [wU - (\Pi - \Pi_{\infty})]^{-1} .
 \tag{5C.5}$$

Assim, substituindo (5C.5) em (5C.4) e após alguma manipulação algébrica, resulta a Eq. 5.18.

A partir de (5C.1), se obtém o componente discreto do espectro expresso por

$$W_{cd} = \sum_{u,v=1}^K A_{u,v}^{**}(f) \left[\sum_r P_{\infty}^{u,v} \exp(-2\pi f r T_m) \right] A_{\downarrow}(f)
 \tag{5C.6}$$

Similarmente, a substituição de (5.12c) em (5C.6) e após alguma manipulação algébrica, se obtém a Eq. (5.19).

CAPÍTULO 6

CÓDIGOS DE TRELIÇA PARA CONSTELAÇÕES MULTIDIMENSIONAIS

6.1 - Introdução

O procedimento de otimização apresentado no Cap. 4, em sua primeira etapa, teve como objetivo determinar os códigos de máxima distância livre, e conseqüentemente o maior ganho assintótico possível, para um código de treliça com uma complexidade de L estados. Entretanto, para qualquer esquema de codificação, existe um limite máximo para o ganho que o mesmo pode proporcionar, estabelecido através do Teorema do Shannon para a codificação. A construção de códigos que se aproximem ao máximo desse limite é o objetivo principal do projetista de sistemas de comunicações. Para os esquemas baseados nos códigos de treliça tratados nesse trabalho e mesmo para outros casos tais como os sistemas de modulação por fase contínua (Anderson e outros, 1981; Aulin e Sundberg, 1981), a aproximação do limite teórico de ganho é obtida aumentando-se o número de estados L da treliça associada ao código. Esses ganhos em geral são conseguidos usando-se no receptor um estimador de seqüências de máxima verossimilhança, cuja implementação se fundamenta no Algoritmo de Viterbi (AV). Conseqüentemente, a aproximação do limite teórico é conseguida ao preço de um aumento da complexidade do receptor.

A tendência à saturação existente na relação ganho versus complexidade (Ungerboeck, 1982; Marsen, 1984), impõe que se estabeleça um compromisso entre o binômio ganho-complexidade. Evidentemente, esse compromisso depende de fatores tais como tecnologia, custos econômicos, disponibilidade de recursos físicos, como por exemplo, energia e faixa de transmissão, e da aplicação a que se destina o sistema. Alguns sistemas dispõem de energia abundante mas trabalham com um espectro para alocação de faixa bastante congestionado. Noutros, a energia é o fator limitante, possuindo ainda alguma flexibilidade na alocação de faixa. Quanto à implementação, o desenvolvimento tecnológico de circuitos integrados com altíssima taxa de integração assegura implementações compactas de alta confiabilidade a custos que chegam a decrescer rapidamente ao longo do tempo.

O material a ser apresentado neste capítulo tem como principal objetivo estudar a viabilidade do uso de códigos de treliça que, para uma dada complexidade do decodificador, possibilitem uma maior eficiência no uso de energia, mesmo que ao preço de um aumento da faixa de transmissão. Tais códigos teriam aplicação em sistemas onde a energia disponível é limitada mas que apresentam flexibilidade na alocação de faixa. O ponto de partida do estudo consiste em se determinar a capacidade de canal para constelações multidimensionais. A comparação entre as capacidades de constelações multidimensionais e as de constela-

ções em uma ou duas dimensões, normalmente utilizadas em sistemas com modulação codificada, permitirá a avaliação do incremento da margem teórica de ganho. O segundo aspecto a ser abordado será a análise dos efeitos do aumento da dimensionalidade do espaço nas propriedades espectrais dos sinais, utilizando os resultados apresentados no Cap. 5. O terceiro e último aspecto, será a obtenção de códigos ótimos associados a uma dada constelação, usando os procedimentos apresentados no Cap. 4.

6.2 - Capacidade de canal para constelações multidimensionais

Seja uma constelação com K símbolos dispostos num espaço de sinais de dimensionalidade M . Usando-se uma base ortonormal, cada símbolo tem representação temporal

$$x_i(t) = \sum_{j=1}^M p_{i,j} \varphi_j(t), \quad 0 \leq t \leq T_m, \quad i=1,2, \dots, K, \quad (6.1)$$

e é especificado completamente pelo vetor

$$x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,M}), \quad i=1,2, \dots, K \quad (6.2)$$

onde $T_m = kT_s$ é o intervalo de modulação, k é o número de elementos q -ários associados a cada símbolo e T_s é o intervalo de tempo associado a cada um desses elementos, isto é, $K = q^k$.

Os K símbolos caracterizados através de (6.2) formam o conjunto de entrada, X , associado ao canal. Cada símbolo ao ser transmitido através do canal é corrompido pelo ruído $n(t)$, suposto aditivo, de modo que o sinal recebido, $y(t)$, é um ponto qualquer do espaço de sinais e especificado através do vetor

$$y = (y_1, y_2, \dots, y_M). \quad (6.3)$$

Decompondo-se o ruído de canal ao longo dos eixos ortogonais caracterizados pela base presente em (6.1), pode-se escrever

$$n(t) = \sum_{j=1}^M n_j \varphi_j(t), \quad 0 \leq t \leq T_m. \quad (6.4)$$

Como o ruído é suposto aditivo,

$$y(t) = x_i(t) + n(t), \quad (6.5)$$

ou ainda

$$y = x_i + n, \quad (6.6)$$

onde $n = (n_1, n_2, \dots, n_M)$ é o vetor de ruído. Assim, pelo efeito do ruído, cada símbolo x_i pode ser mapeado a qualquer ponto do espaço de sinais Y , o conjunto de saída do canal. Observe-se que o conjunto de entrada é finito enquanto que o de saída é infinito. Portanto, o problema da avaliação da capacidade de canal para constelações com K símbolos é um caso particular do problema da avaliação da capacidade de canal para espaços contínuos (Gallager, 1965, Cap. 7 e 8). Isto é, a capacidade por intervalo de modulação é dada por

$$C(T_m) = \text{Max}_{\{Q(x_i)\}} I(X, Y), \quad \text{bits}/T_m \quad (6.7)$$

onde $I(X, Y)$ é a informação mútua entre os conjuntos X e Y e $\{Q(x_i)\}$ é o conjunto de todas as possíveis distribuições de probabilidade sobre os elementos do conjunto de entrada. Porém,

$$I(X, Y) = \sum_{i=1}^M Q(x_i) \int_y p(y/x_i) I(x_i, y) dy \quad (6.8)$$

onde $I(x_i, y)$ é a informação mútua entre x_i e y , expressa por

$$I(x_i, y) = \log_2 \frac{p(y/x_i)}{\sum_{j=1}^M Q(x_j) p(y/x_j)} \quad (6.9)$$

e $p(y/x_i)$ são as funções densidade de probabilidade condicionadas a cada x_i ($i=1, 2, \dots, K$) e que caracterizam estatisticamente o canal. Para canais com ruído gaussiano branco e aditivo,

$$p(y/x_i) = p_n(y - x_i), \quad (6.10)$$

onde p_n é a função densidade de probabilidade do vetor n na Eq. 6.6, cujos componentes são variáveis aleatórias gaussianas univariadas de média zero e variância σ^2 , na hipótese de ruído gaussiano (Wozencraft, 1965, Cap. 4, 5).

Tendo em vista as hipóteses assumidas anteriormente em relação às constelações código, somente a condição de símbolos equiprováveis tem significado. Assim, substituindo (6.10) em (6.8) e (6.9) e usando-se essa condição de equiprobabilidade, (6.7) resulta em

$$C(T_m) = \frac{1}{K} \sum_{i=1}^K \int_n p_n(y-x_i) \log_{2^m} \frac{p_n(y-x_i)}{\frac{1}{M} \sum_{j=1}^M p_n(y-x_j)} dn \quad (6.11)$$

Após algum algebrismo, (6.11) passa a ser expressa como

$$C(T_m) = \log_{2^m} K - \frac{1}{K} \sum_{i=1}^K E \left\{ \log_{2^m} \frac{p_n(y-x_i)}{\sum_{j=1}^M p_n(y-x_j)} \right\} \quad (6.12)$$

onde $E\{ \}$ significa média estatística. Note que (6.12) independe do tipo de ruído de canal, respeitado o modelo de ruído aditivo. No caso particular de ruído gaussiano, a capacidade de canal passa a ser expressa por

$$C(T_m) = \log_{2^m} K - \frac{1}{K} \sum_{i=1}^K E \left\{ \log_{2^m} \sum_{j=1}^M \exp[-(|x_i+n-x_j|^2 - |n|^2/2^m)] \right\} \quad (6.13)$$

A complexidade numérica envolvida na avaliação de (6.13) exige a utilização de métodos numéricos apropriados. Em (Bezerra e Arantes, 1985), foram desenvolvidos dois procedimentos. O primeiro, denominado de direto, avalia as K integrais de multiplicidade M utilizando quadratura de Gauss-Hermite e só é apropriado para $M < 5$. O segundo, denominado de indireto, usa técnicas de Monte Carlo e explora o fato de (6.13) ser uma soma

de médias de funções de variáveis aleatórias gaussianas. Este último procedimento se aplica para qualquer M e se mostrou mais rápido quando comparado com o método direto, mantendo uma precisão aceitável ao tipo de problema.

6.3 - Avaliação da margem de ganho teórico

A avaliação da margem de ganho teórica em relação a duas constelações de sinais, uma tomada como referência e a outra como constelação código, é feita com base no cálculo da capacidade de canal para ambas as constelações. Essa margem de ganho pode ser estabelecida como segue.

Seja SNR_m a razão sinal/ruído, em dB, necessária à transmissão dos símbolos da constelação de referência, a uma taxa de $k \log_2 q$ bits por intervalo de modulação, e que assegura ao sistema uma taxa de erros de transmissão igual ou inferior a um limite previamente estabelecido no sistema. Seja SNR_b a razão sinal ruído, em dB, para a qual é possível a mesma taxa de bits usando-se uma constelação código com $N > K$ símbolos. Então, a margem de ganho proporcionada pela constelação de N símbolos é expressa por

$$G_{max} = SNR_m - SNR_b \quad (6.14)$$

A razão sinal/ruído para um espaço de sinais de dimensionalidade M com ruído gaussiano de média zero e variância σ^2 por dimensão, é definida por

$$SNR = E_m / (M \sigma^2) \quad (6.15)$$

onde E_m é a energia média da constelação. A razão sinal/ruído em dB é expressa por

$$SNR = 10 \log [E_m / (M \sigma^2)] \quad (dB) \quad (6.16)$$

o que resulta em

$$\sigma^2 = \frac{1}{M} [10^{-0,1 SNR(dB)}] \quad (6.17)$$

quando a energia média da constelação é normalizada em $E_m=1$. A substituição de (6.17) em (6.13) possibilita a obtenção da capacidade de canal em função da razão sinal/ruído em dB, supondo $E_m=1$.

A dependência da razão sinal/ruído com a dimensionalidade do espaço de sinais, impõe que ao serem comparadas capacidades de canal para constelações com diferentes dimensionalidades, sejam tomadas precauções que assegurem uma comparação correta. Assim, se SNR_{M_2} é a razão sinal/ruído para uma constelação de dimensionalidade M_2 , a mesma condição de ruído por dimensão é obtida, numa constelação de dimensionalidade M_1 , para uma relação sinal ruído SNR_1 , expressa por

$$SNR_1(\text{dB}) = SNR_{M_2}(\text{dB}) + (M_2/M_1)\text{dB} . \quad (6.18)$$

A Eq. (6.18) simplesmente impõe que ao serem plotadas num mesmo sistema de coordenadas capacidades de canal de constelações com diferentes dimensionalidades e a constelação de dimensionalidade M_2 é tomada como referência, a curva original correspondente à constelação de dimensionalidade M_1 deve ser adiantada de $(M_2/M_1)\text{dB}$. Este é um aspecto inconveniente já que quase sempre a segunda parcela no lado direito de (6.18) é um valor decimal que não guarda relação com as abscissas utilizadas na obtenção das capacidades para cada constelação individualmente. Uma forma de evitar essa dificuldade, consiste em considerar todas as constelações dispostas no espaço de maior dimensionalidade, introduzindo-se coordenadas nulas nas correspondentes posições da representação de cada símbolo das constelações de dimensionalidades menores. Neste caso, o único inconveniente será o maior tempo a ser gasto na avaliação das capacidades das constelações com dimensionalidades menores.

6.4 - Margem de ganho para o sistema OCTAL CÚBICO relativo ao sistema BPSK

As margens de ganho possíveis de serem obtidas usando-se constelações em espaços de uma e duas dimensões (M-AM, M-PSK, M-QAM, M-QASK, etc) já são bem estabelecidas na literatura (Ungerboeck, 1982; Forney e outros, 1984; Marsen e outros, 1984). Por razões a serem apresentadas posteriormente, há o interesse em se determinar qual o incremento na margem de ganho proporcionado por uma constelação tridimensional com $K=8$ símbolos distribuídos nos vértices de um cubo (OCTAL CÚBICO), em comparação com constelações bidimensionais com o mesmo número de símbolos (8-PSK, 8-AMPM). A comparação será feita considerando-se as constelações 2-PSK e 4-PSK como constelações de referência. A Fig. 6.1 ilustra essas constelações, todas supostas com energia média unitária.

Usando-se o método indireto (Bezerra e Arantes, 1986a) para avaliação de (6.13), foram obtidas as respectivas capacidades de canal para as constelações OCTAL CÚBICO, 8-PSK, 4-PSK e 2-PSK. A Fig. 6.2 mostra os resultados obtidos, além da capacidade para o canal gaussiano tridimensional que estabelece o limite máximo para a capacidade de canal (Gallager, 1965, Cap. 7,

B). Observe que o sistema OCTAL CÚBICO possibilita um incremento na margem de ganho de $\approx 1,7$ dB em relação ao 8-PSK, considerando-se o 4-PSK como referência. Esse incremento decresce para $\approx 0,7$ dB se o sistema de referência for o 2-PSK.

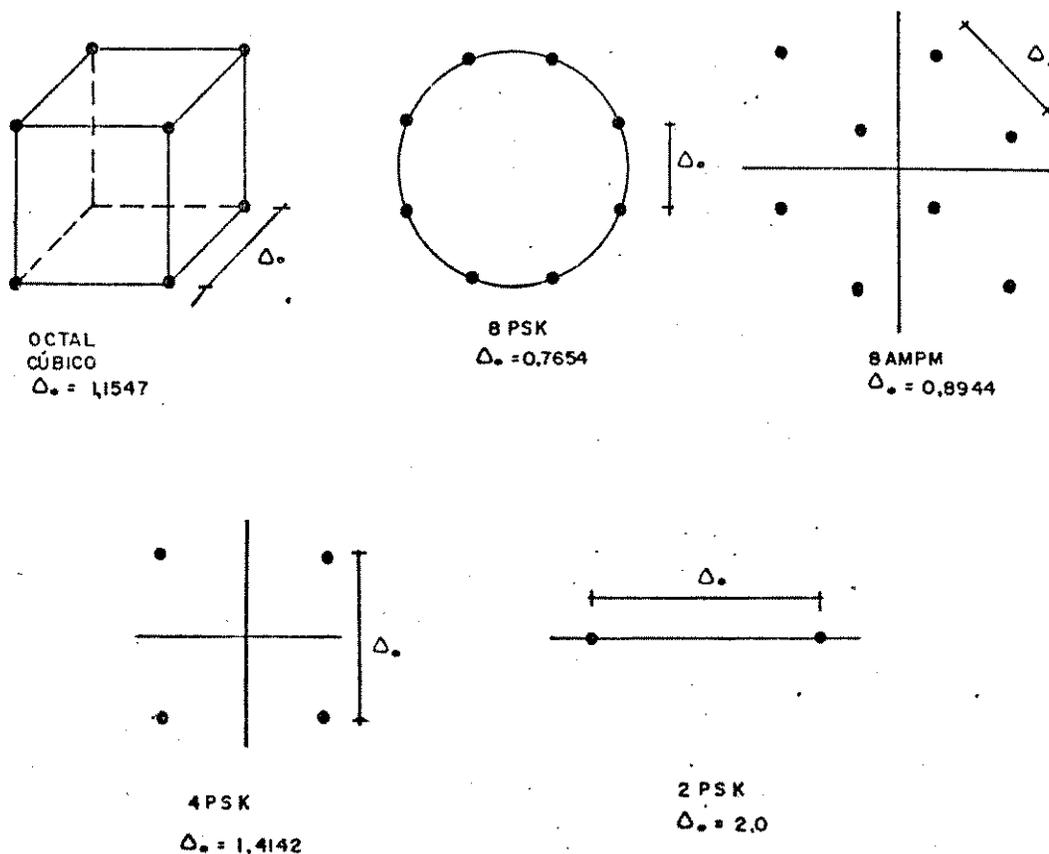


Fig. 6.1: Constelações para alguns esquemas de modulação digital, supostas com energia média unitária ($E_m=1$).

6.5 - Margem de ganho para a constelação OCTAL CÚBICO relativo ao OCTAL PRISMÁTICO

As constelações M-PSK apresentam uma característica importante quando a transmissão digital tiver que ser realizada através de canais não lineares. Essa característica é a envoltória constante para o sinal gerado pelos símbolos pertencentes às constelações. Como será visto na próxima seção, no espaço tridimensional, para a base utilizada neste trabalho, somente duas constelações apresentam a propriedade de gerar sinais com envoltória constante, a constelação OCTAL CÚBICO e a constelação

aqui denominada de OCTAL PRISMÁTICO. Como o nome sugere, a constelação OCTAL PRISMÁTICO é constituída por $K=8$ símbolos dispostos nos vértices de um prisma. Entretanto, o prisma a ser considerado é tal que suas faces, superior e inferior, tangenciem a superfície externa de uma esfera de raio R . Além disto, as dimensões das arestas destas faces devem ser idênticas às arestas do cubo inscrito na esfera de raio R . Exceto o prisma inscrito na esfera com base quadrada de lado R , esta é a constelação prismática que apresenta a menor energia média, $E_m=5R^2/3$, com a maior distância mínima entre seus pontos. Normalizando a energia média em $E_m=1$, o raio da esfera é $R \approx 0,771$, resultando numa distância mínima de $0,8944$.

