



UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA  
DEPARTAMENTO DE TELEMÁTICA

"UMA CONTRIBUIÇÃO À FORMA COMBINADA DE  
CODIFICAÇÃO E MODULAÇÃO UTILIZANDO  
CÓDIGOS DE MEMÓRIA UNITÁRIA"

AUTOR : *Valdir Aparecido Gurian*  
ORIENTADOR : Prof. Dr. REGINALDO PALAZZO JR.

Tese apresentada à Faculdade de Engenharia  
Elétrica, da Universidade Estadual de  
Campinas, UNICAMP, como parte dos  
requisitos exigidos para a obtenção do  
título *Mestre em Engenharia Elétrica*.

02/01/00735

Dezembro 1989

Este exemplar corresponde à redação final da tese  
defendida por *Valdir Aparecido Gurian*  
e aprovada pela Comissão  
Julgada em *21/12/1989*.  
*Reginaldo Palazzo Jr.*  
Orientador



"à memória de meus avós  
maternos, por tudo o  
que sempre representa-  
rão".



## *Agradecimentos*

Ao Prof. Reginaldo Palazzo Jr., meu orientador, por ter me concedido a oportunidade de realizar este trabalho, e também pelas contribuições e estímulos necessários para que pudesse completá-lo:

A Eng<sup>a</sup>. Ivonete Markman, autora dos programas computacionais usados neste trabalho, pelas contribuições, trocas de idéias e incentivo ao longo do desenvolvimento deste:

Aos Profs. dos Departamentos de Telemática e Comunicação, pelos seus ensinamentos durante o período de curso das disciplinas que compuseram a base teórica para o desenvolvimento deste trabalho:

A todos os companheiros do Curso de Mestrado, pela sua amizade e afeição:

Ao CNPq, pelo apoio financeiro, durante a realização deste trabalho:

Aos Professores: Dr. Walter C. Borelli e Dr. Weiler Alves Finamore, que compuseram a banca examinadora deste trabalho:

Enfim, a todas as pessoas que, de uma forma direta ou indireta, contribuíram para que eu chegasse até aqui.

## RESUMO

Este trabalho mostra que o desempenho de códigos TCM com estrutura de memória unitária, de  $k$  entradas e  $n$  saídas, associados a constelações em duas dimensões, é inferior ao desempenho de qualquer outro código TCM com mesmo comprimento total de memória (CTM), mesmo número de entradas ( $k$ ) e mesmo número de saídas ( $n$ ), associado à mesma constelação.

Mostra também que uma modulação ótima não necessariamente apresentará o melhor desempenho quando estiver compondo um esquema combinado (modulação/codificação), pelo não casamento com o codificador.

E ainda, faz uma análise comparativa para o desempenho dos códigos TCM entre diferentes constelações, em duas dimensões, e diferentes estruturas de memória do codificador, levantando características para essa forma combinada de comunicação.



## ABSTRACT

This work shows that the performance of TCM codes with structure of unitary memory, of "k" inputs and "n" outputs, associated to constelations in two dimensions, is worse than the performance of any other TCM code with the same total memory length (CTM), the same number of inputs (k) and the same number of outputs (k) associated to the same constelation.

It also shows that the optimal modulation not necessarily will present the best performance when compounding a combined scheme (Modulation/Codification), due to the mismatch with the encoder.

Finally, it is a mode comparative analysis to the performance of the TCM codes between different constelations, in two dimensions, and different structures of memory of the coder, getting up characteristics to this combined way of communication.



## ÍNDICE

INTRODUÇÃO .....	7
1. TÓPICOS EM MODULAÇÃO DIGITAL	
1.1 - Introdução.....	12
1.2 - Esquemas de Modulação Digital .....	19
2. OS CÓDIGOS CONVOLUCIONAIS BINÁRIOS	
2.1 - Introdução .....	31
2.2 - A Codificação dos Códigos Convolucionais .....	31
2.3 - A Representação Matricial de um Codificador Convolucional .....	37
2.4 - O Diagrama de Estados e a Função de Transferência .....	42
2.5 - O Diagrama de Treliça .....	47
2.6 - A Distância Mínima de um Código Convolucional .....	49
2.7 - A Decodificação por Máxima Verossimilhança e o Algoritmo de Viterbi .....	51
2.8 - Os Limitantes Superiores da Probabilidade de Erro de Evento e de Bit .....	55
2.9 - A Representação de um Codificador Convolucional por Vetores Coeficientes .....	60
2.10 - Os Códigos Convolucionais Binários de Memória Unitária .....	62

### 3. OS CÓDIGOS TCM

3.1 - Introdução .....	68
3.2 - O Processo de Codificação para Códigos TCM .....	69
3.3 - A Partição do Conjunto de Sinais (Set Partitioning) ...	70
3.4 - A Distância Mínima do Código TCM .....	75
3.5 - A Estrutura dos Códigos TCM .....	78
3.6 - O Desempenho de Sistemas de Comunicação Codificados ...	80

### 4. A PROCURA DOS CÓDIGOS TCM DE DESEMPENHO ÓTIMO

4.1 - Introdução .....	84
4.2 - Os Objetivos da Procura .....	84
4.3 - O Processo de Procura .....	88
4.4 - Restrições à Procura .....	96

### 5. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

5.1 - Introdução .....	98
5.2 - Análise Comparativa .....	114

CONCLUSÕES E SUGESTÕES PARA FUTUROS TRABALHOS .....	124
---	-----

REFERÊNCIAS BIBLIOGRÁFICAS .....	128
----------------------------------	-----

APÊNDICE I .....	132
------------------	-----



## INTRODUÇÃO

Os sistemas de comunicações digitais, geralmente, estão sujeitos a limitações tanto de potência quanto de faixa de transmissão. Então, para atender à crescente demanda de serviços de comunicação, satisfazendo as exigências de confiabilidade da informação, são necessárias técnicas de transmissão cada vez mais eficientes.

Observou-se que a utilização de formas de modulação se mostravam eficientes quanto à economia de largura de faixa, exigindo porém maior potência para manter a confiabilidade. Uma solução para esta limitação em potência constituiu-se na utilização de códigos corretores e/ou detectores de erro. No entanto, o uso destes códigos implicava em um aumento da largura de faixa requerida para uma mesma taxa de informação. Assim, verificava-se que a visão em separado dessas duas técnicas nem sempre levava a um resultado encorajador. Massey, [23], foi um dos primeiros a mostrar que se poderia conseguir uma melhoria considerável de desempenho ao se tratar de modulação e codificação como entidade única.

Ungerboeck, [19], considerando a codificação convolucional e a modulação no espaço de sinais como entidade única, introduziu um novo conceito em teoria de comunicações, que aqui é referido como Modulação por Codificação em Treliça (TCM - Trellis-Coded Modulation). O uso de códigos TCM permite alcançar ganhos de codificação assintóticos de 3dB para esquemas simples, e de até 6dB






para esquemas mais complexos, em relação aos sistemas convencionais não codificados, sem alterar a taxa de transmissão de bits de informação, além de manter a largura de faixa.

Assim, o estudo aqui realizado se detém na avaliação do desempenho de códigos TCM, em comparação a sistemas não codificados.

Analisando o desempenho de sistemas de comunicações não codificados, Forney, [9], estabeleceu que o desempenho de um sistema de modulação (constelação) será tanto maior quanto mais esta constelação se aproximar de um círculo (Seção 1.2.2). Então, uma questão levantada neste trabalho é a verificação da validade ou não desta característica para os sistemas combinados (modulação/codificação), em especial para os códigos TCM. Assim, para uma mesma classe simples de codificadores  $(G_{kn}; m_1, m_2, \dots, m_k)$  buscar-se-á determinar os melhores códigos TCM para diferentes constelações, estabelecendo-se em seguida a comparação entre os desempenhos destes sistemas combinados em relação a sistemas não codificados.

Por outro lado, Lee, [10], estudando apenas a codificação convolucional, estabeleceu que para qualquer codificador convolucional binário  $(G_{kn}; M)$ , ou seja, um codificador convolucional binário com 'k' entradas, 'n' saídas e comprimento total de memória 'M', existe um codificador equivalente com estrutura de memória unitária:  $(G_{kn}; 1, 1, 1, \dots)$  com  $k' = Mk$  entradas,  $n' = Mn$  saídas e mesmo comprimento total de memória M (veja Seção 2.9), e que assim a procura dos melhores códigos para a classe  $(G_{kn}; M)$  pode restringir-se aos códigos de memória unitária, em outras palavras: para cada classe simples de codificadores convolucionais binários existe sempre um codificador com estrutura de memória unitária que é ótimo no sentido



de máximo  $d_{free}$  [distância livre (Seção 2.6.1)]. Portanto, a segunda e principal questão que se levanta neste estudo é se esta característica, válida para os códigos convolucionais binários, é ou não válida para os códigos TCM.

Por fim, uma terceira questão é analisada: para os mesmos códigos TCM de memória unitária, iremos analisar o desempenho quando um fator de multiplicidade de taxas é considerado, isto é, comparar os códigos TCM de memória unitária elementares, de taxa  $R=k/n$ , com os respectivos múltiplos, de taxa  $R=\alpha k/\alpha n$ , para  $\alpha \geq 2$ .

Para responder-se a estas questões, o procedimento a ser utilizado será o de buscar para cada classe simples de codificadores  $\{G_{kn}; m_1, \dots, m_k\}$  o codificador TCM ótimo, no sentido de máximo  $\Delta_{min}$  (Definição 4.1), através do algoritmo proposto em [12]. Em linhas gerais, o desenvolvimento do trabalho é apresentado da seguinte forma:

Capítulo 1 - Apresentação de tópicos em Modulação Digital que serão necessários para o entendimento dos códigos TCM;

Capítulo 2 - Revisão de conceitos da teoria de codificação convolucional, que servirão de base para os Capítulos subsequentes;

Capítulo 3 - Descrição dos códigos TCM, discutindo-se os processos de codificação e decodificação, propriedades estruturais e o desempenho destes em relação a sistemas não codificados;



Capítulo 4 - Apresentação do procedimento usado na procura dos códigos TCM ótimos, bem como os objetivos desta busca;

Capítulo 5 - Apresentação dos resultados obtidos na procura do Capítulo 4, discutindo-os e procurando responder às questões levantadas neste trabalho;

Capítulo 6 - Conclusões e sugestões para futuros trabalhos a serem desenvolvidos nesta área.