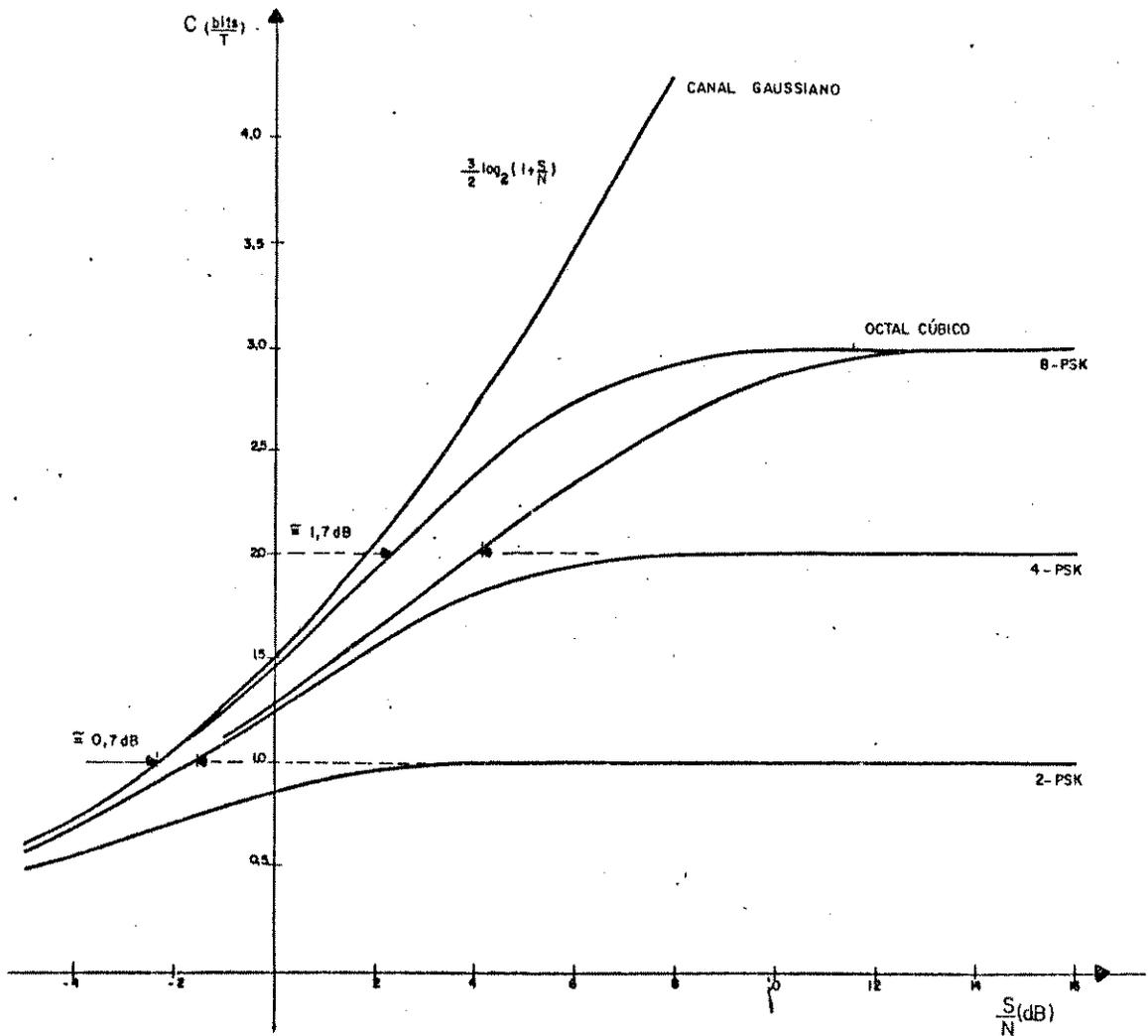


Fig. 6.2: Capacidade de canal para as constelações OCTAL CÚBICO, 8-PSK, 4-PSK, 2-PSK e canal gaussiano.

A Fig. 6.3 ilustra a constelação prismática, enquanto que a Fig. 6.4 mostra as capacidades de canal para as constelações cúbica e prismática, como também para a constelação 8-AMPM. Como se pode observar, a constelação octal prismática apresenta um decréscimo de $\approx 0,2$ dB e $\approx 0,8$ dB, em relação à constelação cúbica, para transmissão de 1,0 e 2,0 bits por intervalo de modulação, respectivamente. Quanto ao esquema 8-AMPM, o desempenho é $\approx 0,2$ dB inferior à constelação prismática. Isto é, o esquema 8-AMPM é potencialmente superior ao 8-PSK em aproximadamente 0,7 dB (veja Fig. 6.2). Infelizmente, o 8-AMPM não gera sinais com envoltória constante e assim está limitado ao uso em canais lineares.

6.6 - Caracterização temporal e espectral para constelações em até tres dimensões

6.6.1 - Constelações bidimensionais

Os símbolos pertencentes a uma constelação bidimensional em geral são caracterizados através da base ortonormal

$$\varphi_1(t) = \sqrt{\frac{2}{T_m}} \cos(\omega_c t)$$

$$0 \leq t \leq T_m$$

$$\varphi_2(t) = \sqrt{\frac{2}{T_m}} \sin(\omega_c t)$$

(6.19)

cuja representação no domínio da frequência é expressa por

$$\Phi_1(f) = \sqrt{\frac{T_m}{2}} \{ \text{Sinc}[(f-f_c)T_m] + \text{Sinc}[(f+f_c)T_m] \} w$$

$$\Phi_2(f) = j \sqrt{\frac{T_m}{2}} \{ \text{Sinc}[(f-f_c)T_m] - \text{Sinc}[(f+f_c)T_m] \} w$$

(6.20)

onde $\text{Sinc}(x) = \text{sen}(\pi x) / (\pi x)$ e $w = \exp(-j \pi f T_m)$. Assim, cada símbolo de coordenadas $(x_{1,1}, x_{1,2})$ pertencente a uma constelação com K elementos, tem representação temporal

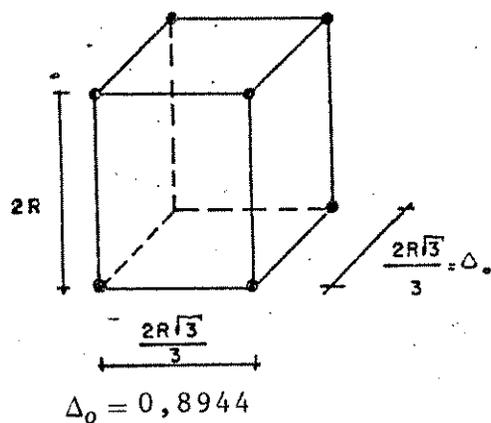


Fig. 6.3: Constelação OCTAL PRISMÁTICA.

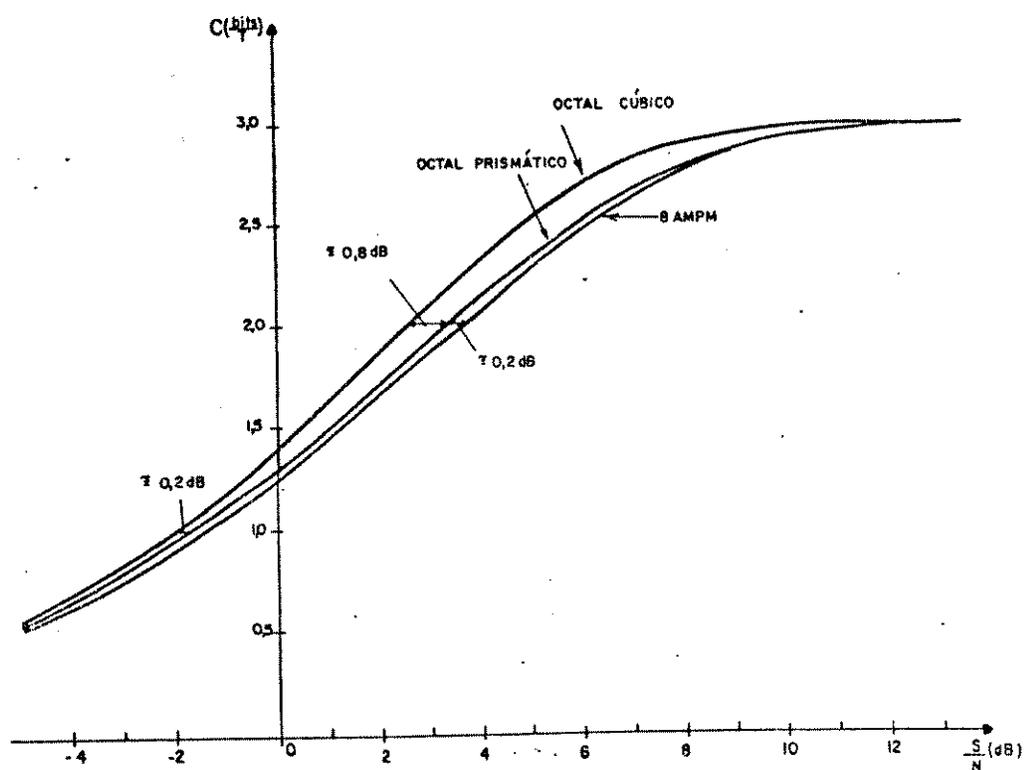


Fig. 6.4: Capacidade de canal para as constelações OCTAL CÚBICA, OCTAL PRISMÁTICA E 8-AMPM.

$$x_i(t) = x_{i,1} \varphi_1(t) + x_{i,2} \varphi_2(t) \quad i=1,2, \dots, K, \quad (6.21a)$$

e representação espectral

$$X_i(f) = x_{i,1} \Phi_1(f) + x_{i,2} \Phi_2(f) . \quad (6.21b)$$

A densidade espectral de energia associada a cada símbolo é então expressa por

$$|X_i(f)|^2 = x_{i,1}^2 |\Phi_1(f)|^2 + x_{i,2}^2 |\Phi_2(f)|^2 . \quad (6.22)$$

Portanto, utilizando a condição de ortonormalidade da base, a energia de cada símbolo é expressa por

$$E_i = \int_{-\infty}^{\infty} |X_i(f)|^2 df = x_{i,1}^2 + x_{i,2}^2 . \quad (6.23)$$

Entretanto, é conveniente uma representação em banda base para a caracterização espectral de cada símbolo de modo a manter sua energia inalterada. Essa condição é obtida se a representação em banda base for definida por

$$X_i(f)_{BB} = \begin{cases} \sqrt{2} X_i(f+f_c) & ; f > -f_c \\ 0 & ; f < -f_c \end{cases} \quad (6.24)$$

onde f_c é suposto muito maior que a faixa de interesse do sinal.

A utilização de (6.20), (6.21) e (6.24) resulta em

$$X_i(f)_{BB} = \sqrt{T_m} [x_{i,1} + jx_{i,2}] \text{Sinc}(fT_m) \exp[-j\pi(f+f_c)T_m] \quad (6.25)$$

ou ainda

$$|X_i|_{BB}^2 = T_m E_i \text{Sinc}^2(fT_m) , \quad (6.26)$$

que assegura a conservação da energia do símbolo na representação em banda base.

6.6.2 - Constelações tridimensionais

A caracterização temporal e espectral para constelações tridimensionais obedece aos mesmos princípios utilizados na caracterização das constelações bidimensionais. Entretanto, existe uma pequena dificuldade associada à escolha da base geradora do espaço de sinais já que, como visto acima, as propriedades espectrais da constelação dependem apenas das propriedades espectrais das funções constituintes da base.

Observe que para o caso bidimensional as funções da base apresentam uma envoltória da forma

$$e(t) = \sqrt{\frac{2}{T_m}} \operatorname{ret}[(t - 0,5T_m)/T_m] \quad (6.27)$$

onde

$$\operatorname{ret}(t/T_m) = \begin{cases} 1 & ; & |t| \leq 0,5T_m \\ 0 & ; & |t| > 0,5T_m \end{cases} \quad (6.28)$$

A Transformada de Fourier (TF) de (6.28) é quem estabelece as propriedades espectrais das funções base em (6.19). Idealmente, para que uma constelação tridimensional apresente as mesmas características espectrais de uma constelação bidimensional, a envoltória das três funções da base deve ser da forma expressa em (6.28). Infelizmente, isto não é possível já que a largura da faixa de transmissão está relacionada com a dimensionalidade do espaço de sinais (Gallager, 1965, Cap. 7, 8). Um outro aspecto a ser levado em consideração, na escolha das funções da base, é a capacidade das mesmas possibilitarem a geração de sinais com envoltória constante que, como já foi dito anteriormente, são apropriados à transmissão em canais não lineares.

A consideração dos aspectos acima mencionados aliado ao compromisso de tentar minimizar a expansão de faixa levou à escolha da seguinte base para a caracterização de constelações em espaços tridimensionais:

$$\varphi_1(t) = \sqrt{\frac{2}{T_m}} \cos(\omega_c t) \quad 0 \leq t \leq T_m$$

$$\varphi_2(t) = \begin{cases} \sqrt{\frac{4}{T_m}} \operatorname{sen}(\omega_c t) & ; & 0 \leq t \leq 0,5T_m \\ 0 & ; & 0,5T_m < t \leq T_m \end{cases}$$

$$\varphi_{\alpha}(t) = \begin{cases} 0 & 0 \leq t < 0,5T_m \\ \sqrt{\frac{4}{T_m}} \operatorname{sen}(w_{\alpha}t) & 0,5T_m \leq t \leq T_m \end{cases} \quad (6.29)$$

onde é assumido que $w_{\alpha} \gg 2\pi/T_m$. Com essa suposição, não é difícil se verificar que a condição de ortonormalidade é satisfeita. Isto é,

$$\int_{-\infty}^{\infty} \varphi_i(t) \varphi_j(t) dt = \begin{cases} 1 & ; \quad i = j \\ 0 & ; \quad i \neq j \end{cases} \quad i, j = 1, 2, 3. \quad (6.30)$$

Note que na base definida em (6.29) uma das funções mantém as propriedades espectrais da base definida em (6.19) enquanto as outras apresentam uma expansão de faixa de um fator dois. Isto é, a representação espectral de (6.29) é expressa, respectivamente, por

$$\begin{aligned} \Phi_1(f) &= \sqrt{\frac{T_m}{2}} \{ \operatorname{Sinc}[(f-f_{\alpha})T_m] + \operatorname{Sinc}[(f+f_{\alpha})T_m] \} w \\ \Phi_2(f) &= j\sqrt{\frac{T_m}{4}} \{ \operatorname{Sinc}[(f-f_{\alpha})T_m/2] - \operatorname{Sinc}[(f+f_{\alpha})T_m/2] \} w^{0,5} \\ \Phi_3(f) &= j\sqrt{\frac{T_m}{4}} \{ \operatorname{Sinc}[(f-f_{\alpha})T_m/2] - \operatorname{Sinc}[(f+f_{\alpha})T_m/2] \} w^{1,5} \end{aligned} \quad (6.31)$$

onde $w = -j\pi f T_m$.

Agora, cada símbolo de coordenadas ($x_{1,1}$, $x_{1,2}$, $x_{1,3}$) pertencente a uma constelação com K pontos é expresso no domínio do tempo por

$$x_1(t) = \begin{cases} \sqrt{\frac{2}{T_m}} [x_{1,1} \cos(w_{\alpha}t) + 2 x_{1,2} \operatorname{sen}(w_{\alpha}t)] & ; \quad 0 \leq t \leq T_m/2 \\ \sqrt{\frac{2}{T_m}} [x_{1,1} \cos(w_{\alpha}t) + 2 x_{1,3} \operatorname{sen}(w_{\alpha}t)] & ; \quad T_m/2 < t \leq T_m \end{cases} \quad (6.32)$$

Reescrevendo (6.32) de modo que a envoltória do símbolo seja explicitada, se obtém

$$x_i(t) = \begin{cases} \sqrt{\frac{2}{T_m}} (x_{i,1} e^{j\omega_c t} + \sqrt{2} x_{i,2} e^{j(\omega_c t + \theta_1)})^{1/2} \cos(\omega_c t + \theta_1), & 0 \leq t \leq T_m/2 \\ \sqrt{\frac{2}{T_m}} (x_{i,1} e^{j\omega_c t} + \sqrt{2} x_{i,3} e^{j(\omega_c t + \theta_2)})^{1/2} \cos(\omega_c t + \theta_2), & T_m/2 < t \leq T_m \end{cases} \quad (6.33)$$

onde $\theta_1 = \text{tg}^{-1}(\sqrt{2} x_{i,2}/x_{i,1})$ e $\theta_2 = \text{tg}^{-1}(\sqrt{2} x_{i,3}/x_{i,1})$.

A partir de (6.33) se conclui que a envoltória de cada símbolo de uma constelação tridimensional, para a base definida em (6.29), só tem envoltória constante se for satisfeita a seguinte condição:

$$|x_{i,2}| = |x_{i,3}| \quad (6.34)$$

Similarmente, a envoltória do sinal gerado por uma constelação tridimensional, para a base definida em (6.29), só será constante se forem satisfeitas uma das seguintes condições:

$$|x_{i,j}| = |x_{i,1}| \quad i=1,2, \dots, K : j, i=1,2,3 \quad (6.35a)$$

ou

$$\begin{cases} |x_{i,1}| = |x_{j,1}| & i, j=1,2, \dots, K, \\ |x_{i,2}| = |x_{i,3}| \neq |x_{i,1}|, & i=1,2, \dots, K. \end{cases} \quad (6.35b)$$

É evidente que somente a constelação cúbica é capaz de satisfazer à condição (6.35a) enquanto que a constelação prismática satisfaz à condição (6.35b).

A caracterização espectral para um símbolo numa constelação tridimensional é expressa por

$$X_i(f) = x_{i,1} \Phi_1(f) + x_{i,2} \Phi_2(f) + x_{i,3} \Phi_3(f) \quad (6.36)$$

e sua densidade espectral de energia por

$$|X_1(f)|^2 = x_{1,1}^2 |\Phi_1(f)|^2 + x_{1,2}^2 |\Phi_2(f)|^2 + x_{1,3}^2 |\Phi_3(f)|^2. \quad (6.37)$$

A ortonormalidade da base assegura que, através da integração de (6.37), a energia do símbolo seja expressa por

$$E_1 = x_{1,1}^2 + x_{1,2}^2 + x_{1,3}^2 \quad (6.38)$$

A densidade espectral de energia em banda base é obtida da mesma forma que no caso bidimensional, resultando em

$$|X_1(f)_{BB}|^2 = T_m [x_{1,1}^2 \text{Sinc}^2(fT_m) + 0,5(x_{1,2}^2 + x_{1,3}^2) \text{Sinc}^2(fT_m/2)] \\ l=1,2, \dots K. \quad (6.39)$$

A substituição de (6.39) em (5.22) e (5.23) possibilita o cálculo do espectro de potência em banda base para constelações tridimensionais.

Finalizando esta seção, resta uma observação a respeito da definição para a representação em banda base expressa por (6.24) e que difere pelo fator $\sqrt{2}$ na forma como usualmente é definida. Como se observou, a finalidade desse fator foi manter inalterada a energia do símbolo na passagem para a representação em banda base e conseqüentemente o mesmo ocorreu com a energia média da constelação. Assim, para uma constelação de energia média unitária, a potência fora da faixa é expressa por

$$P = 1 - 2 \int_0^B W(f)_{BB} df \quad (6.40)$$

onde B é a largura de faixa utilizada e $W(f)_{BB}$ é o espectro de potência em banda base, avaliado também para a condição de energia média unitária. O cálculo da potência fora da faixa será utilizado como critério de medida do efeito do aumento da dimensionalidade na faixa de transmissão.

6.7 - Espectro para as constelações OCTAL CÚBICA, OCTAL PRISMÁTICA e 8-PSK

O procedimento desenvolvido no Cap. 5 para o cálculo do espectro de sinais codimodulados, segundo o controle de um código convolucional, foi implementado num programa para computador digital. Usando-se os mesmos princípios que resultaram nas Eqs.

5.22 e 5.23, o mesmo programa pode ser utilizado para o cálculo do espectro de sinais digitais com modulação digital convencional. Para o caso específico de constelações com $K=8$ símbolos, o código convolucional caracterizado pela matriz

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.41)$$

e pelos parâmetros $k=n=3$, $q=p=2$, $\mu=0$, $k''=3$ e $m=1$, será utilizado para ilustrar as propriedades espectrais das constelações CÚBICA, PRISMÁTICA E 8-PSK.

A Fig. 6.5 ilustra os resultados obtidos para as constelações OCTAL CÚBICA, OCTAL PRISMÁTICA e 8-PSK, todas supostas com energia média unitária (o programa se encarrega de fazer a normalização). A Fig. 6.6 ilustra a potência fora da faixa, em porcentagem, para as mesmas constelações. Como se pode observar, o efeito do aumento da dimensionalidade na largura da faixa de transmissão depende do modo como a mesma é definida. Por exemplo, se o critério usado for a frequência B tal que $BT_w=1/3$ ($BT_w=1$), a perda de potência dos sistemas prismático e cúbico em relação ao 8-PSK é de aproximadamente 5% e 8%, respectivamente. Por outro lado, se o critério usado estabelecer que 90% da potência deve estar contida na faixa de transmissão, para o 8-PSK se obtém $BT_w \approx 0,33$, para o prismático $BT_w \approx 0,44$ e para o cúbico $BT_w \approx 0,5$, o que representa uma aumento de faixa de 33% e 52%, respectivamente, em relação à faixa do 8-PSK.

6.8 - Códigos com máxima distância livre para as constelações 8-PSK e OCTAL CÚBICO

O modelo desenvolvido ao longo dos Caps. 2, 3 e 4 possibilitou a implementação de um programa que faz a seleção dos códigos de máxima distância livre associados a uma constelação código com N símbolos, uma constelação de referência com K símbolos e uma treliça com L estados. A estrutura básica do programa segue o esquema apresentado na Fig. 4.1, a menos do fato de serem consideradas todas as possíveis estruturas de treliça de L estados para cada matriz D .

As características de ganho versus complexidade para o 4-PSK codificado como BPSK foram estabelecidas por Ungerboeck (1982) e estão apresentadas na Tabela 6.1 para treliças com $N_r=1$ e $N_r=2$ ramos por transição. Como se pode observar, mesmo o ganho aumentando cada vez que a complexidade (número de estados) é dobrada, o maior incremento (0,6 dB) é obtido ao se passar de uma treliça com 4 estados (2 ramos paralelos por transição) para uma outra de 8 estados (menor complexidade para as treliças com um

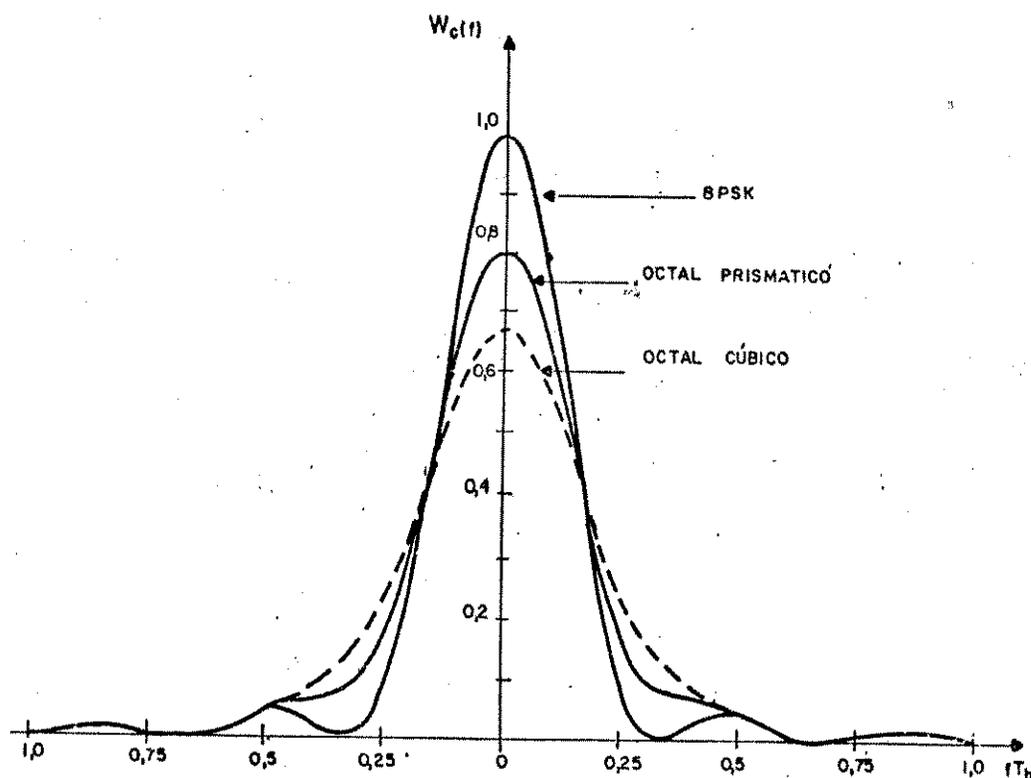


Fig. 6.5: Espectro em banda base para as constelações 8-PSK, OCTAL CÚBICA e OCTAL PRISMÁTICA em função da frequência normalizada fT_b , onde T_b é o período de bit e T_m suposto unitário.

ramo por transição). Ainda se observa, para os casos em que $N_r = 1$, que um aumento no ganho de 58% (3,6 - 5,7 dB) implica num aumento de 64 vezes (8 - 512 estados) na complexidade do sistema. Esses resultados mostram que existe um preço a ser pago na tentativa de se obter códigos com ganhos que se aproximam da margem de ganho teórico. Assim, na busca de códigos de máxima distância livre para as constelações 8-PSK e OCTAL CÚBICO, usadas para transmissão de 4PSK, serão consideradas apenas treliças com 1 ramo por transição e com complexidade mínima, $L=8$ (vide Tab. 6.1). Para estas condições, os parâmetros a serem considerados para os códigos são: $k=2$, $k''=0$, $n=3$, $\mu=3$, $m=3$ e $q=p=2$. A Fig. 6.7 ilustra as estruturas das treliças correspondentes às configurações de memórias possíveis para o codificador, enumeradas abaixo.

$$M = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \quad M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad (6.42a)$$

$$M = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \quad M = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (6.42b)$$

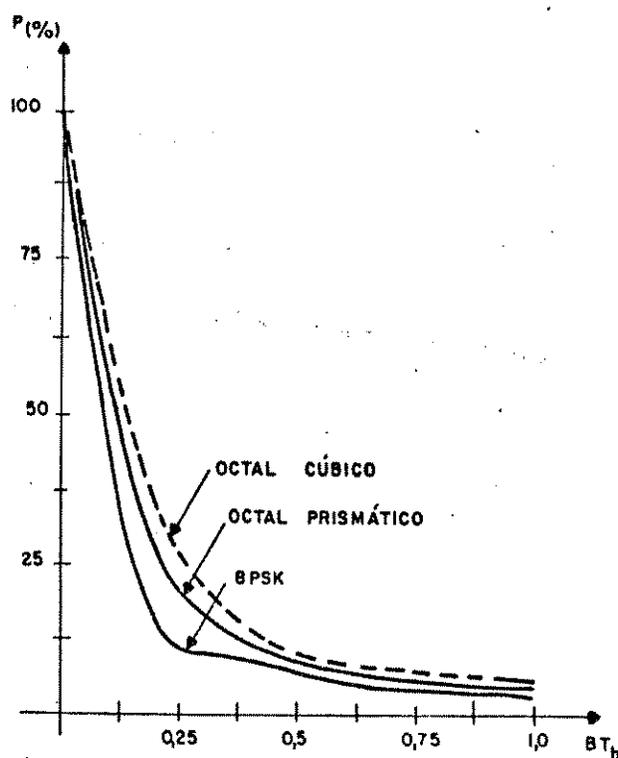


Fig. 6.6: Potência fora da faixa para os espectros da Fig. 6.5.

Tabela 6.1 : Ganho versus complexidade para codificação do 4-PSK no 8-PSK.

N_p	L	4	8	16	32	64	128	256	512
2		3,0	-	-	-	-	-	-	-
1	dB	-	3,6	4,1	4,6	4,8	5,0	5,4	5,7

A Fig. 6.8 ilustra o resultado do MPC aplicado às duas constelações consideradas. Através do procedimento apresentado no Cap. 4 (Sec. 4.3), as matrizes D , para ambas constelações, correspondem às mesmas do Ex. 4.3. Considerando-se a representação decimal para as colunas, as mesmas são especificadas por

$$\begin{aligned} D &= [4 \ 2] & D &= [2 \ 4] \\ D &= [2 \ 6] & D &= [6 \ 2] \\ D &= [6 \ 4] & D &= [4 \ 6] \end{aligned} \quad (6.43a)$$

Aplicando-se a equivalência de códigos, é suficiente serem consideradas as matrizes

$$D = [4 \ 2] \quad D = [2 \ 6] \quad D = [6 \ 4]. \quad (6.43b)$$

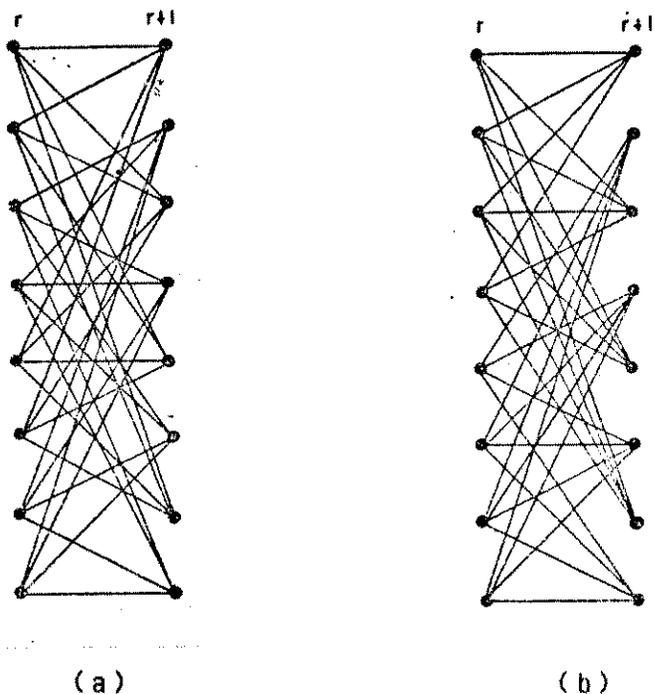


Fig. 6.7: Estruturas para as treliças associadas às configurações de memórias em (6.42). (a) corresponde a (6.42a) e (b) a (6.42b).

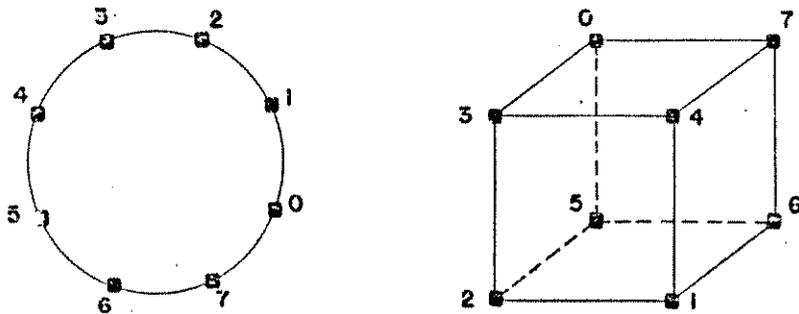


Fig. 6.8: MPC para as constelações 8-PSK e OCTAL CÚBICO, supostas com $E_m=1$. Os subconjuntos da primeira subpartição são $\{0, 2, 4, 6\}$ e $\{1, 3, 5, 7\}$.

As Tabelas 6.2 e 6.3 apresentam os códigos de máxima distância livre para as constelações 8-PSK e OCTAL CÚBICO, respectivamente, e são o resultado do procedimento de otimização ilustrado na Fig. 4.1. Da mesma forma que em (6.43), as matrizes C estão especificadas através da representação decimal de suas colunas. As máximas distâncias livres obtidas são 2,1414 para o 8PSK e 2,3094 para o OCTAL CÚBICO. Assim, os ganhos proporcionados, relativamente ao 4-PSK, são

$$G(8PSK) = 20 \log(2,1414/1,4142) = 3,6 \text{ dB} \quad (6.44a)$$

$$G(OCTAL \text{ CÚBICO}) = 20 \log(2,3094/1,4142) = 4,3 \text{ dB} \quad (6.44b)$$

É oportuno se observar que tanto para o 8-PSK quanto para o OCTAL CÚBICO, quando são considerados códigos com $N_r=2$, a máxima distância livre é limitada superiormente por 2,0 (a distância mínima entre os elementos da partição de ordem 2 no MPC da Fig. 6.8), isto é, um ganho máximo de 3,0 dB. Assim, ao serem consideradas treliças com um ramo por transição e número mínimo de estados, o sistema OCTAL CÚBICO fornece um incremento no ganho de 1,3 dB, enquanto para o 8PSK esse incremento é de 0,6 dB. Note também que o esquema 8-PSK só proporciona ganhos acima de 4,3 dB para uma complexidade igual ou superior a 32 estados (veja Tab. 6.1). Portanto, o esquema OCTAL CÚBICO proporciona uma considerável redução na complexidade do decodificador em relação ao 8-

PSK, para ganhos da ordem de 4,3 dB. Como acontece para os esquemas tridimensionais sem codificação, o preço a ser pago é um aumento na faixa de transmissão. A Fig. 6.9 mostra os espectros de potência para o 4-PSK codificado como 8-PSK e OCTAL CÚBICO e na Fig. 6.10 estão ilustradas as respectivas potências fora da faixa.

6.9 - Otimização dos códigos de máxima distância livre para 4-PSK codificado em 8-PSK e OCTAL CÚBICO

Os códigos listados nas Tabs. 6.2 e 6.3 são ótimos no sentido de que apresentam os mesmos ganhos assintóticos de 3,6 dB e 4,3 dB, respectivamente. A seleção dos códigos ótimos para cada configuração de memórias M e para cada matriz D é feita através do procedimento de otimização apresentado no apêndice A4. As Tabs. 6.4 e 6.5 apresentam os códigos ótimos selecionados pelo referido procedimento. Esses resultados foram obtidos considerando-se como seqüência de referência a seqüência de estados $(0\ 0\ 0 \dots 0)$ com um comprimento correspondente a $\delta = 6\mu = 18$ transições de estados. Assim, entram no processo de otimização somente os códigos para os quais, em relação a essa seqüência de referência, apresentam eventos de erro com métrica igual à distância livre.

Agora, resta selecionar o(s) melhor(es) código(s) para cada configuração de memórias e, se possível, a configuração capaz de gerar o melhor código. Essa etapa final de otimização pode ser implementada se forem listados os parâmetros d_1 , N_1 e W_1 para os códigos das Tabs. 6.4 e 6.5 e, a partir destes, uma inspeção permitirá a seleção final. As Tabs. 6.6 e 6.7 listam esses parâmetros. As entradas assinaladas com (*) na Tab. 6.6 indicam os respectivos códigos ótimos na Tab. 6.4 e listados na Tab. 6.6a. O número de métricas considerado foi o menor possível que permitisse a seleção para cada configuração de memórias. Foi feita uma tentativa de selecionar a melhor configuração de memórias através de um aprofundamento na treliça para $\delta = 7\mu = 21$. Entretanto, todos os códigos apresentam praticamente as mesmas propriedades. A Fig. 6.11 apresenta as probabilidades de erro de evento e de bit para os códigos da Tab. 6.6a, obtidas usando-se os parâmetros d_1 , N_1 e W_1 observados para esta última profundidade na treliça e portanto representam as versões truncadas dos limites superiores verdadeiros.

Tabela 6.2 : Códigos com $d_{1, \text{min}} = 2,1414$ para codificação do 4-PSK no 8-PSK (treliça de 8 estados).

M		D = 6 4		D = 4 2		D = 2 6	
		C	C	C	C	C	C
1	0	4 1 2		2 1 4	6 1 2	4 1 2	
		4 1 6		2 1 6	6 1 4	4 1 6	
		4 3 2		2 3 4	6 3 2	4 3 2	
		4 3 6		2 3 6	6 3 4	4 3 6	
1	1	4 5 2		2 5 4	6 5 2	4 5 2	
		4 5 6		2 5 6	6 5 4	4 5 6	
		4 7 2		2 7 4	6 7 2	4 7 2	
		4 7 6		2 7 6	6 7 4	4 7 6	
1	1	1 2 4	5 2 4	1 4 2		1 4 2	
		1 2 6	5 2 6	1 4 6		1 4 6	
		1 6 2	5 6 2	3 4 2		3 4 2	
		1 6 4	5 6 4	3 4 6		3 4 6	
		3 2 4	7 2 4	5 4 2		5 4 2	
		3 2 6	7 2 6	5 4 6		5 4 6	
1	0	3 6 2	7 6 2	7 4 2		7 4 2	
		3 6 4	7 6 4	7 4 6		7 4 6	

Tabela 6.3: Códigos com $d_{1, \text{min}} = 2,3094$ para codificação do 4-PSK no OCTAL CÚBICO (treliça de 8 estados).

M		D = 2 6		D = 6 4		D = 4 2	
		C	C	C	C	C	C
1	0	4 1 2	6 1 2	2 1 4	4 1 2	2 1 4	6 1 2
		4 1 6	6 1 4	2 1 6	4 1 6	2 1 6	6 1 4
		4 3 2	6 3 2	2 3 4	4 3 2	2 3 4	6 3 2
		4 3 6	6 3 4	2 3 6	4 3 6	2 3 6	6 3 4
1	1	4 5 2	6 5 2	2 5 4	4 5 2	2 5 4	6 5 2
		4 5 6	6 5 4	2 5 6	4 5 6	2 5 6	6 5 4
		4 7 2	6 7 2	2 7 4	4 7 2	2 7 4	6 7 2
		4 7 6	6 7 4	2 7 6	4 7 6	2 7 6	6 7 4

Tabela 6.3 : continuação.

M		D = 6 4		D = 4 2		D = 2 6	
		C	C	C	C	C	C
1	1	1 2 4	5 2 4	1 4 2	5 4 2	1 2 4	5 2 4
		1 2 6	5 2 6	1 4 6	5 4 6	1 2 6	5 2 6
		1 6 2	5 6 2	1 6 2	5 6 2	1 4 2	5 4 2
		1 6 4	5 6 4	1 6 4	5 6 4	1 4 6	5 4 6
1	0	3 2 4	7 2 4	3 4 2	7 4 2	3 2 4	7 2 4
		3 2 6	7 2 6	3 4 6	7 4 6	3 2 6	7 2 6
		3 6 2	7 6 2	3 6 2	7 6 2	3 4 2	7 4 2
		3 6 4	7 6 4	3 6 4	7 6 4	3 4 6	7 4 6

Tabela 6.4: Códigos ótimos para codificação do 4-PSK no 8-PSK para cada M e D.

M	D=4 2		D=6 4		D=2 6	
	C	C	C	C	C	C
1 0	6 1 2	4 3 2	4 3 6			
1 1		4 5 2	4 5 6			
1 1	3 4 2	1 6 2	7 4 2			
1 0	5 4 2	7 6 2				

Tabela 6.5: Códigos ótimos para codificação do 4-PSK no OCTAL CÚBICO para cada M e D.