Apêndice - Apresentação do "Set Partitioning" das constelações aqui estudadas.



## CAPÍTULO 1

TÓPICOS EM MODULAÇÃO DIGITAL



## 1.1 - INTRODUÇÃO

Neste capítulo apresentamos técnicas e conceitos básicos de modulação digital que serão necessários para um bom entendimento dos códigos TCM (Trellis Coded Modulation) descritos no Capítulo 3, que são elementos de estudos deste trabalho.

Assim, inicialmente discorreremos a respeito dos fundamentos de um sistema de comunicação digital, fazendo uma sucinta explanação de seu funcionamento através do modelo por diagrama de blocos. Em seguida mostramos como se realiza a geração de sinais a serem transmitidos e a representação destes como pontos no espaço de sinais (Constelação de Sinais); posteriormente apresentamos os esquemas de modulação digital mais freqüentemente utilizados e que serão abordados nos capítulos seguintes, com uma caracterização de energia associada a estes esquemas. Finalmente uma análise de avaliação do desempenho destes sistemas é apresentada, no sentido de se poder traçar comparações entre diferentes esquemas de modulação digital.

### 1.1.1 - O Modelo de um Sistema de Comunicação Digital

A Figura 1.1 mostra o modelo por diagrama de blocos de um sistema de comunicação digital:

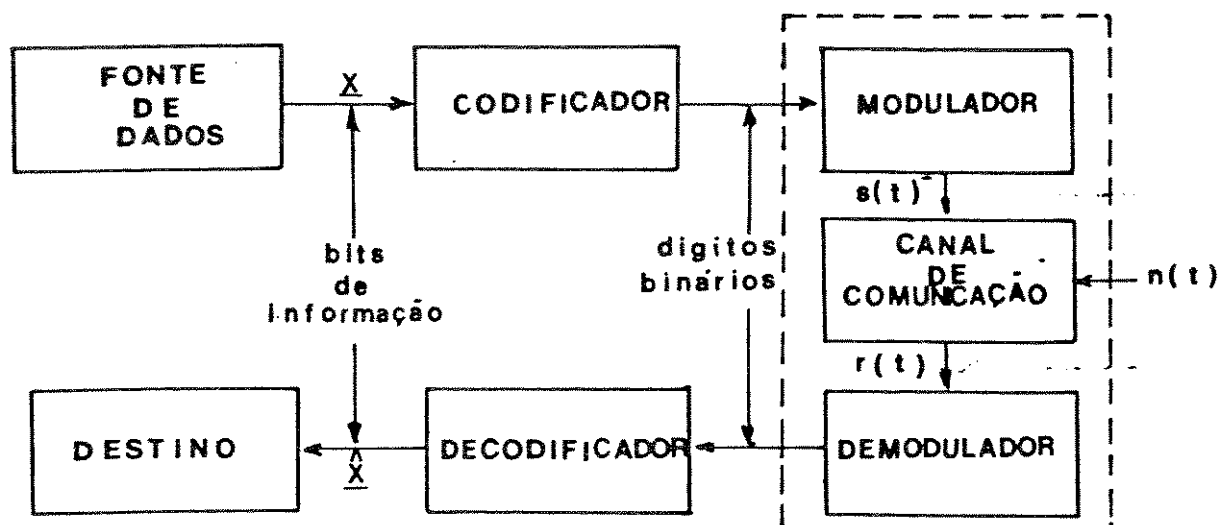


Figura 1.1 - Modelo de um sistema de comunicação digital

Podemos descrever o funcionamento do sistema apresentado na Figura 1.1 de uma forma resumida, como se segue:

A FONTE DE DADOS gera dados na forma de símbolos binários. A suposição normal é que esses dados são processados tal que "uns" e "zeros" ocorram independentemente e com igual probabilidade, [5]. Neste trabalho tal suposição sempre será observada.

A seqüência de informação  $X$  entra no CODIFICADOR onde será acrescentada a redundância, com a finalidade de corrigir e/ou detectar erros que venham ocorrer durante a transmissão. O codificador

de um sistema de comunicação digital é o objeto de estudo deste trabalho.

Depois de codificada, a seqüência passa para o MODULADOR, que gera um conjunto de formas de onda de duração limitada realizando um mapeamento entre esta seqüência e o conjunto de formas de onda. Para um esquema de modulação binário, cada símbolo de saída do codificador é usado para selecionar uma das duas formas de onda possíveis, enquanto que para um esquema  $M^{\text{ario}}$ , a saída do codificador é segmentada em conjuntos de 'j' símbolos (em que  $M = 2^j$ ) e cada conjunto é usado para selecionar uma das M formas de onda, [2].

Assim, o sinal analógico  $s(t)$  será transmitido pelo CANAL DE COMUNICAÇÃO, que fornece uma conexão elétrica entre a saída do modulador e a entrada do demodulador. Este pode ser um par de fios, ou um "link" telefônico ou o espaço livre sobre o qual o sinal será irradiado. Devido a limitações físicas, o canal tem largura de faixa finita (B Hz), e o sinal de informação conduzido sofre muitas vezes distorção em amplitude e fase, [1]. Considera-se que:

- $h(t)$  é a função que modela o canal;
- $n(t)$  é um processo estocástico estacionário, Gaussiano branco de média zero;
- o canal é do tipo aditivo

Assim:

$$r(t) = s(t) + n(t)$$

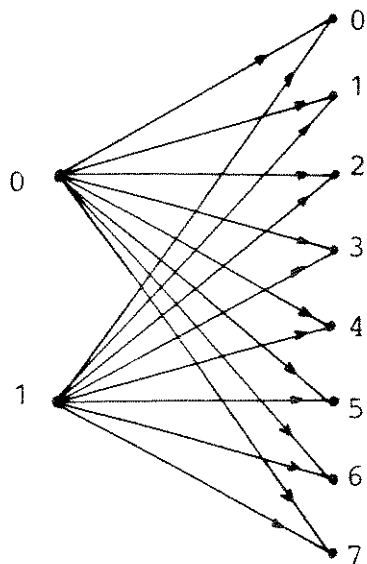
Então, a partir do sinal recebido  $r(t)$  o DEMODULADOR decide qual sinal foi transmitido. Considera-se que o demodulador está em perfeito sincronismo de fase e freqüência com o modulador.

Finalmente o DECODIFICADOR busca recuperar a informação original a partir das mensagens codificadas ao retirar a redundância acrescentada na codificação, corrigindo e/ou detectando alguns dos erros que possam ter ocorrido durante a transmissão e uma seqüência decodificada  $\hat{X}$  chega ao destino.

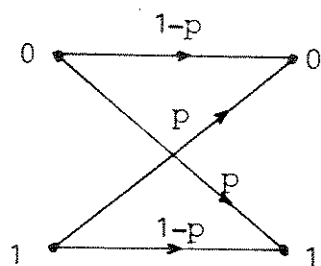
Se olharmos o modulador, o canal de comunicação e o demodulador como um único bloco, temos o canal discreto de dados que é caracterizado por um conjunto de símbolos de entrada, símbolos de saída e probabilidades de transição. No caso mais simples, as probabilidades de transição são invariantes no tempo e independentes símbolo a símbolo. Neste caso, o canal é chamado canal discreto sem memória (DMC - Discrete Memoryless Channel), [5].

A Figura 1.2a ilustra a representação de um DMC com entrada binária e saída octária.

O caso mais comum de um DMC é o canal binário simétrico (BSC - Binary Symmetric Channel) representado na Figura 1.2b.



(a) DMC com entrada binária e saída octária



(b) BSC com probabilidade de transição P

Figura 1.2



Observa-se que para o BSC (Figura 1.2b) não há quantização introduzida, pois os mesmos dois níveis da entrada comparecem na saída, o que leva na recepção a uma decisão abrupta (hard decision).

Já para um canal onde há um maior número de níveis de saída do que os de entrada (como o exemplo da Figura 1.2a) há introdução de uma quantização produzida por um quantizador de oito níveis, o que levará a uma decisão suave (soft decision), onde, como no exemplo, os níveis enumerados 0, 1, 2 e 3 serão decodificados como '0' e os níveis 4, 5, 6 e 7 como '1'.

A relevância dos esquemas quantizados é que estes levam a uma probabilidade de erro na decodificação tanto menor quanto maior for a quantização, porém, fazendo com que a complexidade seja grande.

### 1.1.2 - A Geração de Sinais e os Pontos no Espaço de Sinais (Constelação de Sinais)

#### 1.1.2.1 - A Geração de Sinais

Podemos representar o sistema de comunicações da Figura 1.1 de uma forma simplificada, conforme mostra a Figura 1.3:

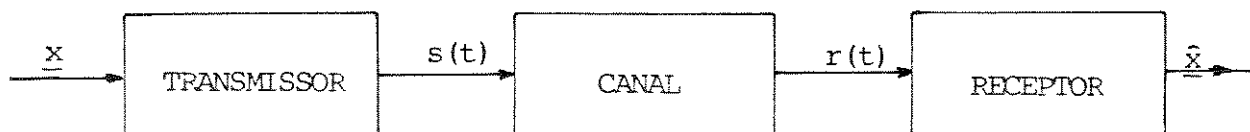


Figura 1.3 - Modelo simplificado de um sistema de comunicações

O transmissor contém o codificador e o modulador, enquanto o demodulador e o decodificador fazem parte do receptor.

Seja  $\{x_i\}$  o conjunto de todas as mensagens de entrada e  $\{s_i(t)\}$  o conjunto discreto de formas de onda a serem transmitidas, onde  $i = 0, 1, \dots, M - 1$ .

Assim, a cada mensagem de entrada  $x = x_i$  teremos correspondentemente um sinal  $s(t) = s_i(t)$  na entrada do canal. Deste modo, então, nos interessa a forma de geração de cada  $s_i(t)$  e sua representação.

Podemos representar cada  $s_i(t)$ ,  $0 \leq i \leq M - 1$ , como uma combinação linear de  $N$  ( $N < M$ ) formas de onda  $\psi_j(t)$ ,  $1 \leq j \leq N$ , em que  $\{\psi_j(t)\}$  é o conjunto de formas de onda ortogonais, isto é:

$$\int_{-\infty}^{\infty} \psi_j(t) \psi_l(t) dt = \begin{cases} 1, & \text{se } j = l \\ 0, & \text{se } j \neq l \end{cases} \quad \text{para todo } j \text{ e todo } l \quad (1.1)$$

então

$$s_i(t) = \sum_{j=1}^N s_{i,j} \psi_j(t) \quad , \quad 0 \leq i \leq M - 1 \quad (1.2)$$

onde:

$$s_{i,j} = \int_0^T s_i(t) \psi_j(t) dt \quad (1.3)$$

onde  $T$  é o período de duração de  $s_i(t)$

Em resumo: o conjunto das N-uplas  $s_i$

$$s_i = (s_{i1}, s_{i2}, \dots, s_{iN}) \quad , \quad 0 \leq i \leq M - 1 \quad (1.4)$$

representa o conjunto de sinais  $s(t) = \{s_i(t)\}$ , ou seja, os N valores  $s_{i1}, s_{i2}, \dots, s_{iN}$  são as projeções do sinal  $s_i(t)$  no espaço Euclidiano N-dimensional. Os M vetores  $\{s_i(t)\}$  definem M pontos neste espaço N-dimensional com N eixos mutuamente perpendiculares  $\varphi_1 \dots \varphi_N$ .

Assim:

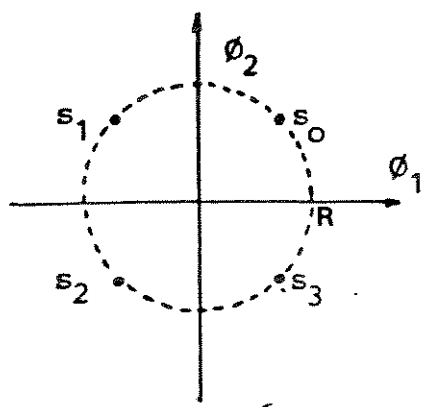
$$s_i = (s_{i1} \psi_1 + s_{i2} \psi_2 + \dots + s_{iN} \psi_N) \quad (1.5)$$

Cabe observar que o processo conhecido como ortogonalização de Gram-Schmidt [22] é um método apropriado para obtenção de um conjunto ortonormal para se representar  $s_i(t)$ .

#### 1.1.2.2 - A Representação por Pontos no Espaço de Sinais (Constelação de Sinais)

A partir da equação 1.5 é possível representar geometricamente os sinais  $s_i(t)$  como pontos neste espaço Euclidiano N-dimensional, chamado espaço de sinais e ainda essa representação é chamada constelação de sinais.

A Figura 1.4 mostra a representação de  $M = 4$  sinais em um espaço de duas dimensões ( $N = 2$ ):



$$N = 2$$

$$M = 4$$

$$\phi_2 = R \operatorname{sen} \theta$$

$$\phi_1 = R \operatorname{cos} \theta$$

Figura 1.4 - Representação geométrica de sinais (Constelação 4-PSK)

Neste trabalho as constelações estudadas se restringem ao caso bi-dimensional ( $N = 2$ ).

## 1.2 - ESQUEMAS DE MODULAÇÃO DIGITAL

### 1.2.1 - Os Esquemas Básicos

Podemos classificar a modulação digital em duas categorias básicas: modulação por pulsos, onde a largura, altura ou posição de um trem de pulsos é alterada conforme a informação a ser transmitida e modulação contínua, onde a amplitude, fase ou frequência de uma portadora senoidal ou co-senoidal é alterada conforme a informação a ser transmitida.

Os sistemas de comunicações básicos utilizam a técnica de modulação linear, em que as características determinadas por esta

técnica possibilitam a ocupação de uma faixa de frequência relativamente pequena para uma potência transmitida relativamente alta, porém, o desempenho do sistema fica afetado pelo ruído introduzido pelo canal durante a transmissão. Tentando amenizar essa degradação, as aplicações se voltaram à modulação angular, que embora apresente uma largura de faixa maior que a exigida pela modulação linear, sua utilização é compensada pela maior imunidade ao ruído. No entanto, essa melhora no desempenho ocorre em detrimento da expansão de faixa.

Com objetivo de contornar esse problema, foram desenvolvidos os sistemas pulsados: PAM, PPM, PWM, PCM, etc. E através desses sistemas, principalmente o PCM, originou-se a codificação, tanto de fonte como de canal.

Dentre esses sistemas pulsados, destaca-se exatamente o PCM, que quantiza e codifica pulsos em amplitude (PAM). O sinal PCM resultante pode ser facilmente regenerado e repetido, mas necessita uma larga faixa de frequência para a transmissão.

Para um casamento com o meio de transmissão, os sinais resultantes de sistemas pulsados passam a ser transmitidos através de modulação digital cujos esquemas binários ou M-ários estabelecem uma relação vantajosa de potência e largura de faixa em relação aos esquemas convencionais de modulação por pulsos.

Então, nosso interesse volta-se para a evolução dos sistemas de modulação contínua e para as características de esquema de comunicações digitais binários e M-ários, que tem como tipos básicos os seguintes esquemas:

- Modulação por chaveamento de amplitude (ASK - Amplitude Shift Keying)
- Modulação por chaveamento de fase (PSK - Phase Shift Keying)
- Modulação por chaveamento de frequência (FSK - Frequency Shift Keying)

Em ASK, uma portadora de frequência fixa é comutada entre diversos níveis. Para o caso binário, o bit "0" é geralmente transmitido como uma tensão zero, e o bit "1" como uma tensão não-zero de amplitude constante.

Na modulação PSK, a fase de uma portadora de amplitude fixa é comutada entre diversos valores. Se são usados pulsos binários (B-PSK), o bit "1" pode corresponder à portadora em fase com alguma referência e o bit "0" à portadora em  $180^\circ$  fora de fase. Em PSK de quatro níveis (4-PSK), a portadora pode ser comutada para  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  e  $270^\circ$  em relação à referência.

Na modulação FSK, a frequência da portadora de amplitude constante é comutada entre diversos valores.

A Figura 1.5 ilustra os três esquemas para o caso binário.

Além desses tipos básicos, são encontradas muitas variações e combinações. Uma delas é a modulação por chaveamento de fase diferencial (DPSK), onde a seqüência de bits a ser transmitida passa por uma codificação transformando-se numa seqüência que indica as transições de bit zero para bit um ou vice-versa.

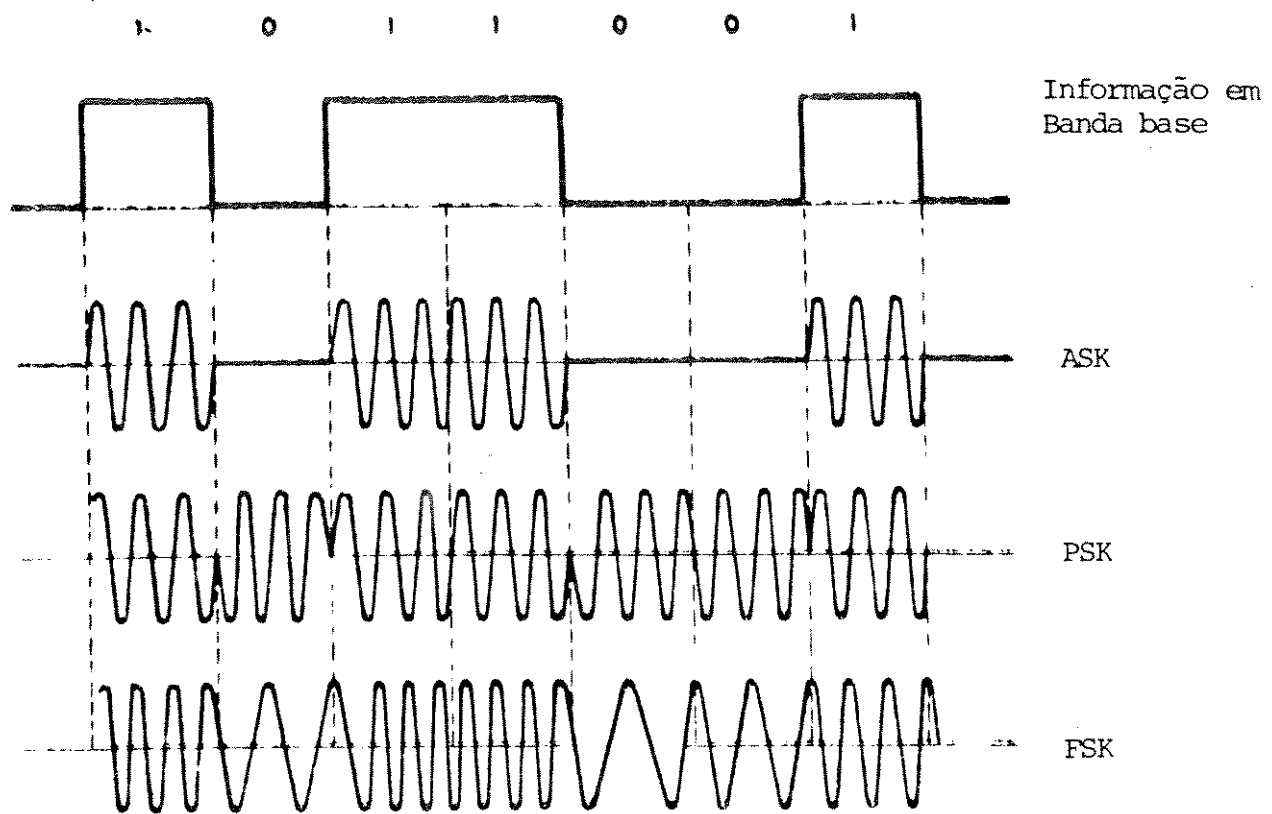


Figura 1.5 - Modulação de uma portadora por pulsos

### 1.2.2 - Esquemas $M^{\text{ário}}$ de Modulação

Quando  $M > 2$ , o esquema é chamado  $M^{\text{ário}}$ , em que os sinais modulados são gerados a partir da variação da amplitude, fase ou frequência de uma portadora em  $M$  níveis, ou ainda uma combinação desses parâmetros. Um exemplo de combinação é o M-APSK (M-ASK e M-PSK), sendo que a modulação por quadratura de amplitude (QAM-Quadrature Amplitude Modulation) é a mais utilizada.