M	D=4 2		D=6 4		D=2 6	
	C	C	C	C	C	C
1 0	6 3 2	2 5 4	4 7 6			
1 1						
1 1	5 6 4	7 2 6	3 4 2			
1 0						

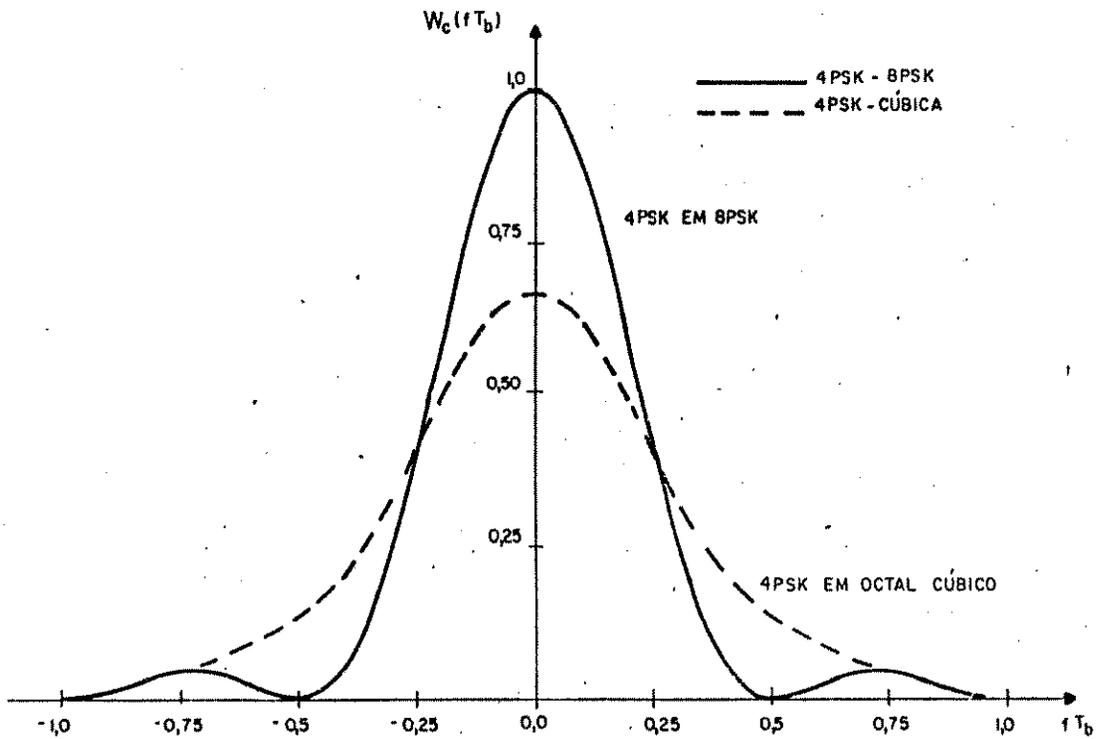


Fig. 6.9: Espectro de potência para o 4-PSK codificado como 8-PSK e OCTAL CÚBICO em função da frequência normalizada fT_b , onde T_b é o intervalo de bit e o intervalo de símbolo $T_{símb} = 1$.

Tabela 6.6: Parâmetros d_1 , N_1 e W_1 para os códigos da Tab. 6.4.

d_1	4,59	5,17	5,76	6,0
D	4 2	6 4	2 6	
M	N_1	W_1	N_1	W_1
	1	1	2	5
1 0	2	8	2	7
1 1	3	16	4	24
	1	1	1	1
	2	5	1	1 *
1 1	3	13	2	7
1 0	5	34	3	17
	1	1	1	1

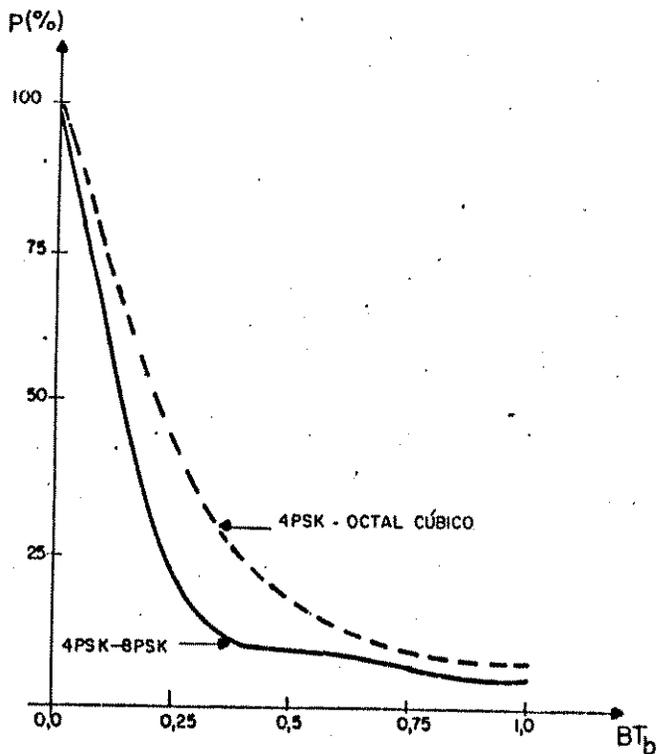


Fig. 6.10: Potência fora da faixa para os espectros da Fig. 6.9.

Tabela 6.6a - Códigos ótimos para cada configuração de memórias (4-PSK codificado em 8-PSK).

	M = 1 0				M = 1 1			
	D		C		D		C	
	6 4	1 2 6			6 4	1 6 2		
	6 4	7 2 6			6 4	7 6 2		
$d,^{22}$	4,59	5,17	5,78	6,00	6,34	6,59	6,93	7,17
	7,51	8,10	8,69	9,27	9,86			
N_1	1	2 3	1 4	2 4	2 4	4 4	4 4	4
W_1	1	7 17	1 30	4 39	6 44	48 56	63 7	

Tabela 6.7: Parâmetros d_1 , N_1 e W_1 para os códigos da Tab. 6.5.

d_1	5,33	6,67	8,0	9,33	10,67	
D	4 2	6 4	2 6			
M	N_1	W_1	N_1	W_1	N_1	W_1
	1	1	1	1	1	1
1 0	4	9	4	9	4	9
1 1	9	35	9	35	9	35
	7	38	7	38	7	38
	8	64	8	64	8	64
	1	1	1	1	1	1
1 1	4	9	4	9	4	9
1 0	9	35	9	35	9	35
	7	38	7	38	7	38
	8	62	8	62	8	62

Tabela 6.7a - Parâmetros para os códigos da Tab. 6.5 obtidos para uma profundidade na treliça de $\delta = 6\mu = 18$.

d_1	5,33	6,67	8,00	9,33	10,67	12,00	13,33	14,67
N_1	1	4	9	7	8	7	8	7
W_1	1	9	35	38	64	60	89	95

Quanto aos códigos da Tab. 6.5, os mesmos já correspondem aos códigos ótimos para cada configuração de memórias. A Tab. 6.7 apresenta os parâmetros d_1 , N_1 e W_1 para esses códigos. Como se pode observar, todos apresentam praticamente as mesmas propriedades. A Fig. 6.12 mostra as probabilidades de erro de evento e de bit obtidas com esses parâmetros observados para uma profundidade na treliça de $\delta = 6\mu = 18$ e listados na Tabela 6.7a.

Para finalizar estes capítulo, resta fazer uma observação com relação à constelação prismática. Foi feita uma tentativa de seleção dos códigos ótimos para a mesma. Entretanto, a distância livre obtida para os códigos foi $d_{1, \text{livre}} = 2,1900$ e por conseguinte inferiores aos códigos obtidos para a constelação cúbica.

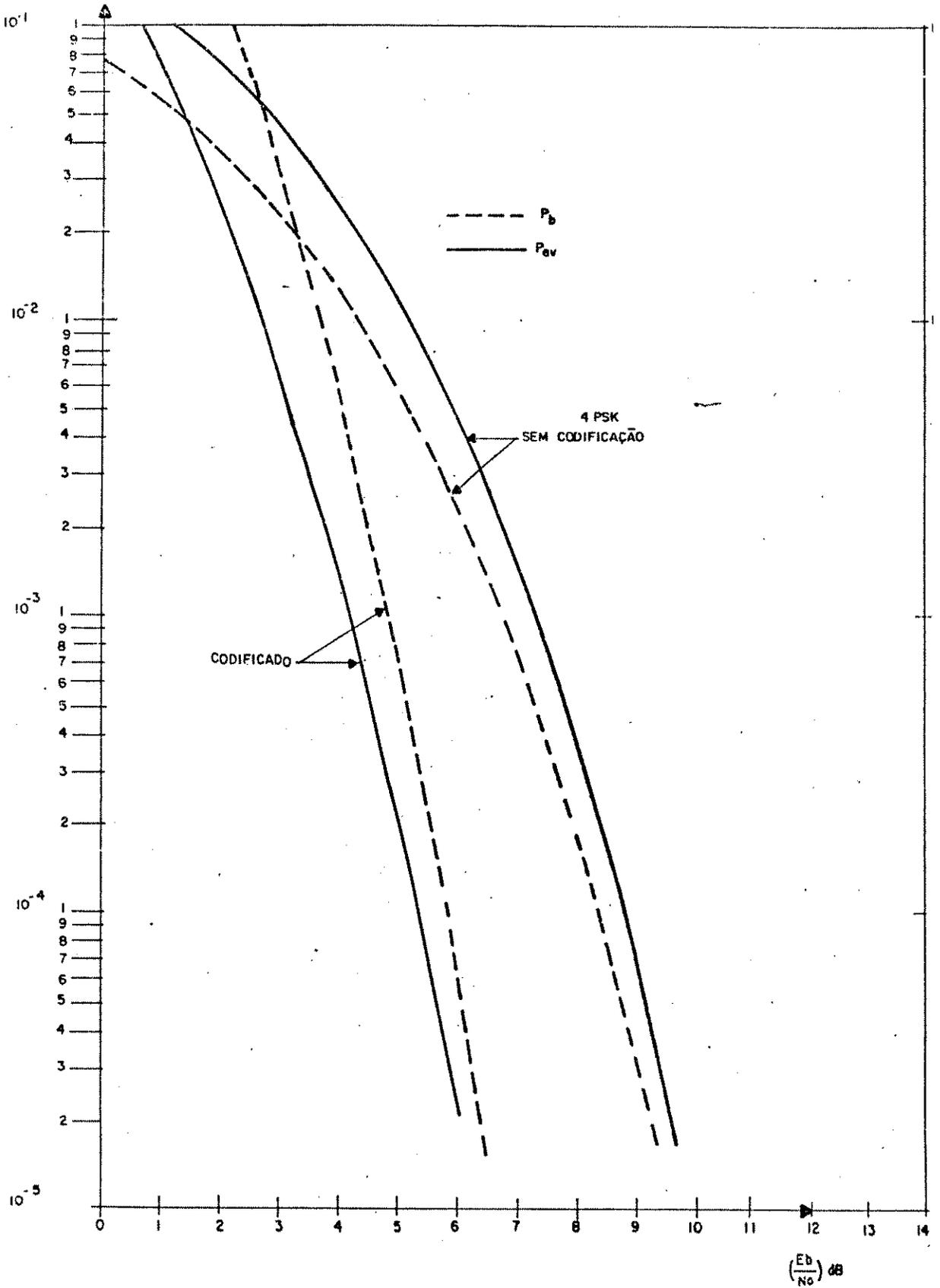


Fig. 6.11: Probabilidades de erro de evento e de bit para os códigos ótimos na Tab. 6.6a, correspondendo às versões truncadas dos limites superiores expressos pela Eq. 4.32.

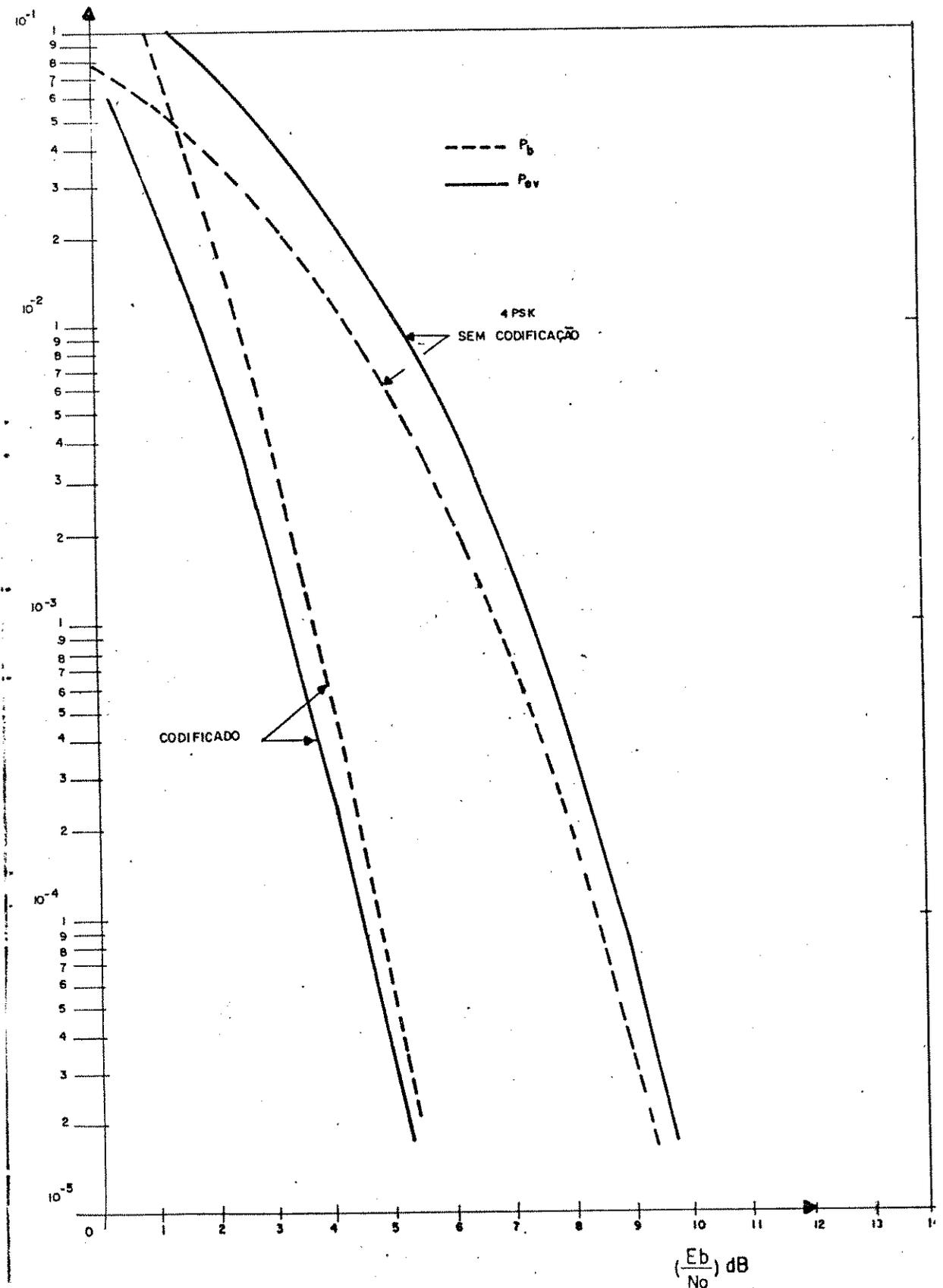


Fig. 6.12: Probabilidades de erro de evento e de bit para os códigos ótimos na Tab. 6.5, correspondendo às versões truncadas dos limites superiores expressos pela Eq. 4.32.

CAPÍTULO 7

SIMULAÇÃO DE SISTEMAS COM MODULAÇÃO CODIFICADA

7.1 - Introdução

O desempenho de um sistema de transmissão digital com modulação codificada, em princípio só pode ser avaliado através do cálculo das probabilidades de erro de evento, P_{we} , e de bit, P_b . Infelizmente, como foi visto no Cap. 4, o cálculo dessas probabilidades é de difícil avaliação e somente são disponíveis estimativas baseadas na probabilidade da união de eventos que, para o sistema estudado ao longo desse trabalho, corresponde a uma série com um número infinito de termos (Eqs. 4.29 e 4.30). No problema específico da avaliação do desempenho de códigos convolucionais, podem ser obtidas expressões fechadas para P_{we} e P_b aplicando-se procedimentos fundamentados na teoria dos fluxogramas lineares e que consiste em se determinar a função de transferência entre os pontos do diagrama de estados gerado pela abertura da automalha associada à transição de estado $0 \rightarrow 0$ (Viterbi e Omura, 1979, Cap. 4). Entretanto, esse procedimento pressupõe que a uniformidade da probabilidade de erro (UPE) seja verdadeira, fato que não ocorre para os códigos de treliça em estudo neste trabalho, exceto para os códigos superlineares. Além dessa restrição, mesmo que a UPE seja verdadeira, o problema aumenta de complexidade à medida que cresce o número de estados do sistema. Presentemente, esse problema é resolvido por um procedimento de truncagem de uma série de potências matricial (Clark e Cain, 1981, Cap. 6). Assim, restam duas possibilidades para se avaliar o desempenho de um código de treliça. A primeira consiste na exploração das propriedades da treliça a uma profundidade que possibilite a obtenção de uma boa aproximação para os limitantes superiores de P_{we} e P_b . Entretanto, este procedimento requer um esforço computacional muito grande, só tendo sentido sua aplicação para a geração de poucos termos nas séries caracterizadas por (4.29) e (4.30), tal como foi utilizado na otimização de códigos com máxima distância livre no Cap. 4. A segunda opção consiste em simular o sistema de transmissão digital que utiliza o código cujo desempenho se deseja avaliar. Entretanto, esta opção também apresenta suas limitações pelos longos tempos de execução necessários à obtenção de cada ponto na curva de probabilidades e só é viável para probabilidades maiores que 10^{-4} .

Uma solução pode ser dada para esse problema se for utilizado o fato de que para probabilidades de erro abaixo de 10^{-4} o limite superior expresso por (4.29) é dominado pelos primeiros termos da série e limitado inferiormente somente pelo primeiro termo, o que assegura uma excelente aproximação para a probabilidade de erro real. Assim, é conveniente o uso de um procedimento que usa simulação para baixas e médias relações sinal/ruído ($P_{we} > 10^{-4}$) e o uso dos primeiros termos da série em

(4.29) para medir o desempenho para altas relações sinal/ruído ($P_{\text{m}} < 10^{-4}$).

O objetivo deste capítulo é a apresentação do modelo de um sistema de transmissão digital com modulação codificada controlada por um código convolucional, que usa constelação de sinais em até três dimensões. Usando o modelo desenvolvido ao longo dos Caps. 2 - 4, o sistema será caracterizado de forma apropriada à implementação de sua simulação.

7.2 - Modelo para um sistema com modulação codificada

A Fig. 7.1 ilustra o modelo em diagrama de blocos para um sistema de transmissão digital com modulação codificada controlada por um código convolucional e com constelações de sinais em até três dimensões. A fonte q-ária gera uma seqüência de símbolos $\{a_i\}$. Os símbolos dessa seqüência são agrupados em blocos de k símbolos e, após a conversão série/paralelo (S/P), transformados numa seqüência de vetores q-ários de k componentes, $\{x^n\}$, onde cada vetor x^n corresponde à entrada do codificador no n -ésimo intervalo de símbolo T_m . A saída do codificador é uma seqüência $\{z^n\}$, onde z^n é a palavra-código associada à entrada x^n e ao estado y^n através de (2.11b). Assim, à seqüência de entrada $\{x^n\}$ corresponde uma seqüência de estados $\{y^n\}$.

Através do MPC, cada z^n dá origem à transmissão de um símbolo da constelação com duração T_m , cuja seqüência forma o sinal codimodulado $z(t)$ que, após sofrer as interferências do canal, suposto com ruído aditivo gaussiano e branco, é transformado no sinal $r(t)$ na entrada do receptor. Este sinal, após a detecção, é transformado numa seqüência de pontos do espaço de sinais, $\{r^n\}$, onde cada vetor r corresponde às coordenadas de cada um desses pontos. A partir dessa seqüência de pontos, o receptor de Viterbi estima a seqüência de informação transmitida, $\{\hat{x}^n\}$, que após a conversão paralelo/série (P/S) resulta numa estimativa para a seqüência de fonte transmitida, $\{\hat{a}_i\}$.

A seguir cada um dos blocos do sistema acima descrito será modelado de forma apropriada à implementação dos procedimentos computacionais que possibilitarão a simulação do sistema. Antes porém, é conveniente que se estabeleçam os critérios a serem adotados na implementação da simulação tendo em vista a disponibilidade dos recursos computacionais do usuário no que se refere à capacidade de armazenamento em memória. Neste trabalho se optou por uma simulação onde o experimento é subdividido em blocos de NP símbolos. Desse modo, os procedimentos a serem desenvolvidos poderão ser adaptados aos recursos de cada sistema. Isto é, se a disponibilidade de memória não for um fator limitante, o experimento poderá ser processado de uma só vez por um redimensionamento compatível com o tamanho do experimento,

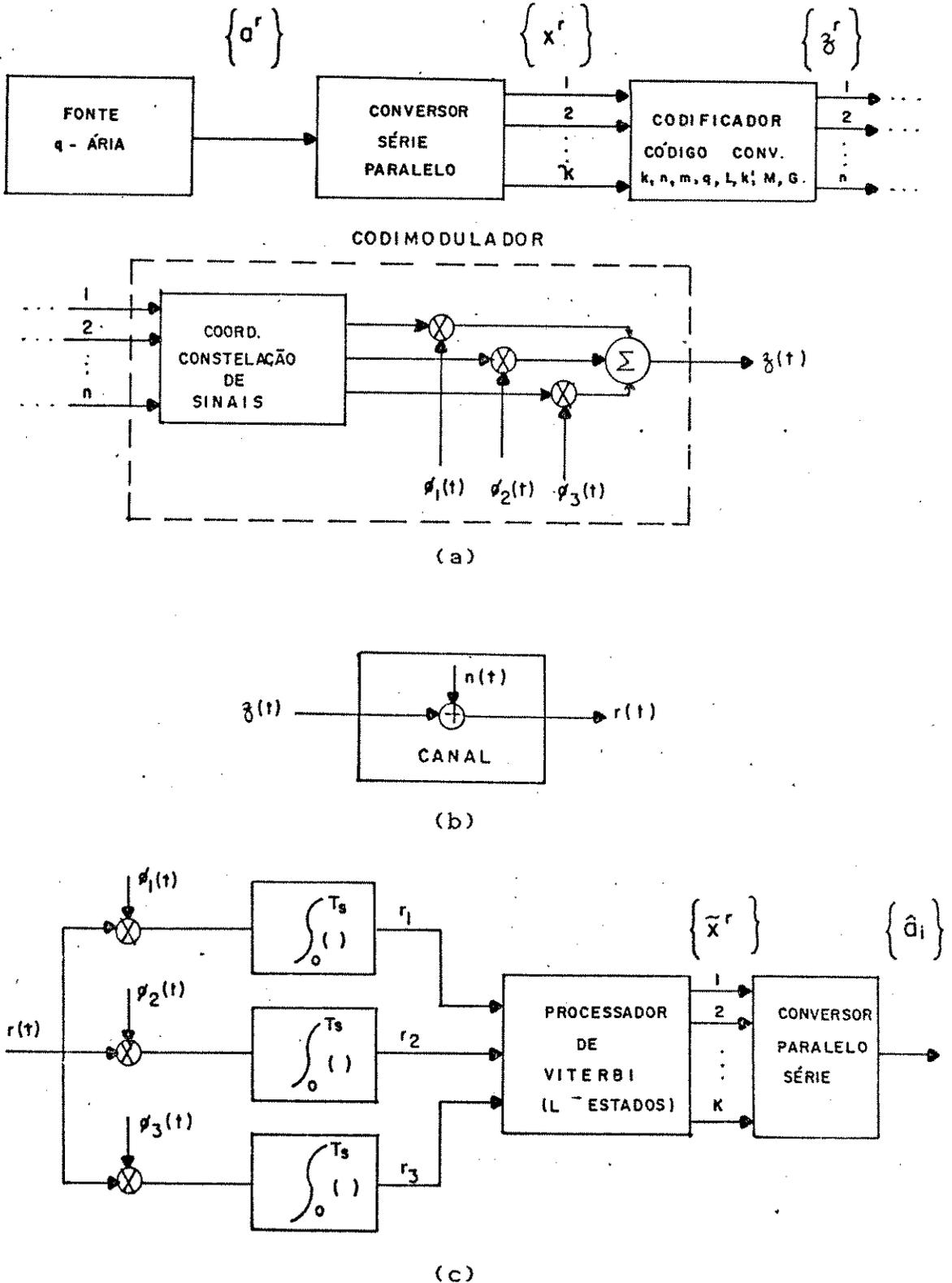


Fig. 7.1: Modelo em diagrama de blocos para um sistema com modulação codificada controlada por um código convolucional. a) Transmissor. b) Canal. c) Receptor

atribuindo-se a NP o valor correspondente. Por outro lado, se a disponibilidade de memória for restritiva, o experimento será subdividido em subexperimentos com NP símbolos, NP escolhido de modo a compatibilizar a disponibilidade de armazenamento.

7.2.1 - Fonte q-ária e conversor S/P

Seja $X = (x_1, x_2, \dots, x_{NP})$ uma seqüência de números inteiros K-ários, sendo $K=q^k$ o número de entradas distintas do codificador associado ao código convolucional. Se para cada x_i se obtém sua representação q-ária de k componentes, x_i , a seqüência $[x_1, x_2, \dots, x_{NP}]$ pode ser considerada como a saída do conversor S/P. Assim, para simular a fonte q-ária e o conversor S/P é suficiente a geração de uma seqüência pseudoaleatória de inteiros K-ários seguido da obtenção de suas respectivas representações vetoriais.

Em geral, a geração de qualquer experimento com alguma característica estatística costuma ser realizada a partir de um gerador pseudoaleatório padrão que simula a geração de uma variável aleatória distribuída uniformemente no intervalo $[0, 1]$. A implementação desse gerador pressupõe a observação de alguns pré-requisitos tais como rapidez na geração dos números, confiabilidade estatística e período da seqüência pseudoaleatória, que deve ser o mais longo possível. Em sistemas de grande porte, a implementação do gerador básico se fundamenta no método congruencial multiplicativo (Hull e Dobell, 1962; Chambers, 1967) implementado de acordo com Payne, Rabung e Bogyo (1969). Nessa implementação, o período da seqüência é máximo (2.147.483.646) e a velocidade de geração é assegurada por uma codificação em linguagem de máquina de tal forma que somente são realizadas operações de deslocamento de bits. Entretanto, tal procedimento só pode ser implementado em máquinas com palavras de comprimento acima de 31 bits. Assim, se for aceitável uma perda na velocidade de geração, é possível a implementação em linguagem de alto nível de um gerador básico com as características acima citadas, desde que o sistema a ser utilizado disponha de inteiro duplo (4 bytes). Uma implementação desse tipo foi codificada em FORTRAN de modo que, exceto por uma divisão que normaliza o número no intervalo $[0, 1]$, todas as outras operações são adições ou comparações.

Admitindo-se a disponibilidade de um bom gerador básico, um gerador de números pseudoaleatórios K-ários pode ser obtido simplesmente multiplicando-se a saída do gerador básico por K seguido de retenção da parte inteira. Como nunca pode ocorrer o aleatório 1.0, este procedimento assegura inteiros aleatórios no intervalo fechado $[0, K-1]$. Assim, ficam estabelecidas as condições para a simulação do sinal de entrada do codificador.

7.2.2 - Codificação

A simulação da codificação é imediata através da utilização das Eqs. 2.11a e 2.11b. Tudo que é necessário é o conhecimento prévio da estrutura de memórias do codificador (matriz M) para implementação da função transição de estado (veja apêndice 2A) e da matriz geradora do código para a implementação da função de saída. Entretanto, para uma simulação baseada num processamento em blocos com NP símbolos de entrada, é necessário que sejam asseguradas condições que garantam a continuidade da seqüência de estados do codificador, como se o processamento fosse realizado de forma contínua.

Sejam então X^i , Y^i e Z^i os blocos de entrada, estado e saída, respectivamente, correspondentes à i -ésima iteração do procedimento de simulação e cujos componentes estão interrelacionados através de (2.11). Isto é,

$$\begin{aligned} X^i &= (x_{1i}^i, x_{2i}^i, \dots, x_{NPi}^i) \\ Y^i &= (y_{1i}^i, y_{2i}^i, \dots, y_{NPi}^i) \\ Z^i &= (z_{1i}^i, z_{2i}^i, \dots, z_{NPi}^i) \end{aligned} \quad (7.1)$$

onde todos os componentes são inteiros correspondentes às respectivas representações dos vetores de entrada, estado e saída.

A continuidade da seqüência de estados do codificador é assegurada se forem satisfeitas as seguintes condições entre os componentes dos blocos em (7.1):

$$y_{1i}^i = A \cdot y_{NPi}^{i-1} + B \cdot x_{1i}^i \quad (7.2a)$$

$$z_{1i}^i = C \cdot y_{NPi}^{i-1} + D \cdot x_{1i}^i \quad (7.2b)$$

$$y_{ji}^i = A \cdot y_{j-1i}^i + B \cdot x_{ji}^i \quad (7.3a)$$

$$z_{ji}^i = C \cdot y_{j-1i}^i + D \cdot x_{ji}^i, \quad j=2,3, \dots, NP. \quad (7.3b)$$

Note que em (7.2) o estado atual corresponde ao último estado do bloco processado na iteração imediatamente anterior e é esta relação que assegura a continuidade da seqüência de estados a cada iteração do processamento. Como os blocos X^i , Y^i e Z^i são

armazenados como seqüências de inteiros, ao serem usadas as Eqs. 7.2 ou 7.3 devem ser obedecidos os seguintes passos:

- 1 - Tomar a componente $x_{j'} (x_{1'})$ e obter a representação q-ária $x_{j'} (x_{1'})$ de dimensão k ;
- 2 - Tomar a componente $y_{j-1'} (y_{NP^{i-1}'})$ e obter sua representação q-ária $y_{j-1'} (y_{NP^{i-1}'})$ de dimensão μ ;
- 3 - Determinar a transição de estado $y_{j'} (y_{1'})$ através da Eq. 7.3a (7.2a) e obter sua representação decimal $y_{j'} (y_{1'})$;
- 4 - Determinar a saída $z_{j'} (z_{1'})$ através da Eq. 7.3b(7.2b) e obter sua representação decimal $z_{j'} (z_{1'})$.

7.2.3 - Codimodulação

Como visto anteriormente, a saída do codificador do código convolucional na i -ésima iteração é o bloco de inteiros

$$Z^i = (z_{1'}^i, z_{2'}^i, \dots, z_{NP^i}^i) \quad (7.4)$$

onde cada componente é a representação decimal da respectiva palavra-código gerada à saída do codificador.

No Cap. 3 foi visto que a matriz P em (3.6) traduz, através do índice de linhas, o mapeamento entre as palavras-código e os símbolos da constelação código. Assim, a partir de (7.4) e (3.6) é possível a geração da seqüência de símbolos (sinal codimodulado) a ser processada pelo canal na i -ésima iteração, sendo completamente especificada pela matriz

$$S^i = [s^i(j,l)] , \quad \begin{array}{l} j=1,2, \dots, NP \\ l=1,2, \dots, NDIM \end{array} \quad (7.5)$$

onde $s^i(j,l) = p(z_{j+1}^i, l)$ e NDIM é o número de dimensões do espaço de sinais.

Gerada a seqüência de símbolos a ser transmitida através do canal, o bloco Z^i perde sua utilidade e o respectivo espaço em memória pode ser utilizado para outros fins. Entretanto, os blocos X^i e Y^i devem ser preservados até o final da i -ésima iteração pois, como será visto oportunamente, os mesmos serão de importância fundamental na contagem dos eventos de erro e dos erros de bit de informação.

7.2.4 - Caracterização do canal

O efeito do canal sobre o sinal transmitido, em geral se faz sentir através do somatório de vários componentes tais como o ruído propriamente dito, a interferência intersimbólica e as interferências cocanal e de canais adjacentes. Entretanto, neste trabalho somente será considerado o efeito do ruído de canal, suposto gaussiano branco e aditivo de média zero e variância σ^2 por dimensão.

A geração de ruído gaussiano de média m e variância σ^2 pode ser feita a partir do gerador aleatório básico através das transformações esquematizadas na Fig. 7.2 e denominadas de método "log-trig" de geração de variáveis gaussianas (Mihram, 1972). Entretanto, pelo número de operações necessárias, esse método só deve ser usado quando não houver outros recursos. Neste trabalho, será usado como gerador de ruído gaussiano uma sub-rotina adaptada a partir de um gerador gaussiano pertencente à biblioteca de "software" para aplicações científicas NAGLIB. Essa subrotina usa um banco de dados interno de modo que os números são gerados usando-se comparações, adições e pouquíssimas multiplicações. Essas características asseguram uma geração veloz para os números gaussianos.

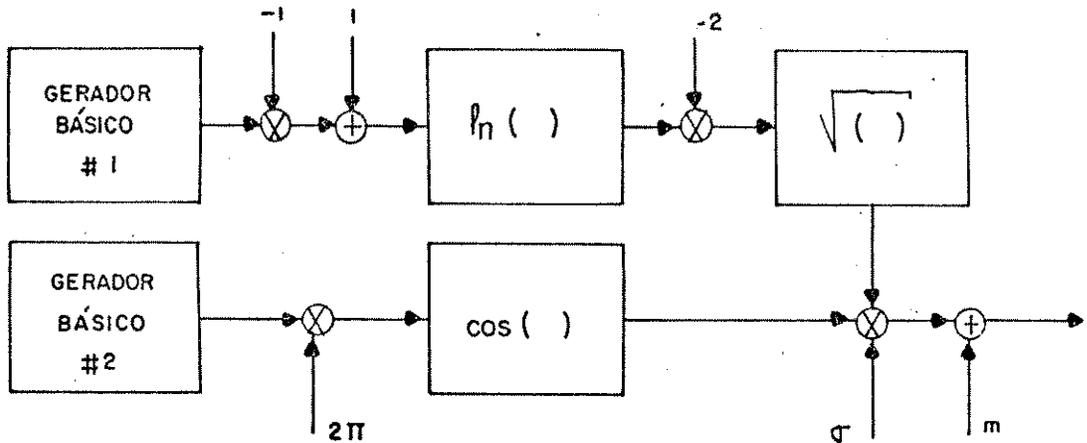


Fig. 7.2: Geração de números aleatórios gaussianos de média m e variância σ^2 pelo método "log-trig".

Solucionado o problema da geração de números aleatórios gaussianos, o efeito do canal sobre o sinal transmitido é implementado através da relação

$$r'(J,1) = s'(J,1) + n'(J,1), \quad (7.6)$$

onde $s'(J,1)$ ($1=1,2, \dots, \text{NDIM}$) são os componentes da J -ésima linha de S em (7.5) e $n'(J,1)$ são os componentes do vetor de ruído que o canal introduz no respectivo elemento de sinal transmitido. Note que sob o ponto de vista computacional não há necessidade de se definir um novo espaço em memória para armazenar o sinal que chega à entrada do receptor. Esse armazenamento é feito na própria matriz S através da relação $s'(J,1) \leftarrow s'(J,1) + n'(J,1)$. Assim, é a partir de S que o receptor deve estimar a seqüência de informação transmitida.

7.2.5 - Receptor

O ponto crítico na simulação do sistema, é o receptor, cuja implementação se fundamenta no algoritmo de Viterbi, modelado segundo a estrutura apresentada no Cap. 3. A implementação em "software" do referido modelo possibilita que a partir de (7.5), após o uso de (7.6), o receptor forneça um bloco de estimativas correspondente à i -ésima iteração,

$$X^i = (\tilde{x}_{11}^i, \tilde{x}_{21}^i, \dots, \tilde{x}_{NP}^i) . \quad (7.7)$$

Entretanto, devido ao atraso de decodificação, (7.7) não corresponde à estimativa de X^i . Na realidade, se o atraso de decodificação é de δ símbolos, $\tilde{x}_{11}^i, \tilde{x}_{21}^i, \dots, \tilde{x}_{\delta}^i$ são estimativas de $x_{NP-\delta+2}^{i-1}, x_{NP-\delta+3}^{i-1}, \dots, x_{NP}^{i-1}$. Assim, durante a contagem de erros de símbolos de informação, as comparações terão que ser feitas segundo o esquema abaixo, componente a componente.

$$\begin{bmatrix} x_{NP-\delta+2}^{i-1}, & x_{NP-\delta+3}^{i-1}, & \dots, & x_{NP}^{i-1} \end{bmatrix} \begin{bmatrix} x_{11}^i, & x_{21}^i, & \dots, & x_{NP-\delta+1}^i \end{bmatrix} \\ \begin{bmatrix} \tilde{x}_{11}^i & , & \tilde{x}_{21}^i & , & \dots, & \tilde{x}_{\delta-1}^i & \tilde{x}_{\delta}^i, & \dots, & \tilde{x}_{NP}^i \end{bmatrix} \quad (7.8)$$

Portanto, a cada iteração i , a cauda correspondente aos últimos $\delta-1$ símbolos em X^i deve ser preservada até o final da iteração $i+1$.

7.2.6 - Contagem de erros de evento e de bit

A contagem dos eventos de erro pode ser implementada de uma forma relativamente simples, já que ao longo do processamento estão preservados os blocos X^i e Y^i . Assim, como Y^i é a seqüência de estados correta (trajetória de referência), a geração dos eventos de erro pode ser implementada através da codificação da seqüência de informação estimada pelo receptor, obtendo-se a respectiva seqüência de estados (trajetória decidida como a verdadeira pelo receptor),

$$Y^i = (\tilde{y}_{1^i}, \tilde{y}_{2^i}, \dots, \tilde{y}_{N_{IP}^i}) , \quad (7.9)$$

através de (7.2) e (7.3) aplicadas a X^i . Uma comparação semelhante àquela expressa em (7.8) permitirá a identificação dos eventos de erro. Isto é,

$$\begin{bmatrix} y_{N_{IP} - \delta + 2^i - 1}, & y_{N_{IP} - \delta + 3^i - 1}, & \dots & y_{N_{IP}^i - 1} \end{bmatrix} \begin{bmatrix} y_{1^i}, & y_{2^i}, & \dots & y_{N_{IP} - \delta + 1^i} \end{bmatrix} \\ \begin{bmatrix} \tilde{y}_{1^i} & , & \tilde{y}_{2^i} & , & \dots & \tilde{y}_{\delta - 1^i} & \tilde{y}_{\delta^i}, & \dots & \tilde{y}_{N_{IP}^i} \end{bmatrix} \quad (7.10)$$

Assim, a cada iteração i , também deve ser preservada até o final da iteração $i+1$ a cauda do bloco Y^i correspondente aos últimos $\delta - 1$ componentes.

As comparações estabelecidas em (7.8) e (7.10) permitem a contagem de erros de bit de informação e de eventos de erro, respectivamente. Entretanto, a implementação do algoritmo que possibilita a contagem de erros ao longo da simulação deve prover meios que possibilitem a verificação das seguintes condições:

- 1 - Identificação do início e final do evento, ainda que os mesmos estejam localizados em blocos processados em iterações distintas;
- 2 - Identificado o início de um evento, incrementar o contador de eventos de erro de uma unidade, iniciar a contagem dos símbolos de informação errados contidos no evento e simultaneamente contar a duração do evento;
- 3 - Para cada erro de símbolo, por exemplo, $\tilde{x}_j^i \neq x_j^i$, obter os correspondentes vetores q -ários de k componentes e contar o número de componentes em que os mesmos diferem, obtendo-se assim o número de símbolos q -ários errados (bits para $q=2$). Incrementar o contador de erros de símbolos q -ários com esse valor obtido;

- 4 - Identificado o final de um evento, incrementar o contador que mede o tempo do sistema em erro com o valor correspondente à duração do evento.

Satisfeitas estas condições, ao final da simulação estarão disponíveis os seguintes dados:

- 1 - Número de símbolos processados: $NSYMB = NP \cdot NITER$, onde $NITER$ é o número de iterações da simulação;
- 2 - Número de eventos de erro observados: $NEVTO$;
- 3 - Número de símbolos q-ários errados: $NEBIT$;
- 4 - Tempo do sistema em erro (em símbolos de canal): $TERRO$.