A representação desses esquemas é feita através das

constelações no espaço de sinais bi-dimensionais. Para se enviar  $n$  bits/símbolo será necessária uma constelação de  $M=2^n$  pontos.

A Figura 1.6 mostra exemplos de constelações  $M$ árias.

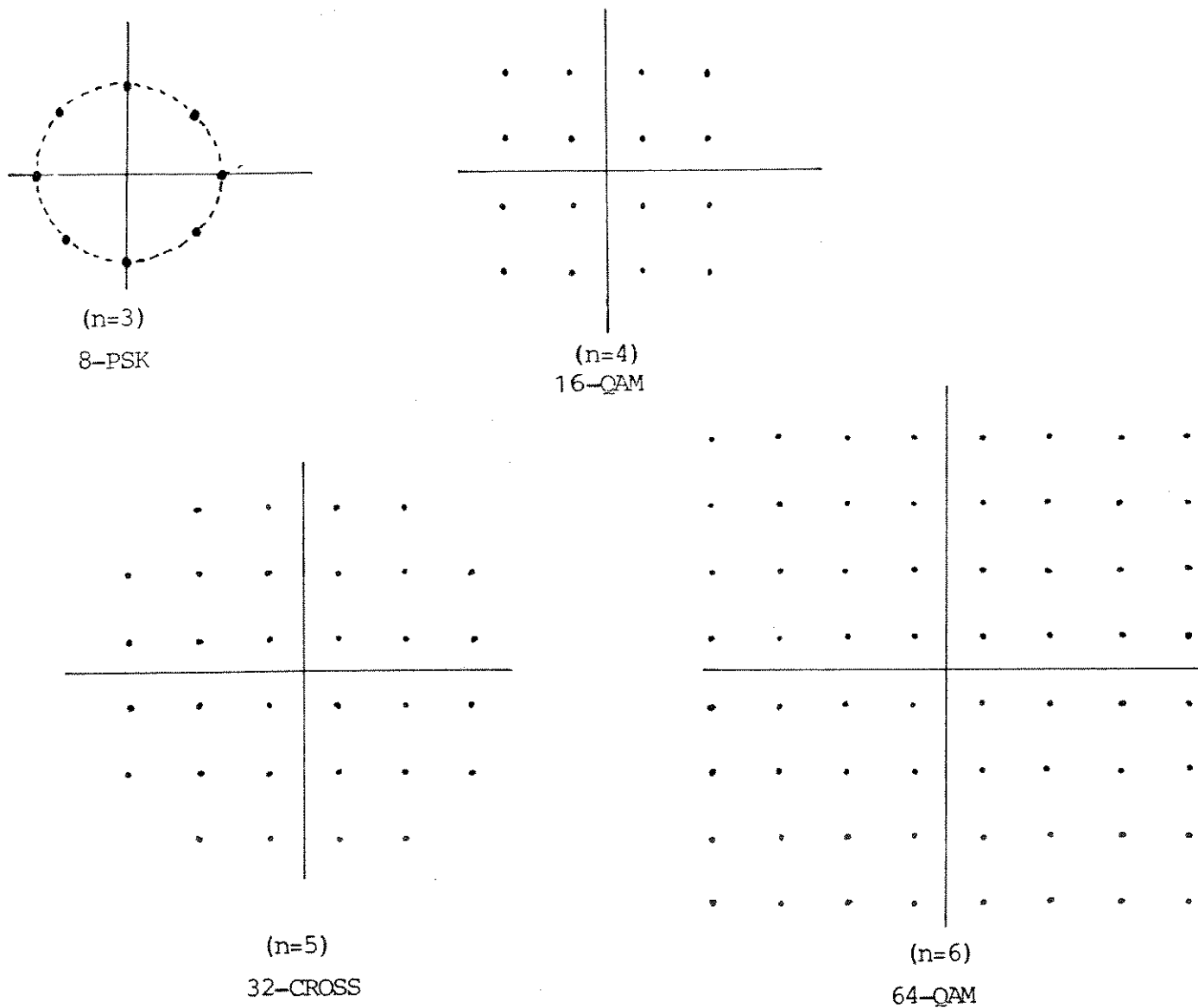


Figura 1.6 - Representação de Esquemas  $M$ ários de modulação por constelações no espaço de sinais



Gabe aqui discorrer a respeito de algumas características matemáticas das constelações, no sentido de se fazer uma comparação a cerca da eficiência destas.

Define-se a energia média ( $E_m$ ) dos  $M$  ( $M=2^n$ ) pontos de sinais de uma constelação da seguinte forma:

$$E_m = \frac{1}{M} \sum_{i=1}^M d_i^2 \quad (1.6)$$

onde  $d_i$  é a distância Euclidiana entre o  $i$ ésimo ponto da constelação e a origem.

Essa energia calculada por aproximação integral é dada por, [9]:

$$\hat{E}_m = (2/3) 2^n \quad (\text{para constelação QAM}) \quad (1.7)$$

$$\hat{E}_m = (31/48) 2^n \quad (\text{para constelação CROSS}) \quad (1.8)$$

Algumas comparações entre  $E_m$  e  $\hat{E}_m$  são mostradas na Tabela 1, devendo-se observar que as constelações apresentadas nesta tabela são as estudadas por Campopiano e Glazer [4].

Tabela 1

n.º de pontos	constelação	$E_m$ (dB)	$\hat{E}_m$ (dB)
4	4-QAM	3,0	4,3
8		7,4	
16	16-QAM	10,0	10,3
32	32-CROSS	13,0	13,2
64	64-QAM	16,2	16,3
128	128-CROSS	19,1	19,2
256	256-QAM	22,3	22,3

Nota-se então que a constelação CROSS é mais eficiente que a QAM por um fator 31/32 ou 0,14 dB (porque ela é mais próxima a um círculo). Assim, quanto mais próxima de um círculo, melhor deverá ser a eficiência da constelação em si, por ser o círculo a figura geométrica de menor energia média por área dada [9].

Um círculo de raio R tem área  $\pi R^2$  e energia média  $R^2/2$ . Colocando-se  $R^2 = 4 \times 2^n$ , nós percebemos que a energia média para uma constelação circular de  $2^n$  pontos deverá ser:

$$\hat{E}_m = (2/\pi) 2^n \quad (\text{constelação circular}) \quad (1.8)$$

a qual deverá ser  $\pi/3$  ou 0,2 dB melhor que a constelação QAM ou 0,06 dB melhor que a constelação CROSS, [9].



### 1.2.3 - Avaliação do Desempenho de Sistemas de Modulação Digital

Vimos na Seção 1.1.2 que um conjunto de  $M$  sinais especificados por  $s(t) = \{s_i(t)\}$  podem ser representados por um conjunto de  $M$  pontos  $\{s_i\}$  num espaço  $N$ -dimensional e que essa representação é chamada constelação de sinais. Levando em consideração que este conjunto é transmitido sobre um canal perturbado somente por um ruído aditivo, Gaussiano, branco e de média zero, a avaliação do desempenho do sistema de modulação digital pode ser feita a partir da probabilidade de erro ( $P[e]$ ) na decisão. É necessário dizer que o ruído é aleatório,  $N$ -dimensional com função densidade de probabilidade dada por:

$$P_n(\alpha) = \frac{1}{(\pi N_0)^{1/2}} e^{-|\alpha|^2 / N_0} \quad (1.9)$$

onde  $N_0$  é a densidade espectral do ruído.

Faremos essa avaliação baseados na hipótese que os  $M$  sinais transmitidos são equiprováveis (o que sempre será verificado neste trabalho), e a partir do exemplo simples para  $M=2$ , ou seja, sinais binários.

#### 1.2.3.1 - A Probabilidade de Erro ( $P[e]$ )

O caso particular de dois sinais, cada um com probabilidade igual a  $1/2$ , pode ser representado conforme a Figura 1.7a.

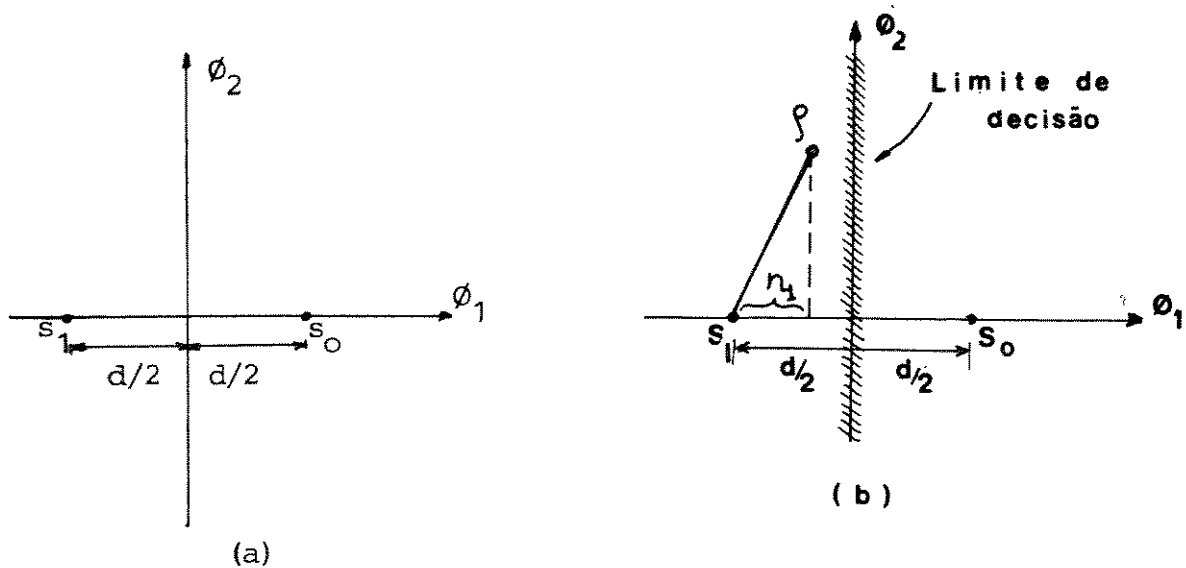


Figura 1.7

Assim, um erro será cometido quando  $s_1$  for transmitido, no receptor, decidir-se por  $s_0$  ou vice-versa. Então devemos determinar as regiões de decisão para cada ponto no espaço de sinais, de forma a minimizar a probabilidade de decisão errada.