7.3 - Estimativa das Probabilidades de erro

A avaliação da probabilidade de evento de erro usualmente é feita através da medida da taxa de eventos de erro por unidade de tempo (Forney, 1972), ou de outra forma, através do inverso do tempo médio entre eventos de erro consecutivos. Assim, a partir dos dados disponíveis do modelo de simulação da seção 7.2, se obtém

$$P_{e\downarrow} = NEVTO/NSYMB, \quad (7.11)$$

cujo inverso é o tempo médio entre dois eventos de erro consecutivos, onde o tempo é medido em intervalos de símbolos T_m .

Entretanto, segundo Forney (1972), (7.11) é uma boa estimativa de $P_{e\downarrow}$ apenas para altas relações sinal/ruído. Assim, como o principal objetivo da simulação é a avaliação de $P_{e\downarrow}$ nas baixas e médias relações sinal/ruído, o uso de (7.11) não fornece uma avaliação realista do desempenho do sistema. Aqui é sugerido que se considere $P_{e\downarrow}$ como o inverso do tempo médio em que o sistema permanece sem erro entre dois eventos de erro consecutivos, o que resulta em

$$P_{e\downarrow} = NEVTO/(NSYMB-TERRO), \quad (7.12)$$

e que equivale a (7.11) para altas relações sinal/ruído.

A Fig. 7.3 ilustra os princípios usados na obtenção de (7.11) e (7.12), onde T_{ev} representa o tempo entre dois eventos de erro consecutivos enquanto T_e e T_o representam os tempos de duração do evento e do sistema em acerto, respectivamente.

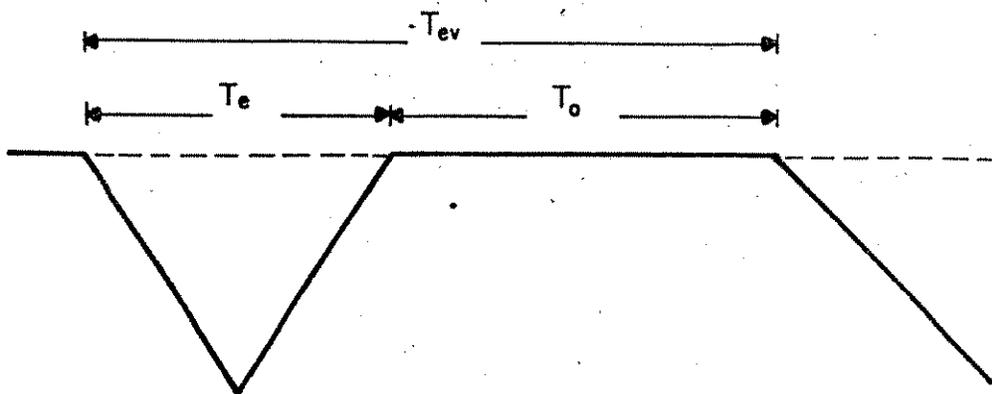


Fig. 7.3: Ilustração dos princípios usados na avaliação de P_{ev} através das Eqs. (7.11) e (7.12).

Quanto à avaliação da probabilidade de erro de bit ($q=2$), não existem dificuldades, sendo definida como a razão entre o número de bits errados e o número de bits processados. Isto é,

$$P_b = \text{NEBIT} / (k \times \text{NSYMB}) \quad (7.13)$$

onde k é o número de bits de informação associados a cada intervalo de modulação.

Por fim, resta estabelecer qual o número de símbolos a ser usado na simulação para uma razão sinal/ruído específica. Um critério que se usa normalmente é que o comprimento do experimento seja pelo menos dez vezes o inverso da probabilidade a ser estimada. Como a priori não se conhece a probabilidade de erro esperada, esse critério fica difícil de ser utilizado. A solução adotada neste trabalho é que a probabilidade esperada seja estimada através da função

$$P_{ev} = Q(d_{1, ev} / 2\sigma), \quad (7.14)$$

onde σ se relaciona com a relação sinal/ruído através de (6.15)-(6.17). Assim, o número de símbolos a serem processados é dado por

$$NSYMB = 10/P_{\text{min}} \quad (7.15)$$

Como a simulação é feita em blocos de NP símbolos, NSYMB tem que ser o menor múltiplo de NP que excede (7.15). Isto é,

$$NSYMB = \lceil 10/P_{\text{min}} \rceil + NP \quad (7.16)$$

onde $\lceil a \rceil$ representa a parte inteira de a . Dessa forma, fica estabelecido o número de iterações a serem consideradas no procedimento de simulação (número de blocos), especificado por

$$NITER = NSYMB/NP \quad (7.17)$$

onde NSYMB é dado por (7.16).

O critério estabelecido através de (7.14)-(7.17) não pode ser usado indefinidamente para qualquer relação sinal/ruído devido aos longos tempos computacionais envolvidos na simulação, à medida que crescem as relações sinal/ruído. Assim, deve ser estabelecido um limite máximo para o comprimento do experimento. O limite adotado neste trabalho é $NSYMB \leq 100.000$. Este limite assegura uma boa estimativa para probabilidades maiores que 10^{-4} .

7.4 - Teste do modelo de simulação

Um programa baseado no modelo da seção 7.2 foi desenvolvido com o fim de implementar a simulação de sistemas com modulação codificada controlada por códigos convolucionais e com constelações de sinais em até tres dimensões. Com o objetivo de testar o programa e o modelo desenvolvido ao longo dos Caps. 2-4, inicialmente foi simulado um sistema que transmite 4PSK codificado como BPSK, usando o código convolucional com matriz geradora

$$G = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (7.18)$$

que corresponde ao código exemplo apresentado por Taylor e Chan (1981) para probabilidade de erro de bit e por Ungerboeck (1982) para probabilidade de evento de erro.

O programa desenvolvido trabalha com relações sinal/ruído em dB, definidas segundo (6.15) e (6.16). Entretanto, é conveniente que os resultados obtidos sejam expressos em função da relação E_b/N_ϕ , onde E_b é a energia média por bit de informação e $N_\phi/2$ é a variância do ruído, suposto gaussiano branco, por dimensão. Assim, (6.15) passa a ser expressa como

$$\text{SNR} = 2E_m / (M \times N_\phi). \quad (7.19)$$

A energia média por bit é expressa por

$$E_b = E_m / (k \cdot \log_2 q) \quad (7.20)$$

para o caso em que o código é q-ário. Assim,

$$(E_b/N_\phi) \text{dB} = (\text{SNR}) \text{dB} - (2k \cdot \log_2 q / M) \text{dB}. \quad (7.21)$$

Para o caso específico de 4PSK codificado em 8PSK, a segunda parcela do segundo membro de (7.21) corresponde a 3,0 dB.

A Tab. 7.1 mostra os parâmetros d_i , N_i e W_i obtidos em relação à seqüência de referência (0 0 ... 0) para uma profundidade de $\delta=18$ símbolos na treliça associada ao código (7.18) e ao mapeamento da Fig. 6.8 (MPC da Fig. 3.1).

Tabela 7.1: Parâmetros N_i , d_i e W_i para o código caracterizado pela Eq. 7.18 e pelo MPC da Fig. 3.1.

d_i	4,59	5,17	5,76	6,00	6,34	6,59	6,93	7,51
N_i	2	4	5	1	5	2	5	5
W_i	5	18	34	1	46	4	57	65

A Fig. 7.4 mostra os resultados obtidos, através da simulação para $P_{\text{er}} \downarrow$, usando-se como estimadores as expressões (7.11) e (7.12). Como se pode observar, o uso de (7.12) apresenta um melhor comportamento para baixas relações sinal/ruído, se os resultados forem comparados com o limite superior truncado, obtido usando-se os parâmetros N_i e d_i da Tab. 7.1 na Eq. 4.32a. Também está mostrado o limite inferior expresso por 4.31a.

A curva resultante da simulação usando-se como estimador de P_{er} a Eq. 7.11 está coerente com os resultados apresentados por Ungerboeck (1982) para um atraso de decodificação de 18 símbolos de canal. A Fig. 7.5 mostra os resultados obtidos para P_b , estimada através de (7.13), como também os limites estabelecidos através de (4.31b) e (4.32b), usando-se os valores de W_i da Tab. 7.1. Os resultados da simulação estão coerentes com aqueles apresentados por Taylor e Chan (1981).

Entretanto, uma análise dos resultados obtidos leva às seguintes observações: 1) Apesar de coerentes com os resultados existentes na literatura, há um deslocamento à direita dos limites superiores truncados de aproximadamente 0,6 dB para P_{er} e 0,2 dB para P_b ; 2) Aparentemente, os resultados da simulação estão mais coerentes com os limites superiores truncados em termos de P_b ; 3) O afastamento dos limites superiores truncados é mais pronunciado nas baixas relações sinal/ruído. Porém, a explicação para esses fatos pode ser feita com base nas seguintes considerações: a) Na obtenção dos limites superiores truncados só foi possível, por dificuldades computacionais, a utilização dos primeiros termos da série e os termos ausentes tem influência principalmente nas baixas e médias relações sinal/ruído; b) Os limites superiores para P_{er} e P_b , estabelecidos por meio das Eqs. 4.29 e 4.30 são obtidos com a hipótese de decodificação ideal (atraso de decodificação infinito) c) O limite superior para P_b é muito menos restrito que o limite para P_{er} já que os coeficientes W_i em geral, na média, são maiores que os coeficientes N_i (veja Tab. 7.1).

As Figs. 7.6 e 7.7 mostram o efeito do atraso de decodificação sobre os resultados da simulação. Como se pode observar, para $\delta = 36$ símbolos, os resultados obtidos praticamente coincidem com os limites superiores truncados, diferenças em torno de 0,1 dB para o caso de P_{er} e no caso de P_b os resultados ou estão sobre a curva do limite superior ou à esquerda da mesma. Portanto, esses resultados asseguram que o programa desenvolvido e assim o modelo no qual o mesmo é baseado caracterizam de modo adequado um sistema de transmissão digital com modulação codificada.

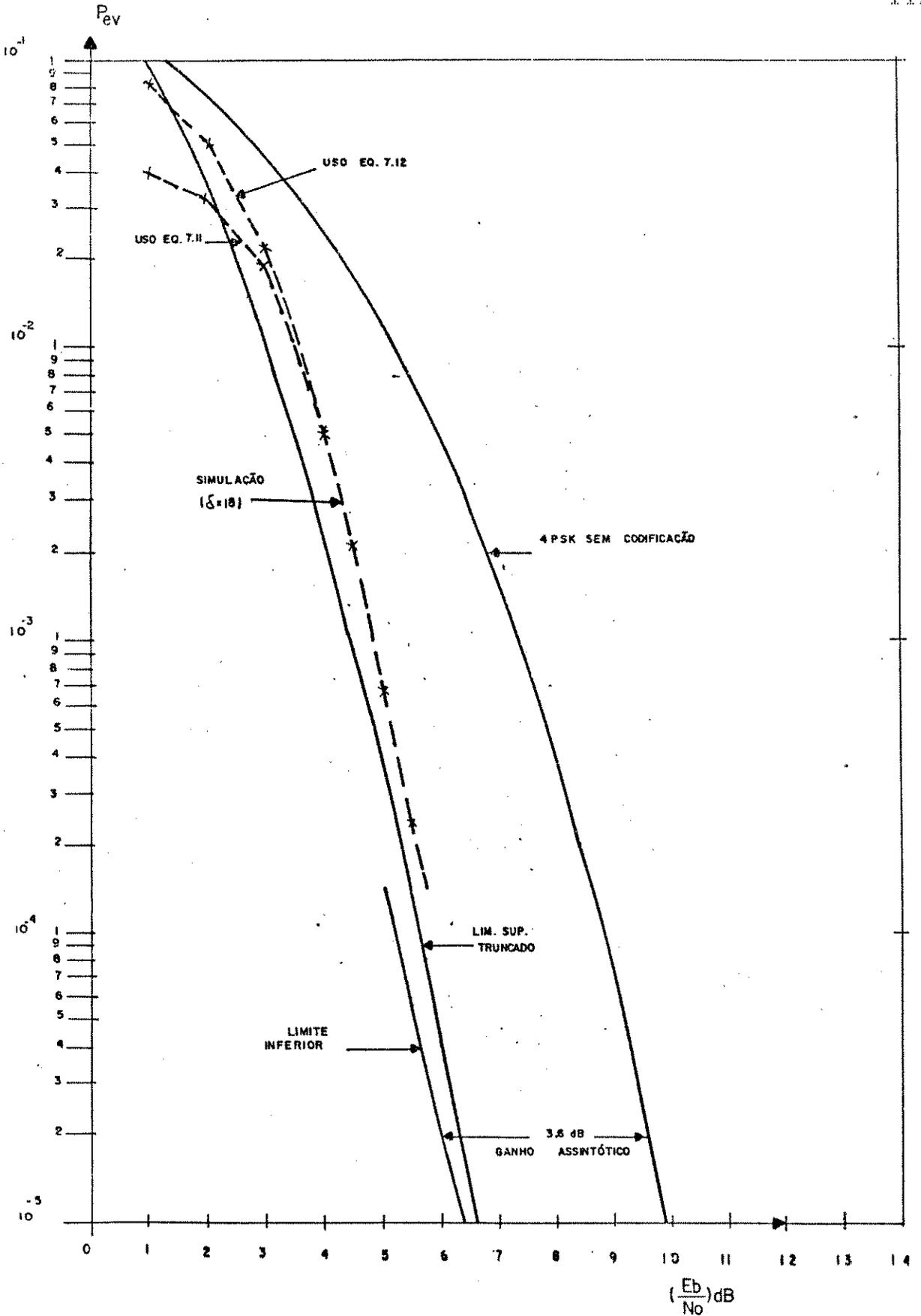


Fig. 7.4: Probabilidade de erro de evento para o 4PSK codificado no BPSK usando o código da Eq. 7.18 e o MPC da Fig. 6.8. Simulação para $\delta = 18$ símbolos.

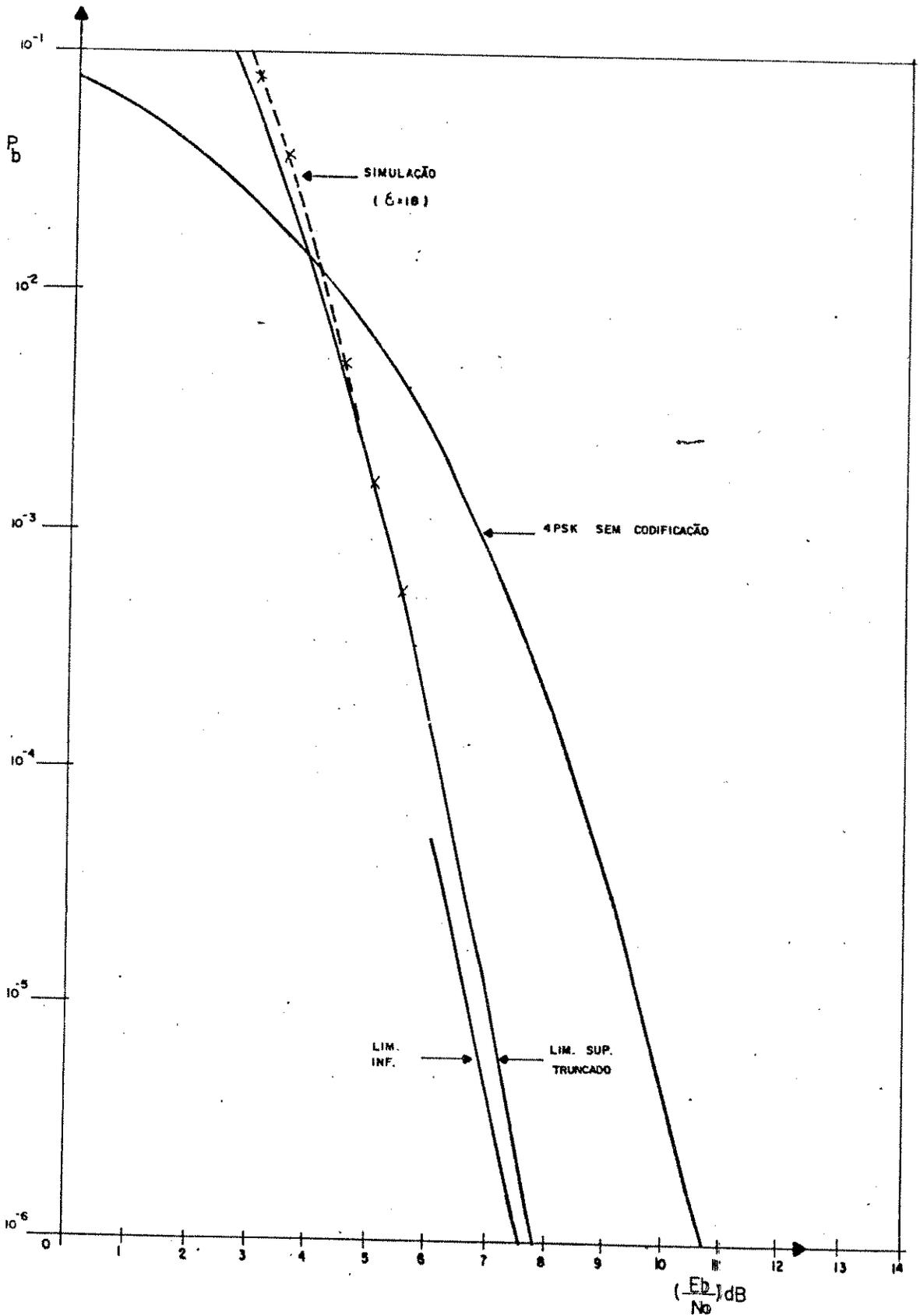


Fig. 7.5: Probabilidade de erro de bit para o 4PSK codificado no BPSK usando o código da Eq. 7.18 e o MPC da Fig. 6.8. Simulação para $\delta = 18$ símbolos.

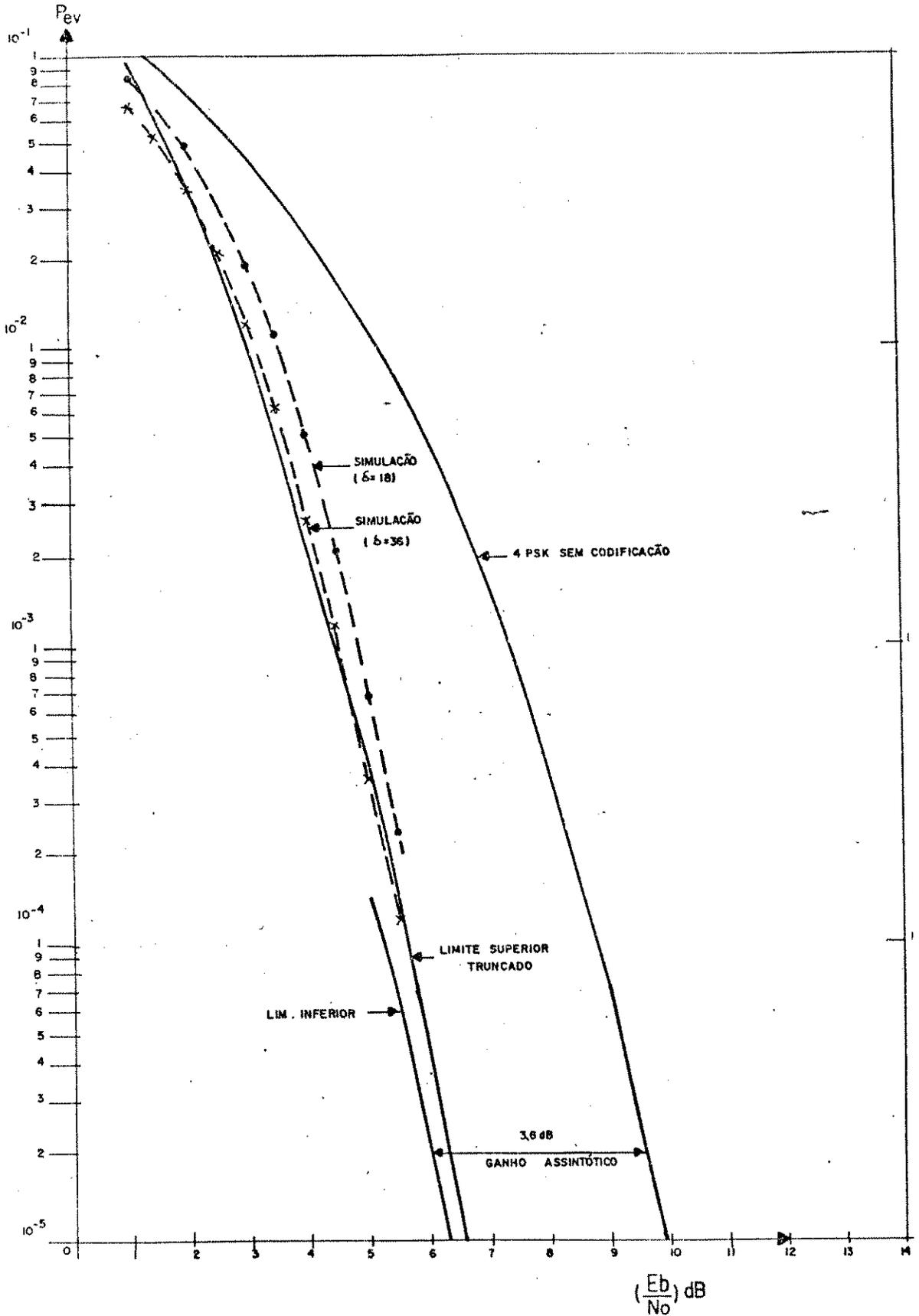


Fig. 7.6: Probabilidade de erro de evento para o 4PSK codificado no 8PSK usando o código da Eq. 7.18 e o MPC da Fig. 6.8. Simulação para $\delta=18$ e $\delta=36$ símbolos.

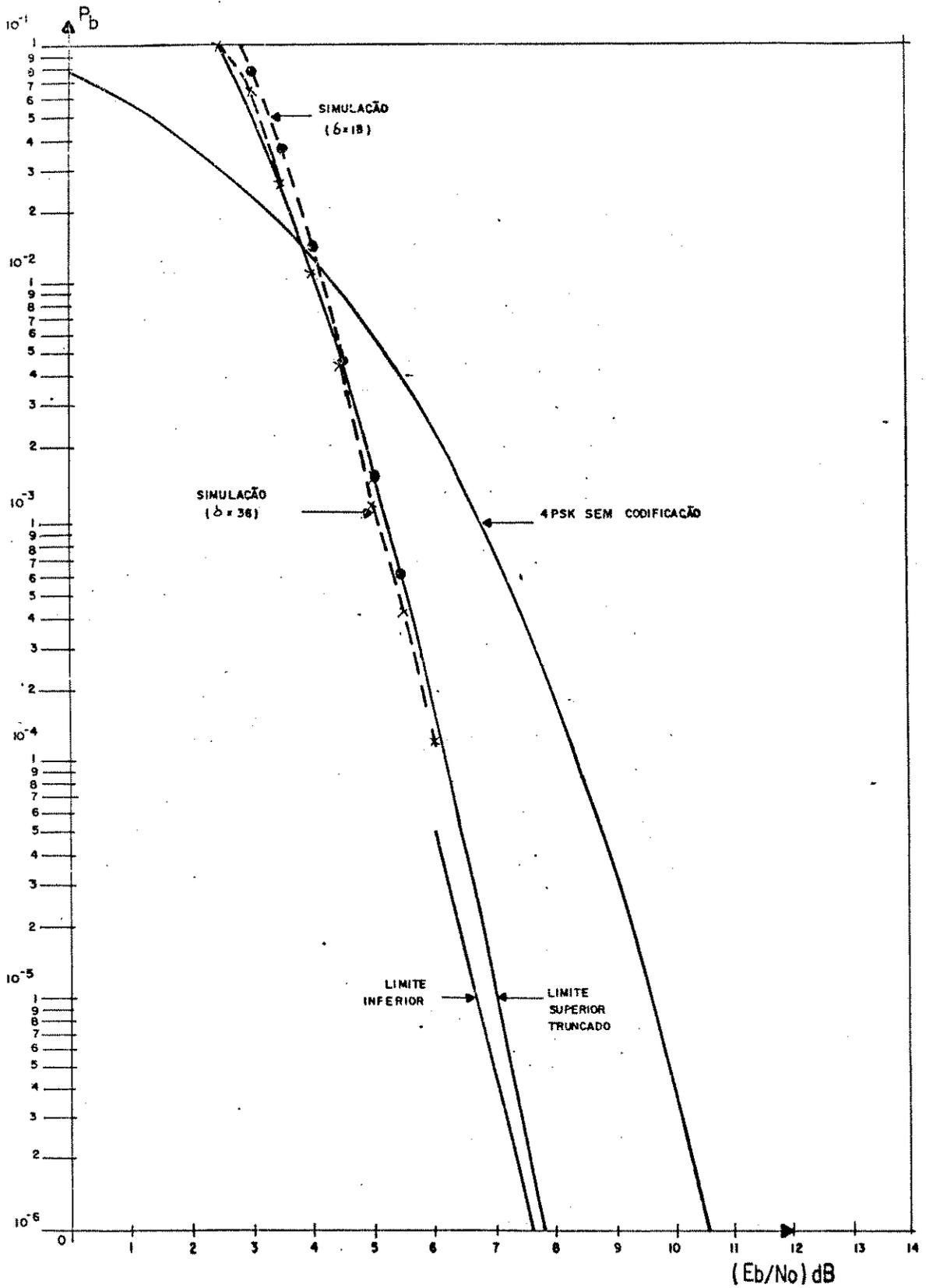


Fig. 7.7: Probabilidade de erro de bit para o 4PSK codificado no 8PSK usando o código da Eq. 7.18 e o MPC da Fig. 6.8. Simulação para $\delta = 18$ e $\delta = 36$ símbolos.

7.5 - Simulação do código (4 6 6 1 2) para codificação do sistema 4PSK no 8PSK

As Figs. 7.8 e 7.9 apresentam os resultados da simulação para o código com matriz geradora

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (7.22)$$

que corresponde ao código ótimo obtido no Cap.6, seção 6.9, para codificação do 4PSK no 8PSK.

Os resultados obtidos mostram que P_b , como para o código caracterizado por (7.18), é pouco sensível ao aumento no atraso de codificação de 18 para 36 símbolos. Entretanto, mesmo para $n=36$ símbolos, os resultados estão deslocados à direita dos limites superiores truncados por valores que variam de 0,2 a 0,4 dB para P_{av} e de 0,1 a 0,4 dB para P_b , fato que não ocorre para o código de (7.18). Uma possível explicação para o fenômeno é que os parâmetros d_i , N_i e W_i obtidos em relação à trajetória de vetores nulos não apresentam a realidade do código em relação a outras trajetórias na treliça associada ao mesmo. Uma comparação entre os resultados obtidos para os dois códigos e apresentados conjuntamente na Fig. 7.10 parece comprovar essa hipótese. Um outro fato que deve ser levado em consideração é que o procedimento de otimização assegura melhores desempenhos para altas relações sinal/ruído já que, como foi visto no Cap. 4, são minimizados os coeficientes dos termos correspondentes às menores métricas. Apesar de ser proibitivo a simulação em altas relações sinal ruído, foi observado que para uma razão de 9,0 dB o código (7.22) não gerou qualquer evento de erro em um experimento de 100.000 símbolos, enquanto que o código (7.18) gerou 5 eventos de erro. Além desse fato, P_b passa a ser menor para o código (7.22) a partir de SNR=8,5 db. Assim, esses fatos confirmam o efeito esperado pela otimização do código.

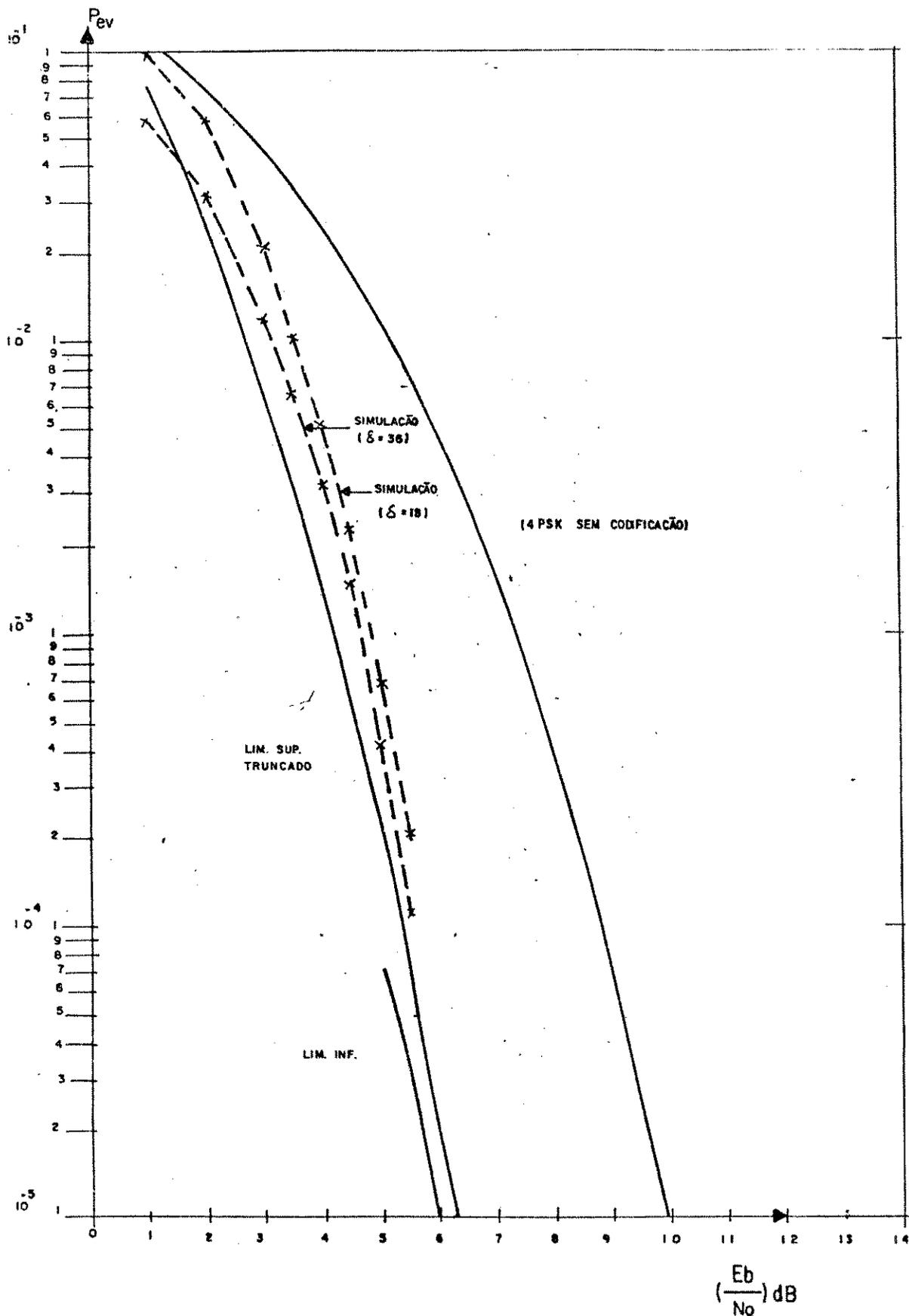


Fig. 7.8: Probabilidade de erro de evento para o 4PSK codificado no 8PSK usando o código da Eq. 7.22 e o MPC da Fig. 6.8. Simulação para $\delta = 18$ e $\delta = 36$ símbolos.

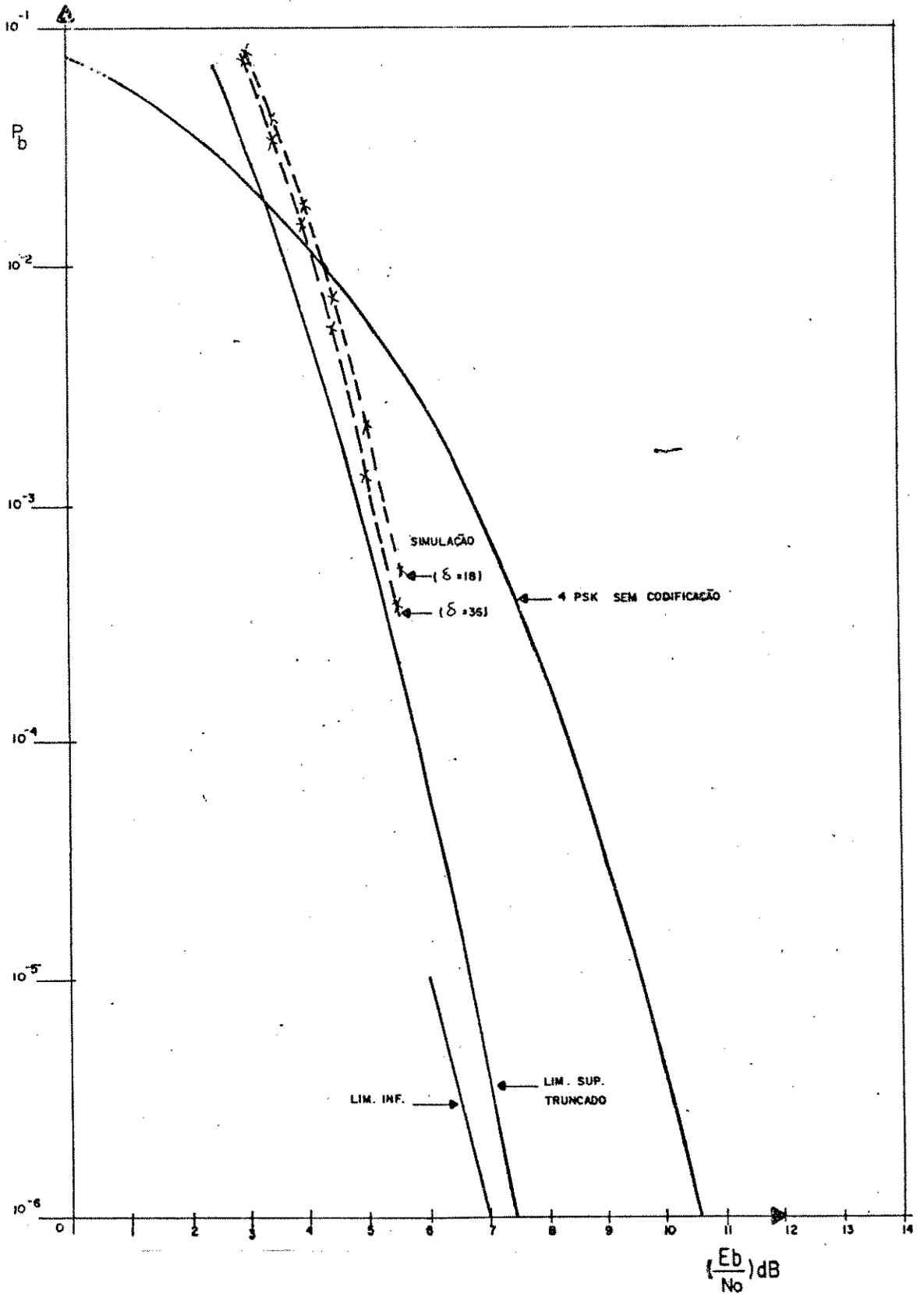


Fig. 7.9: Probabilidade de erro de bit para o 4PSK codificado no BPSK usando o código da Eq. 7.22 e o MPC da Fig. 6.8. Simulação para $\delta = 18$ e $\delta = 36$ símbolos.

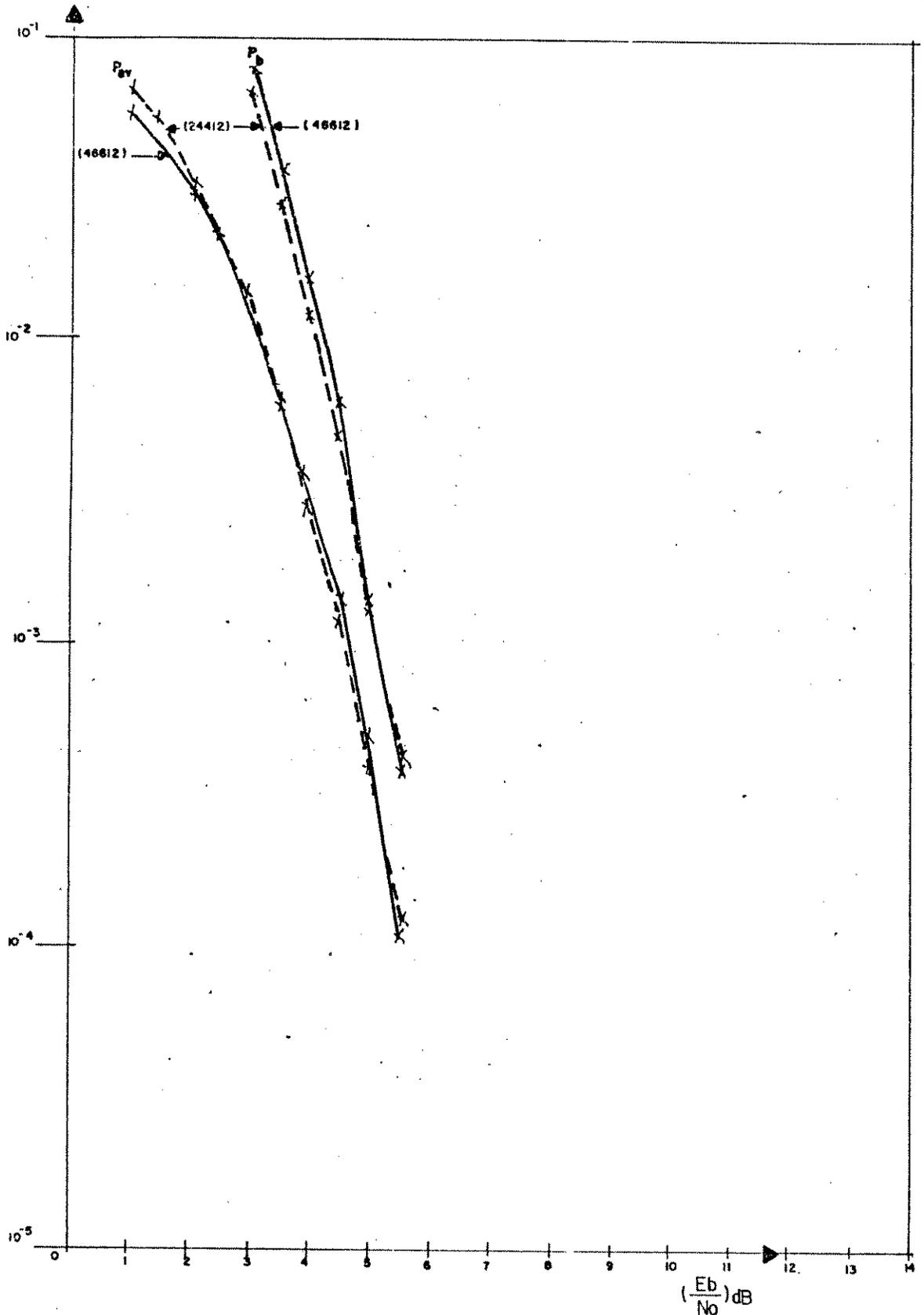


Fig. 7.10: Probabilidades de erro de evento e de erro de bit para o 4PSK codificado no 8PSK. Comparação entre os códigos da Eq. 7.18 e da Eq. 7.22, para o MPC da Fig. 6.8. Simulação para $\delta = 36$ símbolos.