As regiões de decisão ótimas são determinadas pela expressão, [22]:

$$\min_i |\rho - s_i|^2 - N_0 \ln P[m_i] \quad (1.10)$$

onde:

$P[m_i]$  é a probabilidade de ser transmitido o sinal  $s_i$ , e  $\rho$  é representado na Figura 1.7b

Para iguais probabilidades a priori, a regra de decisão é, [22]:



decide-se por  $m_j$ , tal que:

$$|\rho - s_j|^2 = \min_i |\rho - s_i|^2 \quad (1.11)$$

Assim, fica claro a partir da Figura 1.7b que o lugar de todos os pontos  $\rho$  igualmente distante de  $s_0$  e  $s_1$  é o eixo  $\varphi_2$ . Portanto, um erro ocorre quando  $s_1$  é transmitido se e somente se uma componente de ruído  $n_1$  exceder a  $d/2$ , onde  $d$  é a distância entre os sinais, [22].

Assim:

$$P[e/m_1] = P\{n_1 > d/2\}, \quad (1.12)$$

onde  $P[e/m_1]$  é a probabilidade de erro dado que  $s_1$  foi transmitido

$$d^2 = \int_{-\infty}^{\infty} [s_0(t) - s_1(t)]^2 dt \quad (1.13)$$

ou seja,  $d$  é a distância Euclidiana entre os pontos  $s_0$  e  $s_1$ .

Mas  $n_1$  é Gaussiano de média zero com variância  $N_0/2$ , então:

$$P[e/m_1] = \int_{d/2}^{\infty} \frac{1}{(\pi N_0)^{1/2}} e^{-\alpha^2/N_0} d\alpha \quad (1.14)$$

fazendo-se  $\gamma = \alpha(2 N_0)^{1/2}$ , teremos:

$$P[e/m_1] = \int_{d/2}^{\infty} \frac{1}{(N_0/2)^{1/2}} \frac{1}{(2\pi)^{1/2}} e^{-\gamma^2/2} d\gamma = Q(d/\sqrt{2 N_0}) \quad (1.15)$$



Como, por simetria, a probabilidade de erro condicional é a mesma para cada sinal, temos que:

$$P[e] = \sum_{l=0}^{1} P[m_l] P[e/m_l] = P[e/m_1] = Q(d/\sqrt{2 N_0}) \quad (1.16)$$

A equação 1.16 determina a probabilidade de erro para qualquer par de sinais equiprováveis separados por uma distância  $d$ , independentemente de suas localizações no espaço de sinais, [22].

### 1.2.3.2 - O Critério de Avaliação

Naturalmente um sistema de modulação será tanto melhor quanto menor for a probabilidade de erro associada a ele, assim um critério útil é o da mínima probabilidade de erro.

Da equação 1.16 temos que para minimizar  $P[e]$  devemos maximizar  $d$ , pois a função  $Q(\cdot)$  é monotonicamente decrescente. Assim, quanto maior a distância  $d$  entre os dois sinais  $s_0$  e  $s_1$  no espaço de sinais, menor será a probabilidade de erro e conseqüentemente melhor o desempenho do sistema de modulação em questão.

Para o caso onde  $M > 2$ , o critério é praticamente o mesmo: devemos maximizar a menor distância Euclidiana entre dois pontos quaisquer da constelação, ou seja, maximizar a distância mínima Euclidiana entre os pontos da constelação. Esse critério será o utilizado neste trabalho para a comparação entre diferentes constelações.



## CAPÍTULO 2

### OS CÓDIGOS CONVOLUCIONAIS BINÁRIOS



## 2.1 - INTRODUÇÃO

O objetivo primeiro deste trabalho é a avaliação do desempenho dos códigos do tipo TCM (descritos no Capítulo 3), para o qual se faz necessária a apresentação de uma revisão a respeito dos códigos convolucionais binários, visando um perfeito entendimento dos conceitos desenvolvidos posteriormente.

Para isso descrevemos o processo de codificação para os códigos convolucionais, até chegarmos à definição destes. Mostramos diferentes formas da representação de um codificador convolucional: a representação matricial, o diagrama de estados, a função de transferência, o diagrama de treliça, e posteriormente os vetores coeficientes. Além disso, definimos a distância mínima de um código convolucional, bem como apresentamos o processo de decodificação por máxima verossimilhança através do algoritmo de Viterbi.

Finalmente, discorremos a respeito dos códigos convolucionais binários de memória unitária, cujo estudo é um dos objetivos deste trabalho.

## 2.2 - A CODIFICAÇÃO DOS CÓDIGOS CONVOLUCIONAIS

"Um codificador convolucional  $G_{kn}$  com ' $k$ ' entradas e ' $n$ ' saídas, sobre um corpo finito  $GF(q)$  é uma máquina seqüencial de estados finitos, linear, inversível e realizável, a qual estava no estado zero num passado longínquo".

Neste trabalho serão abordados os codificadores convolucionais do tipo binário em que suas operações são realizadas no



onde:

$u_i^t$  representa o  $i$ .<sup>ésimo</sup> elemento do bloco das 'k' entradas, com  $1 \leq i \leq k$ , no instante de tempo 't'.

Seja  $\underline{v}$  a seqüência de saída (palavra código) que será racional e causal, já que o codificador é realizável por definição, e a seqüência de informação é racional e causal por hipótese:

$$\begin{aligned} \underline{v} &= \{ \underline{v}^0, \underline{v}^1, \underline{v}^2, \dots \} = \\ &= \{ v_1^0 \ v_2^0 \ \dots \ v_n^0, \ v_1^1 \ v_2^1 \ \dots \ v_n^1, \ v_1^2 \ v_2^2 \ \dots \ v_n^2, \dots \} \end{aligned} \quad (2.2)$$

onde :

$v_j^t$  representa o  $j$ .<sup>ésimo</sup> elemento do bloco das 'n' saídas, com  $1 \leq j \leq n$ , no instante de tempo 't'.

A seqüência de informação  $\underline{u}$  dada por (2.1) entra no codificador como um bloco de 'k' bits a cada unidade de tempo. Como o codificador é um sistema linear, as 'n' seqüências de saída dadas por:

$$\begin{aligned} \underline{v}_1 &= \{ v_1^0, v_1^1, v_1^2, \dots \} \\ \underline{v}_2 &= \{ v_2^0, v_2^1, v_2^2, \dots \} \\ &\vdots \\ \underline{v}_n &= \{ v_n^0, v_n^1, v_n^2, \dots \} \end{aligned} \quad (2.3)$$

podem ser obtidas pela convolução das 'k' seqüências de entrada dadas por:



$$\begin{aligned}
 \underline{u}_1 &= \{u_1^0, u_1^1, u_1^2, \dots\} \\
 \underline{u}_2 &= \{u_2^0, u_2^1, u_2^2, \dots\} \\
 &\vdots \\
 \underline{u}_k &= \{u_k^0, u_k^1, u_k^2, \dots\}
 \end{aligned}
 \tag{2.4}$$

com as respostas impulsivas do codificador.

Essas respostas impulsivas podem ser obtidas uma vez que se alimente um e somente um registro com o dígito '1' e os demais com zero (o que é equivalente à aplicação de um impulso na entrada de um sistema linear para a obtenção da função de transferência do sistema) e observando-se as seqüências de saída. Assim:

$$\begin{aligned}
 \underline{g}_{i1} &= \{g_{i1}^0, g_{i1}^1, g_{i1}^2, \dots, g_{i1}^m\} \\
 \underline{g}_{i2} &= \{g_{i2}^0, g_{i2}^1, g_{i2}^2, \dots, g_{i2}^m\} \\
 &\vdots \\
 \underline{g}_{in} &= \{g_{in}^0, g_{in}^1, g_{in}^2, \dots, g_{in}^m\}
 \end{aligned}
 \tag{2.5}$$

onde:

$\underline{g}_{ij}$  representa a "função de transferência" relativa à saída 'j' e entrada 'i', com  $1 \leq i \leq k$  e  $1 \leq j \leq n$  e,

m é a ordem de memória do codificador dada por:

$$m = \max_i (m_i), \quad \text{com } 1 \leq i \leq k
 \tag{2.6}$$

em que  $m_i$  é o número de registradores de deslocamento na  $i$ ésima linha de entrada do codificador.

Assim, as equações de codificação são dadas por:

$$v_1 = \sum_{i=1}^k u_i * q_{1i}$$

$$v_2 = \sum_{i=1}^k u_i * q_{2i}$$

⋮

$$v_n = \sum_{i=1}^k u_i * q_{ni}$$

(2.7)

onde \* denota a operação de convolução discreta e  $\Sigma$  o somatório módulo-2.

Daí o nome dado a essa classe de codificadores - codificadores convolucionais, [21].

As seqüências  $q_{ij}$  são chamadas seqüências geradoras, e depois de multiplexadas as 'n' seqüências de saída  $v_1, v_2, \dots, v_n$ , obtem-se a palavra código  $v$  dada por (2.2).

Para maior clareza, observemos um exemplo:

Seja o codificador da Figura 2.1, para este codificador temos:

$$k=1, \quad n=2, \quad m=2$$

Então se fizermos  $u = \{1, 0, 0, \dots\}$  e observarmos as saídas  $v_1$  e  $v_2$  teremos:

$$q_{11} = \{1 \ 0 \ 1\}$$

$$q_{12} = \{1 \ 1 \ 1\}$$



que são as seqüências geradoras do codificador. Assim, para a seqüência de informação  $\underline{u} = \{1\ 0\ 1\ 1\}$ :

$$\underline{v}_1 = \underline{u} * \underline{g}_{11} = \{1\ 0\ 1\ 1\} * \{1\ 0\ 1\} = \{1\ 0\ 0\ 1\ 1\ 1\}$$

$$\underline{v}_2 = \underline{u} * \underline{g}_{12} = \{1\ 0\ 1\ 1\} * \{1\ 1\ 1\} = \{1\ 1\ 0\ 0\ 0\ 1\}$$

multiplexando-se as saídas temos:

$$\underline{v} = \{11, 01, 00, 10, 10, 11\}$$

### 2.2.2 - O Código Convolutional

**Definição 2.2.2.1** : "O código convolutional  $C$ , gerado por um codificador convolutional  $G_{kn}$ , é o conjunto de todas as seqüências de saídas  $\underline{v}$ , produzidas pelo conjunto de todas as seqüências de entrada  $\underline{u}$ ".