7.6 - Simulação do código (2 4 6 5 4) para codificação do sistema 4PSK no OCTAL CÚBICO

Os resultados ilustrados nas Figs. 7.11 e 7.12 foram obtidos pela simulação do código com matriz geradora

$$G = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (7.23)$$

para codificação do 4PSK no OCTAL CÚBICO. A análise dos resultados mostra que não há melhora de desempenho ao se aumentar o atraso de decodificação de 18 para 36 símbolos, ao contrário do que ocorre para a codificação do 4PSK no BPSK, principalmente para $P_{e\text{v}}$. As pequenas diferenças existentes entre os resultados da simulação e os limites superiores truncados têm as mesmas explicações feitas anteriormente, sendo que aqui o número de termos usado nos limites superiores truncados foi bem menor que para o caso da codificação do 4PSK no BPSK. Os resultados obtidos na simulação comprovam que, com a mesma complexidade de decodificação, o sistema 4PSK codificado no OCTAL CÚBICO apresenta uma economia de potência da ordem de 0,7dB em relação ao 4PSK codificado no BPSK. E mais, esse incremento só é conseguido no sistema 4PSK/BPSK se for duplicada a complexidade de decodificação.

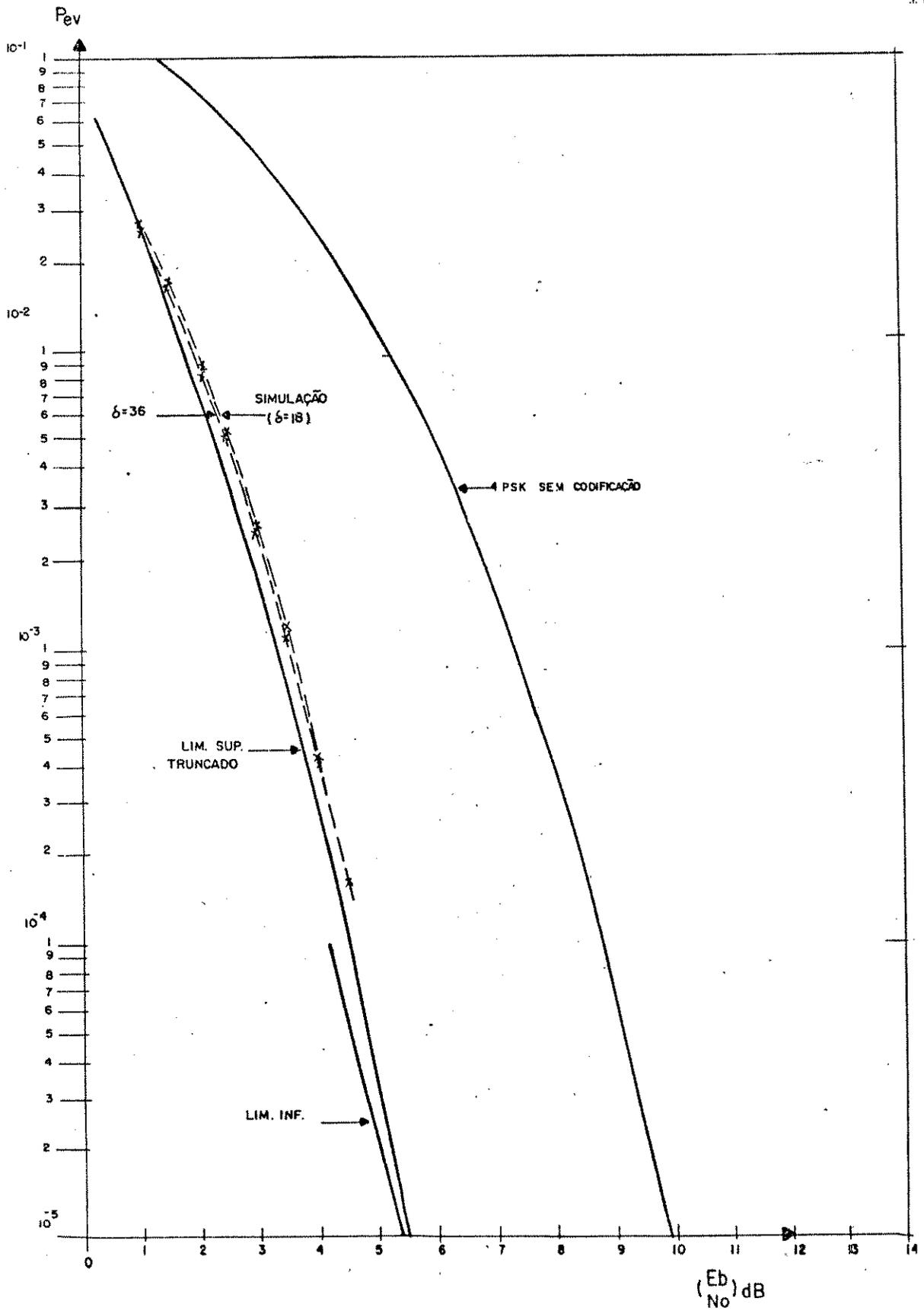


Fig. 7.11: Probabilidade de erro de evento para o sistema 4PSK codificado no OCTAL CÚBICO usando o código da Eq. 7.23. Simulação com $\delta = 18$ e $\delta = 36$ símbolos.

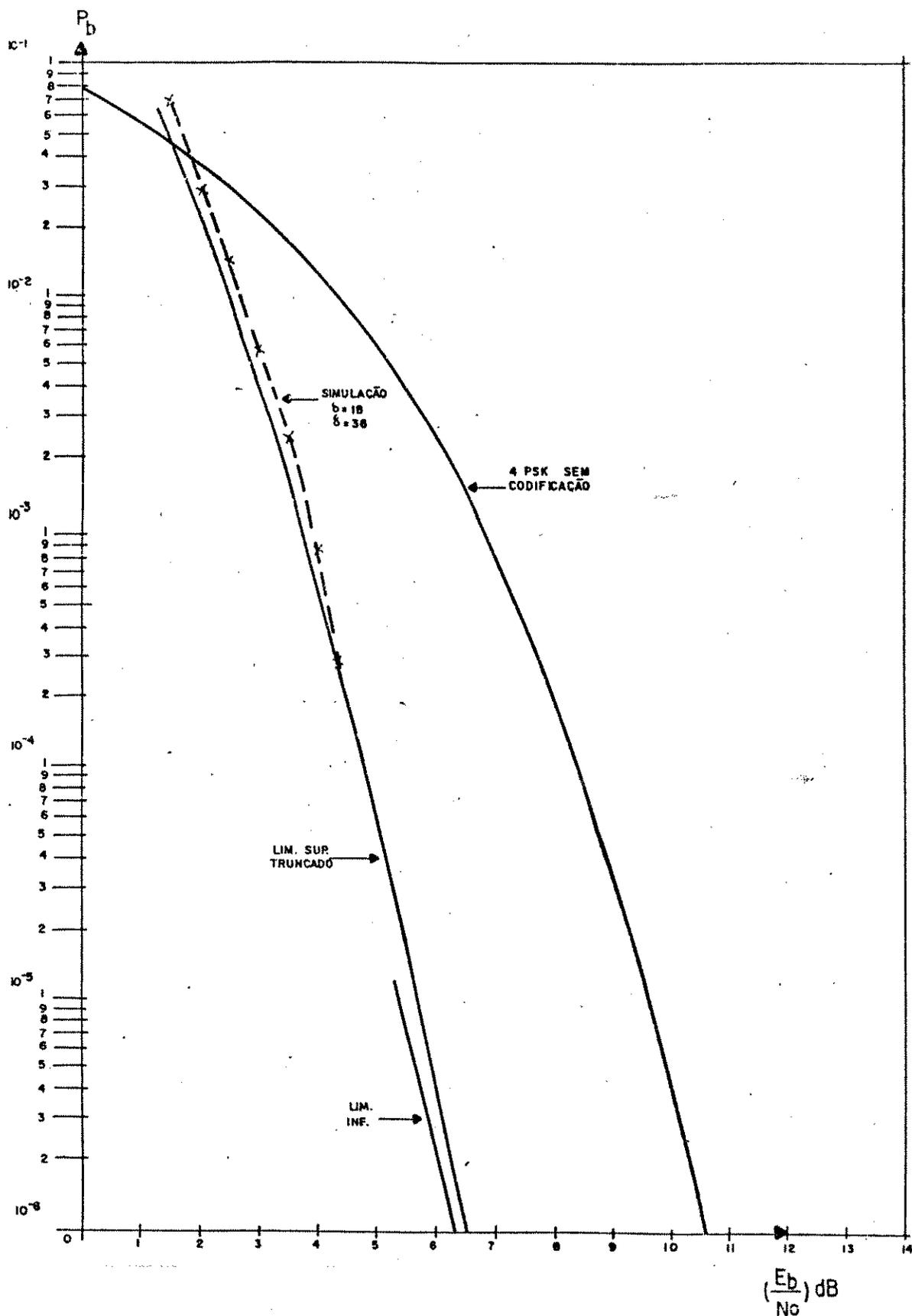


Fig. 7.12: Probabilidade de erro de bit para o sistema 4PSK codificado no OCTAL CÚBICO usando o código da Eq. 7.23. Simulação com $\delta = 18$ e $\delta = 36$ símbolos.

CAPÍTULO 8

CONCLUSÕES E COMENTÁRIOS

Neste trabalho de dissertação foram desenvolvidos modelos e procedimentos que possibilitaram a investigação do uso de constelações tridimensionais em sistemas de modulação codificada controladas por códigos convolucionais. Foi desenvolvido um conjunto de programas com base nos modelos apresentados ao longo do trabalho e que permitiram a investigação do principal objetivo deste trabalho, descrito no Cap. 1. A Fig. 8.1 ilustra em diagrama de blocos os programas desenvolvidos e sua interdependência, segundo a ordem cronológica em que o trabalho foi desenvolvido. Apesar de terem sido desenvolvidos de forma independente, sua integração pode ser realizada de forma relativamente simples, já que no todo formam um conjunto de etapas sequenciais.

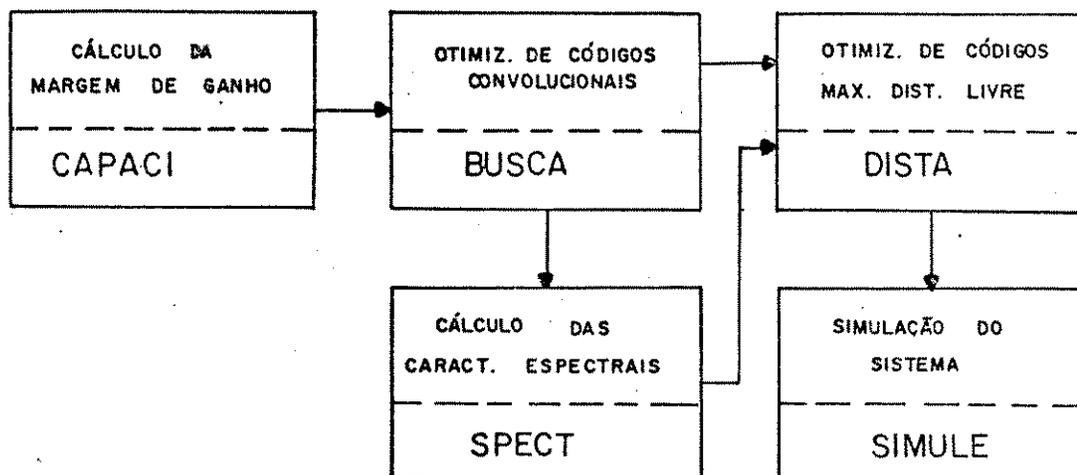


Fig. 8.1: Programas desenvolvidos e sua interdependência de acordo com a seqüência de desenvolvimento do trabalho.

Quanto ao objetivo principal do trabalho, os resultados obtidos demonstraram que com uma complexidade mínima de decodificação, a constelação tridimensional estudada proporciona ganhos de potência que só são obtidos para a correspondente constelação bidimensional com a duplicação ou triplicação da complexidade de decodificação (Cap. 6). O preço pago, como era de se esperar, é o aumento da faixa de transmissão. Porém como foi visto nos Caps. 5 e 6, o seu efeito depende principalmente da definição usada para a faixa de transmissão.

Entretanto, alguns estudos ainda terão que ser realizados no que se refere ao efeito de filtragem de canal sobre o desempenho do sistema, principalmente para o caso de canais não

lineares. Um estudo nesse sentido foi iniciado mas foi abandonado por limitações nos recursos computacionais disponíveis. Um outro problema a ser estudado é a implementação em "hardware" do modelo proposto para a decodificação apresentado no Cap. 3. Também merece ser investigada a possibilidade do uso de constelações tridimensionais assimétricas no sentido de buscar um possível aumento da eficiência dos códigos. Estudos neste sentido foram realizados para constelações com pontos em uma e duas dimensões (Divsalar e Simon, 1985). Finalmente, como o procedimento de otimização de códigos com máxima distância apresentado neste trabalho usou como métrica as distâncias Euclidianas reais entre os símbolos da constelação código, é interessante usar o procedimento apresentado no apêndice A4, considerando como métrica os pesos definidos em (4.20) que asseguram a superlinearidade dos códigos.

REFERÊNCIAS BIBLIOGRÁFICAS

- Anderson, J. B. & Taylor, D. P. (1978). "A Bandwidth-Efficient Class of Signal-Space Codes". *IEEE Trans. Inform. Theory*. Vol. IT-24, no 6: 703-712.
- Anderson, J. B. et al., (1981). "Power-Bandwidth Performance of Smoothed Phase Modulation Codes". *IEEE Trans. Commun.* Vol. COM-29, no 3: 187-195.
- Aulin, T. & Sundberg, C. W. & Rydbeck, N., (1981). "Continuous Phase Modulation - Part I: Full Response Signalling; Part II: Partial Response Signalling". *IEEE Trans. Commun.* Vol. COM-29, no 3: 196-225.
- Bahl, L. R. et al., (1972). "An Efficient Algorithm for Computing Free Distance". *IEEE Trans. Inform. Theory*, Vol. IT-18, no 5: 437-439.
- Bezerra, J. B. & Arantes, D. S., (1984). "Análise Espectral em Sistemas Digitais com Modulação Codificada Linearmente". *Anais do Segundo Simpósio Brasileiro de Telecomunicações*. : 119-125.
- Bezerra, J. B. & Arantes, D. S., (1985). "Optimization of Convolutional Codes for Encoded Modulation". *Anais do Second IASTED- International Symposium in Telecommunications and Control, IELECON'85*. : 58-61.
- Bezerra, J. B. & Arantes D. S. (1986a). "Capacidade de Canal para Esquemas de Modulação Digital em Espaços Multidimensionais". *Publicação Interna da EEC-UNICAMP*, no 36.
- Bezerra, J. B. & Arantes, D. S., (1986b). "Spectral Analysis of q-ary Digital Signals Encoded by p-ary Convolutional Codes." *IEEE Trans. on Inform. Theory*. Vol. IT-32, no 2: 313-319.
- Billardi, G. & Padovani, R. & Pierobon, G. L. (1983). "Spectral Analysis of Functions of Markov Chains with Applications". *IEEE Trans. on Commun.* Vol. COM-31, no. 7: 853-860.
- Borelli, W. G., (1983). "Convolutional Codes for Multilevel Data Transmission". *PhD Thesis, Electronic Laboratories University of Kent, Canterbury, England*.
- Calderbank, R. J. & Mazo, J. E. (1984). "A New Description of Trellis Codes". *IEEE Trans. Inform. Theory*. Vol. IT-30, no 6: 784-791.
- Carlolaro, G. L. & Tronca, G. P., (1974). "Spectra of Block Coded Digital Signals". *IEEE Trans. Commun.* Vol. COM-22, no. 10: 1555-1563.

- Chambers, P. P. (1967). "Random-number Generation". IEEE Spectrum. no 2.
- Clark Jr., G. C. & Cain, J. B. (1981). Error Correction Coding for Digital Communications. Plenum Press, New York.
- Divsalar, D. & Simon, M. K. (1985). "Combined Trellis Coding with Asymmetric Modulations". Digest of GLOBECOM'85: 21.2.1-21.2.7.
- Forney Jr., G. D., (1970). "Convolutional Codes I: Algebraic Structure". IEEE Trans. Inform. Theory. Vol. IT-16, no. 6: 720-738.
- Forney Jr., G. D., (1972). "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference". IEEE Trans. Inform. Theory, Vol. IT-18, no 3: 363-378.
- Forney Jr., G. D., (1973). "The Viterbi Algorithm". Proceedings of IEEE. Vol. 61: 268-278.
- Forney Jr., G. D. e outros, (1984). "Efficient Modulation for Band-Limited Channels". IEEE Journal on Selected Areas in Commun., Vol. SAC-2, no 5: 632-647.
- Gallager, R. G. (1965). Information Theory and Reliable Communication. New York, Wiley.
- Gersho, A. & Lawrence, V. B., (1984). "Multidimensional Signal Constellations for Voiceband Data Transmission". IEEE Journal on Selected Areas in Commun., Vol. SAC-2, no 5: 687-702.
- Heller, J. A. & Jacobs, I. M., (1971). "Viterbi Decoding for Satellite and Space Communication". IEEE Trans. Commun., Vol. COM-19, no 5: 835-848.
- Hull, T. E. and Dobell, A. R., (1962). "Random Number Generators". SIAM Review. Vol. 42, no. 3.
- Kohavi, Z., (1970). Switching and Finite Automata Theory. McGraw-Hill Computer Science Series.
- Marsen, M. A. & outros, (1984). "Theoretical Aspects and Performance Evaluation of Channel Encoding with Redundant Modulator Alphabets". Digest of GLOBECOM'84: 20.5.1-20.5.5
- Mihram, G. A., (1972). Simulation, Statistical Foundations and Methodology. New York, Academic Press.
- Payne, W. H. & Rabung, J. R. & Boggio, T. P., (1969). "Coding the Lhemer Pseudo-random Number Generator". Communication of the ACM. Vol. 12, no 2: 85-86.

- Peterson, W. W. & Weldon Jr., E. J., (1972). Error Correcting Codes. The MIT press, Cambridge, Massachusetts.
- Prabhu, V. K. & Rowe, H. E. (1974). "Spectra of Digital Phase Modulation by Matrix Methods". *B.S.I.J.* Vol. 53, no 5: 899-935.
- Prabhu, V. K. & Rowe, H. E. (1975). "Power Spectrum of a Digital, Frequency-Modulation Signal". *B.S.I.J.* Vol. 54, no 6: 1095-1125.
- Taylor, D. P. & Chan, H. C. (1981). "A simulation Study of two Bandwidth-Efficient Modulation Techniques". *IEEE Trans. Commun.* Vol. COM-29, no 3: 267-275.
- Ungerboeck, G., (1982). "Channel Coding with Multilevel/Phase Signals". *IEEE Trans. Inform. Theory*, Vol. IT-28, no 1: 55-67.
- Wilson, S. G. & Gaus, R. C. (1981). "Power Spectra of Multi-h Phase Codes". *IEEE Trans. Commun.* Vol. COM-29, no 3: 250-256.
- Wozencraft, J. M. & Jacobs, I. M. (1965). *Principles of Communication Engineering*. New York, Wiley.