**Definição 2.2.2.2** : A razão  $k/n$  chama-se taxa do código e denota-se:  
 $R = k/n$ .

**Definição 2.2.2.3** : "Dois codificadores são ditos equivalentes se apresentam o mesmo conjunto de seqüências de saída sobre o mesmo conjunto de todas as seqüências de entrada".

## 2.3 - A REPRESENTAÇÃO MATRICIAL DE UM CODIFICADOR CONVOLUCIONAL

Em qualquer sistema linear, no domínio do tempo, as operações envolvendo convolução podem ser substituídas, no domínio da frequência, por operações envolvendo multiplicação polinomial. Como um codificador convolucional é um sistema linear, cada seqüência nas equações de codificação pode ser substituída pelo correspondente polinômio, e a operação de convolução substituída por multiplicação polinomial. Num polinômio que representa uma seqüência binária, a seqüência em si é representada pelos coeficientes deste polinômio. Assim:

$$\underline{U}(D) = [\underline{U}_1(D) \underline{U}_2(D) \dots \underline{U}_k(D)] \quad (2.8)$$

onde :

$$\underline{U}_i(D) = u_i^0 + u_i^1 D + u_i^2 D^2 + \dots, \quad \text{para } i = 1, 2, \dots, k \quad (2.9)$$

$$\underline{V}(D) = [\underline{V}_1(D) \underline{V}_2(D) \dots \underline{V}_n(D)] \quad (2.10)$$

onde:

$$\underline{V}_j(D) = v_j^0 + v_j^1 D + v_j^2 D^2 + \dots, \quad \text{para } j = 1, 2, \dots, n \quad (2.11)$$

E ainda as seqüências geradoras do codificador  $g_{ij}$  com  $1 \leq i \leq k$  e  $1 \leq j \leq n$  são substituídas por polinômios  $\underline{G}_{ij}(D)$  que são chamados polinômios geradores. Cada polinômio  $\underline{G}_{ij}(D)$  é o polinômio gerador para a saída 'j' relativo à entrada 'i'.

Em um codificador há 'n' polinômios geradores para cada uma das 'k' entradas, onde cada conjunto de 'n' polinômios geradores representa as conexões de um dos registradores de deslocamento para as 'n' saídas. Assim, os polinômios geradores podem ser determinados



diretamente a partir da representação esquemática do codificador, desde que cada estágio de registradores de deslocamento represente uma unidade de tempo de atraso, a seqüência de conexões (um "1" representa conexão e um "0" não conexão) de um registrador de deslocamento para uma saída é a seqüência de coeficientes no correspondente polinômio gerador.

Por exemplo: no codificador da Figura 2.1 a seqüência de conexões para a primeira saída é  $\underline{g}_{11} = \{1 \ 0 \ 1\}$  e o correspondente polinômio gerador é  $\underline{G}_{11}(D) = 1 + D^2 = 1.D^0 + 0.D^1 + 1.D^2$ .

Uma vez que o codificador é um sistema linear, e  $\underline{U}_i(D)$  representa a  $i$ .<sup>ésima</sup> seqüência de entrada e  $\underline{V}_j(D)$  a  $j$ .<sup>ésima</sup> seqüência de saída, o polinômio gerador  $\underline{G}_{ij}(D)$  pode ser interpretado como a função de transferência relativa à entrada 'i' e saída 'j'. Como em qualquer sistema linear de 'k' entradas e 'n' saídas, há um total de k.n funções de transferência. Este pode ser representado por uma matriz k x n chamada matriz função de transferência e dada por:

$$G(D) = \begin{bmatrix} \underline{G}_{11}(D) & \underline{G}_{12}(D) & \dots & \underline{G}_{1n}(D) \\ \underline{G}_{21}(D) & \underline{G}_{22}(D) & \dots & \underline{G}_{2n}(D) \\ \vdots & \vdots & & \vdots \\ \underline{G}_{k1}(D) & \underline{G}_{k2}(D) & \dots & \underline{G}_{kn}(D) \end{bmatrix} \quad (2.12)$$

Figura 2.2 - Representação matricial de um codificador convolucional

onde:  $\underline{G}_{ij}(D) = g_{ij}^0 + g_{ij}^1 D + g_{ij}^2 D^2 + \dots + g_{ij}^m D^m$

para  $i = 1, 2, \dots, k$   
 $j = 1, 2, \dots, n$

Usando a matriz função de transferência (Matriz Geradora) as equações de codificação podem ser expressas como:

$$\underline{V}(D) = \underline{U}(D).G(D) \tag{2.13}$$

Para o codificador da Figura 2.1 temos polinômios geradores para o codificador  $\{G_{12}, 2\}$  e representação matricial para o mesmo codificador, respectivamente.

$$G_{11}(D) = 1 + D^2$$

$$G(D) = [1 + D^2 \quad 1 + D + D^2]$$

$$G_{12}(D) = 1 + D + D^2$$

**Definição 2.3.1 :** Um código convolucional é dito sistemático quando 'k' dos 'n' dígitos de saída são idênticos aos 'k' dígitos de entrada.

Usando a representação matricial, para um código sistemático, temos:

$$G(D) = \begin{bmatrix} \underline{G}_{11}(D) & \underline{G}_{12}(D) & \dots & \underline{G}_{1 \ n-k}(D) & 1 & 0 & \dots & 0 \\ \underline{G}_{21}(D) & \underline{G}_{22}(D) & \dots & \underline{G}_{2 \ n-k}(D) & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \underline{G}_{k1}(D) & \underline{G}_{k2}(D) & \dots & \underline{G}_{k \ n-k}(D) & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$= [G'(D) : I_k] \tag{2.14}$$

onde :  $I_k$  é a matriz identidade de ordem k

Vejamos o exemplo da Figura 2.3

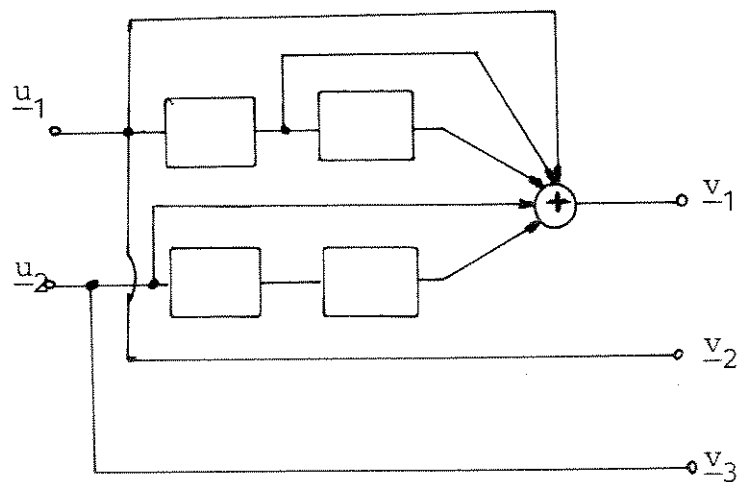


Figura 2.3 - Codificador convolucional sistemático  $[G_{23} : 2,2]$

Para determinarmos os polinômios geradores  $\underline{G}_{ij}(D)$  devemos considerar que, como  $m=2$ , o grau deste polinômio também será 2. Assim:

$$\underline{G}_{ij}(D) = \underline{a}_{ij}^c D^0 + \underline{a}_{ij}^1 D^1 + \underline{a}_{ij}^2 D^2$$



lembrando que uma conexão é representada por um elemento '1' e uma não-conexão por um elemento '0'. Teremos, portanto,

$$\underline{G}_{11}(D) = 1 D^0 + 1 D^1 + 1 D^2 = 1 + D + D^2$$

$$\underline{G}_{12}(D) = 1 D^0 + 0 D^1 + 0 D^2 + 0 D^2 = 1$$

$$\underline{G}_{13}(D) = 0 D^0 + 0 D^1 + 0 D^2 = 0$$

$$\underline{G}_{21}(D) = 1 D^0 + 0 D^1 + 1 D^2 = 1 + D^2$$

$$\underline{G}_{22}(D) = 0 D^0 + 0 D^1 + 0 D^2 = 0$$

$$\underline{G}_{23}(D) = 1 D^0 + 0 D^1 + 0 D^2 = 1$$

$$\text{logo: } G(D) = \begin{bmatrix} \underline{G}_{11}(D) & \underline{G}_{12}(D) & \underline{G}_{13}(D) \\ \underline{G}_{21}(D) & \underline{G}_{22}(D) & \underline{G}_{23}(D) \end{bmatrix} = \begin{bmatrix} 1 + D + D^2 & 1 & 0 \\ 1 + D^2 & 0 & 1 \end{bmatrix}$$

A relevância dos códigos sistemáticos advém do fato de que o codificador correspondente é mais simples em comparação aos codificadores correspondentes aos códigos não-sistemáticos (que não obedecem à definição 2.3.1) de mesmo número de entradas 'k' e mesmo número de saída 'n'.

## 2.4 - O DIAGRAMA DE ESTADOS E A FUNÇÃO DE TRANSFERÊNCIA

### 2.4.1 - O Diagrama de Estados

Como um codificador convolucional é uma máquina seqüencial, sua operação pode ser descrita por um diagrama de estados, onde o estado do codificador é definido como o conteúdo dos registradores de deslocamento.

Definindo:

$$M = \sum_{l=1}^k m_l \quad (2.15)$$

como a memória total do codificador, ou ainda, comprimento total de memória (GTM), para um codificador convolucional binário, o número de estados possíveis será  $2^M$ .

Cada bloco de 'k' entradas causa transição para um novo estado. Associando-se a cada transição um ramo e a cada estado um nó, há  $2^k$  ramos saindo de cada nó.

A Figura 2.4 mostra o diagrama de estados para o codificador da Figura 2.1:

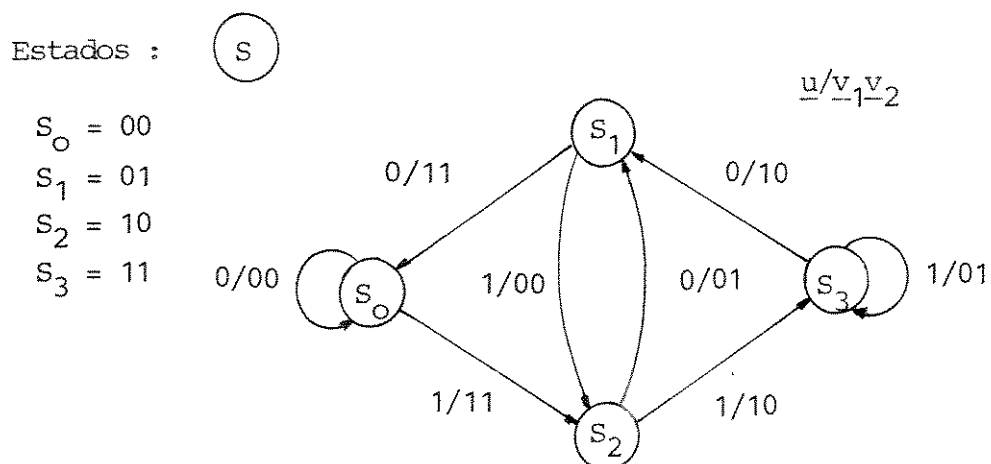


Figura 2.4 - Diagrama de estados

Assumindo que o codificador está inicialmente no estado zero ( $S_0$ ), a palavra código correspondente a qualquer seqüência de informação dada, pode ser obtida seguindo-se o caminho completo no diagrama de estados determinados por esta seqüência de informação, e anotando-se as correspondentes saídas sobre os ramos indicados. O codificador retorna ao estado  $S_0$  por uma seqüência de  $M$  elementos zeros anexados à seqüência de informação.

Para o exemplo da Figura 2.5, se  $\underline{u} = \{1 \ 0 \ 1 \ 1 \ 0 \ 0\}$ , o caminho completo é dado pelos ramos que ligam os seguintes estados pela ordem:  $S_0 S_2 S_1 S_2 S_3 S_2 S_0$ , e a palavra código correspondente é:

$$\underline{v} = \{11, 01, 00, 10, 10, 11\}$$

#### 2.4.2 - O Diagrama de Estados Modificado

O diagrama de estados pode ser modificado para produzir uma descrição completa dos pesos de Hamming (número de elementos na seqüência iguais a '1') de todas as palavras códigos diferentes da palavra código toda zero. Isso representa uma função distribuição de peso para o código. Este novo diagrama é caracterizado da seguinte maneira: como o estado zero ( $S_0$ ) possui uma auto-malha de peso zero, procede-se a separação do mesmo em um estado inicial e um estado final, abrindo-se esta auto-malha. Associa-se um ganho  $X^i$  a cada ramo do diagrama, cujo expoente 'i' é igual ao peso de Hamming dos 'n' bits codificados sobre o ramo. A multiplicação dos ganhos dos ramos para cada caminho que parte e retorna ao estado  $S_0$  é chamado ganho de caminho, o expoente deste ganho é o peso da palavra código associada

ao caminho, ou ainda, a distância desse caminho em relação ao caminho todo zero.

A Figura 2.5 mostra o diagrama de estados modificado correspondente à Figura 2.4:

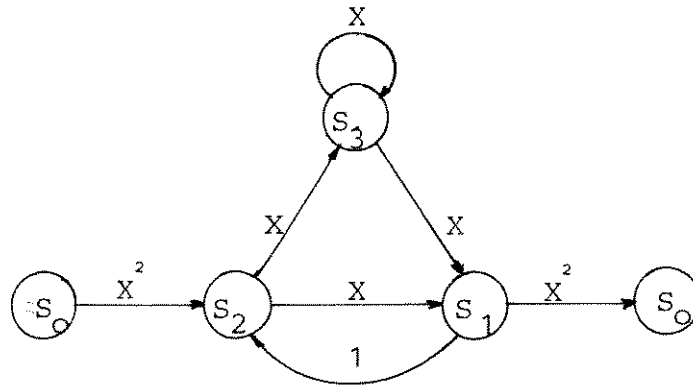


Figura 2.5 - Diagrama de estados modificado

**Definição 2.4.1** - Um código convolucional é dito catastrófico quando um número finito de erros introduzidos pelo canal causa um número infinito de erros de decodificação.

Por isso deve-se evitar sua escolha para o emprego de correção de erros aleatórios.

Em termos do diagrama de estados, um código é catastrófico se e somente se o diagrama contém um "loop" de peso zero além do "loop" ao redor do estado zero ( $S_0$ ).

### 2.4.3 - A Função de Transferência

A função de transferência ou função distribuição de pesos de um código é o polinômio que especifica o número de palavras código com peso de Hamming  $w_H$ , e pode ser obtido considerando-se o diagrama de estado modificado como um sistema linear dinâmico discreto.

Para o exemplo da Figura 2.6 as equações de estado são:

$$\begin{aligned} E(i+1) &= A(i).E(i) + B(i) \\ T(i) &= H(i).E(i) \end{aligned} \tag{2.16}$$

onde :

$E(i)$  é a matriz de estados  $3 \times 1$  específica dos valores das transições de estado inicial para os estados intermediários no instante  $t=i$ ;

$A(i)$  é uma matriz de transição  $3 \times 3$  onde seus elementos são os valores das transições entre os estados intermediários;

$H(i)$  é uma matriz de saída especificando os valores das transições dos estados intermediários para o estado final;

$B(i)$  é a matriz inicial  $3 \times 1$  especificando os valores das transições do estado inicial para os estados intermediários.

No exemplo temos:

$$E(i) = [\xi_{i1}, \xi_{i2}, \xi_{i3}]$$

$$H(i) = [0 \quad X^2 \quad 0]$$

$$B^t(i) = [X^2 \quad 0 \quad 0]$$

$$A(i) = \begin{bmatrix} 0 & 1 & 0 \\ X & 0 & X \\ X & 0 & X \end{bmatrix} \quad (2.17)$$

Substituindo-se (2.17) em (2.16) e resolvendo-se o sistema de equações, temos:

$$T(X) = X^5 / (1-2X) = X^5 + 2X^6 + 4X^7 + \dots \quad (2.18)$$

A interpretação da equação 2.18 é que existe um caminho com peso 5, dois (2) caminhos com peso 6, quatro (4) caminhos com peso 7 e assim por diante, todos em relação ao caminho todo nulo.

Se estivermos interessados no número de "1" contidos na seqüência de informação, podemos utilizar o artifício de introduzir a variável Z aos valores da função de transição que tenham sido causados pelo valor "1" na seqüência de informação. A Figura 2.6 mostra o diagrama de estados modificados da Figura 2.5 com a introdução da variável Z.

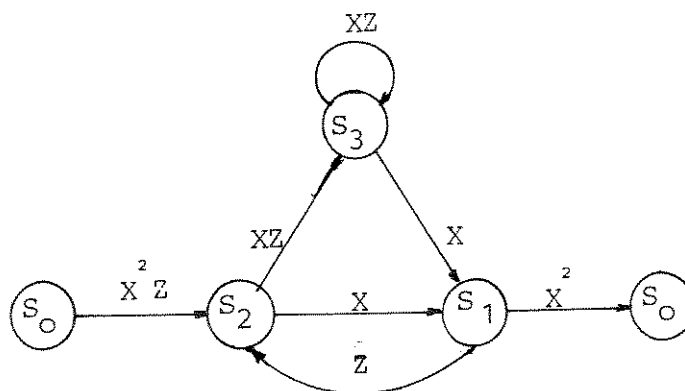


Figura 2.6 - Diagrama de estados modificado contendo a variável Z

Então as matrizes da equação 2.17 passam a ter a seguinte composição:

$$E(i) = [\xi_{i1}, \xi_{i2}, \xi_{i3}]$$

$$H(i) = [0 \quad X^2 \quad 0]$$

$$B^t(i) = [X^2Z \quad 0 \quad 0]$$

$$\begin{bmatrix} 0 & Z & 0 \\ X & 0 & X \\ XZ & 0 & XZ \end{bmatrix} \quad (2.19)$$

Substituindo-se (2.19) em (2.16) e resolvendo-se o sistema temos:

$$T(X,Z) = X^5Z/(1-2XZ) = X^5Z + 2X^6Z^2 + 4X^7Z^3 + \dots \quad (2.20)$$

A interpretação de (2.20) é que existe um caminho com peso 5 e com um bit "1", existem dois caminhos com peso 6 e com dois bits "1", existem 4 caminhos com peso 7 e 3 bits "1", e assim por diante, sempre em relação ao caminho todo nulo da seqüência de informação.

## 2.5 - O DIAGRAMA DE TRELIÇA

Uma outra representação para um codificador convolucional é uma estrutura chamada diagrama de treliça que é uma expansão do diagrama de estados no tempo.

A Figura 2.7 mostra uma estrutura chamada célula de treliça para o código da Figura 2.1:

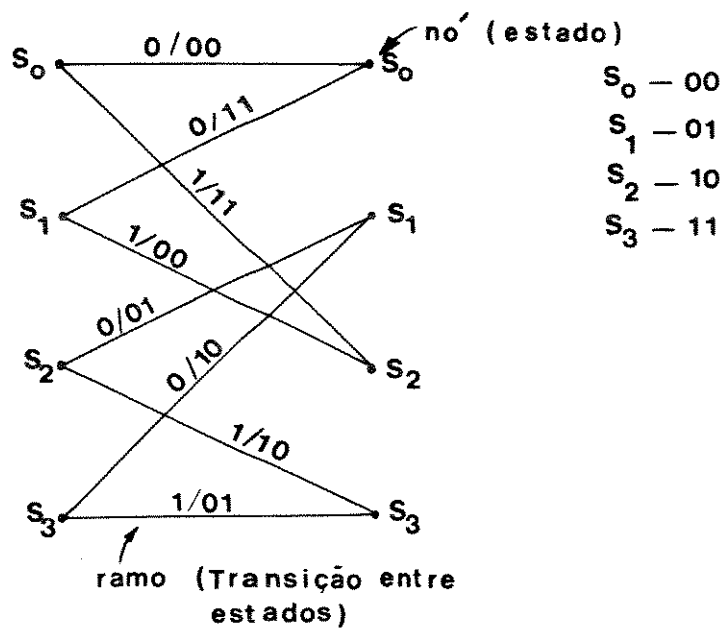


Figura 2.7 - Célula de treliça

A indicação sobre cada ramo representada por  $\underline{u}^t/\underline{v}^t$  significa que a uma entrada  $\underline{u}^t$  o codificador muda de estado produzindo  $\underline{v}^t$  na saída.

A Figura 2.8 mostra o diagrama de treliça para o mesmo código, que é uma réplica da célula de treliça a cada unidade de tempo. Estados

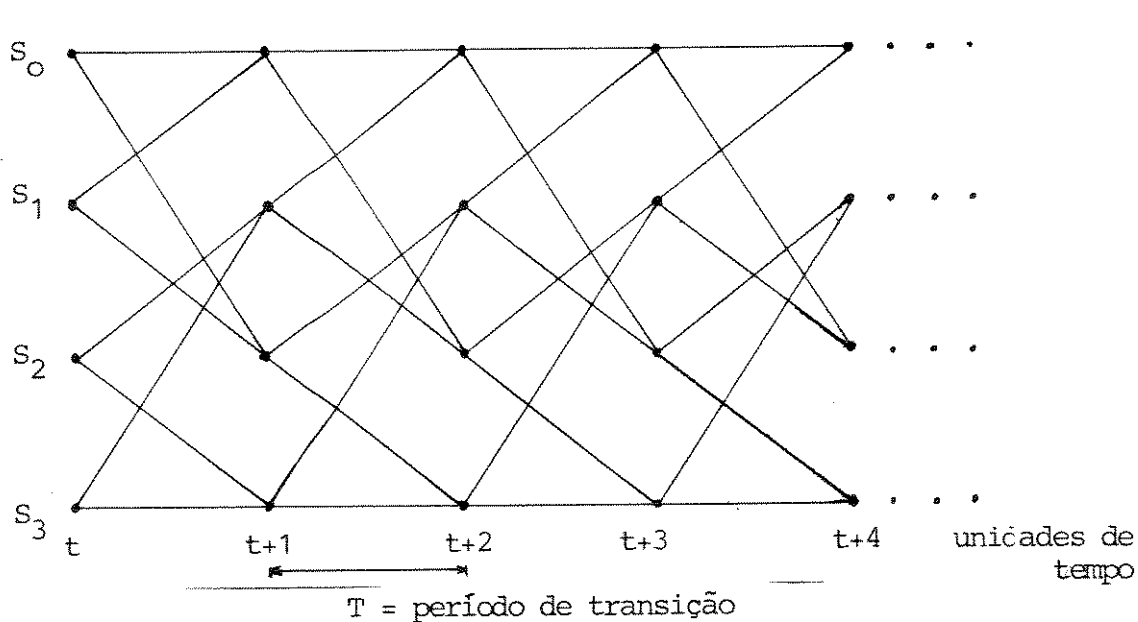


Figura 2.8 - Diagrama de treliça



A representação de um codificador pelo seu diagrama de treliça é conveniente para o processo de decodificação através do Algoritmo de Viterbi que será apresentado na Seção 2.7.

## 2.6 - A DISTÂNCIA DE UM CÓDIGO CONVOLUCIONAL

### 2.6.1 - A Distância Mínima de Hamming

O desempenho de um código convolucional depende do procedimento usado na decodificação e das propriedades de distância do código. Para a decodificação por máxima verossimilhança, que será abordada neste trabalho, a distância mais propícia é a mínima distância livre ( $d_{free}$ , do inglês: free distance) [6], aqui chamada apenas distância mínima ( $d_{min}$ ).

Para códigos convolucionais binários, define-se, [11]:

$$d_{min} = \min_{\underline{v} \neq \underline{v}'} d_H(\underline{v}, \underline{v}') \quad (2.21)$$

onde:  $d_H(\underline{v}, \underline{v}')$  é a distância de Hamming (número de bits em que as palavras diferem) entre as duas palavras  $\underline{v}$  e  $\underline{v}'$  em C.

Assim,  $d_{min}$  é a menor distância entre quaisquer duas palavras no código C. Como um código convolucional é um código linear, temos, [11]:

$$d_{min} = \min_{\underline{v}} \{w_H(\underline{v}) / \underline{v} \neq 0\} \quad (2.22)$$

onde:  $w_H(\underline{v})$  é o peso de Hamming de palavras não-nulas  $\underline{v}$  em C.

Então  $d_{\min}$  é o mínimo peso da palavra código de qualquer comprimento produzida por uma seqüência de informação não-nula. Em termos do diagrama de estados, é o mínimo peso de todos os caminhos que partem e retornam ao estado zero ( $S_0$ ), e ainda o menor expoente de  $X$  na função de transferência  $T(X)$ .

Para o código cujo diagrama de estados modificado é o da Figura 2.5, temos que o caminho de menor peso é dado pela seqüência de estados na ordem :  $S_0, S_2, S_1, S_0$ , com peso  $X^2.X.X^2 = X^5$ , e portanto:

$$\underline{d_{\min} = 5}$$

### 2.6.2 - A Distância Mínima Euclidiana

Após a modulação, obtem-se seqüências de sinais, e então pode-se definir a distância mínima Euclidiana entre seqüências de sinais modulados,

$$d_{\min} = \min_{\underline{v} \neq \underline{v}'} d_E [a(\underline{v}), a(\underline{v}')] \quad (2.23a)$$

onde:

$$d_E [a(\underline{v}), a(\underline{v}')] = \left\{ \sum_{\tau} d_E^2 [a(\underline{v}), a(\underline{v}')] \right\}^{1/2} \quad (2.23b)$$

é a distância Euclidiana entre as seqüências de sinais modulados (símbolos de canal). Analogamente à distância mínima de Hamming, a distância mínima Euclidiana é a menor distância entre quaisquer duas seqüências de sinais modulados.

Neste trabalho avaliaremos o desempenho de classes simples de codificadores convolucionais associados à modulação, por isso, a partir daqui, sempre que houver referência à distância mínima, esta será a distância mínima Euclidiana, a menos que se especifique o contrário.

## 2.7 - A DECODIFICAÇÃO POR MÁXIMA VEROSSIMILHANÇA E O ALGORITMO DE VITERBI

### 2.7.1 - A Decodificação por Máxima Verossimilhança

O critério de decodificação por máxima verossimilhança de um código convolucional é aquele que proporciona a menor probabilidade de erro  $P(e)$ .

Seja uma seqüência de informação  $\underline{u}$  dada por (2.1), de comprimento finito  $kL$ , ou seja,  $L$  blocos de comprimento  $k$ , codificada na palavra código  $\underline{v}$  dada por (2.2) que será comprimento  $N = n(L + m)$ .

Seja a seqüência  $Q^{aria}$   $\underline{r} = (r_0, r_1, \dots, r_{N-1})$  recebida de um DMC (canal discreto sem memória) de entrada binária e saída  $Q^{aria}$ .

Um decodificador de máxima verossimilhança para um DMC escolhe  $\hat{\underline{v}}$  como a palavra código  $\underline{v}$  que maximiza a função  $\log P(r/v)$  chamada função de verossimilhança, pois tem-se que, [11]:

$$P(e/\underline{r}) \stackrel{\Delta}{=} P(\hat{\underline{v}} \neq \underline{v}/\underline{r}) \quad (2.24)$$

é a probabilidade de erro dado  $\underline{r}$  e portanto,

$$P(e) = \sum_{\underline{r}} P(e/\underline{r}) \cdot P(\underline{r}) \quad (2.25)$$

Então para minimizar  $P(e)$  devemos minimizar  $P(e/\underline{r})$  e assim maximizar  $P(\hat{\underline{v}} = \underline{v}/\underline{r})$  onde:

$$P(\hat{\underline{v}} = \underline{v}/\underline{r}) = \frac{P(\underline{r}/\underline{v}) \cdot P(\underline{v})}{P(\underline{r})} \quad (2.26)$$

o que implica em maximizar  $P(\underline{r}/\underline{v})$  para todo  $\underline{r}$ .

Agora para um DMC:

$$P(\underline{r}/\underline{v}) = \prod_{i=0}^{N-1} P(r_i/v_i) \quad (2.27)$$

logo:

$$\log P(\underline{r}/\underline{v}) = \sum_{i=0}^{N-1} \log P(r_i/v_i) \quad (2.28)$$

em que  $N = n(L + m)$  dígitos codificados, e

$$P(r_i/v_i)$$

é uma probabilidade de transição.

Portanto, deve-se maximizar a função  $\log P(\underline{r}/\underline{v})$  que é a função de verossimilhança.

## 2.7.2 - O Algoritmo de Viterbi

O Algoritmo de Viterbi é um dos algoritmos utilizados na decodificação de códigos convolucionais, o qual determina a partir da seqüência recebida, o caminho através da treliça que possui a maior métrica, ou seja, o caminho de máxima verossimilhança.

As métricas normalmente utilizadas neste algoritmo são definidas como:

$$\text{Métrica de caminho} : \quad M(\underline{r}/\underline{v}) \triangleq \log P(\underline{r}/\underline{v}) \quad (2.29)$$

$$\text{Métrica de ramo} : \quad M(\underline{r}_i/\underline{v}_i) \triangleq \log P(\underline{r}_i/\underline{v}_i) \quad (2.30)$$

$$\text{Métrica de bit} : \quad M(r_i/v_i) \triangleq \log P(r_i/v_i) \quad (2.31)$$

logo podemos escrever a métrica de caminho como:

$$M(\underline{r}/\underline{v}) = \sum_{i=0}^{L+m-1} M(\underline{r}_i/\underline{v}_i) = \sum_{i=0}^{N-1} M(r_i/v_i) \quad (2.32)$$

Note que a métrica parcial de caminho para os  $j$  primeiros ramos é dada por:

$$M([\underline{r}/\underline{v}]_j) = \sum_{i=0}^{j-1} M(\underline{r}_i/\underline{v}_i) \quad (2.33)$$

O algoritmo de Viterbi processa o vetor recebido  $r$  de maneira iterativa. A cada passo, ele compara a métrica de todos os caminhos que chegam a cada estado (nó) e armazena o caminho de maior métrica (caminho sobrevivente) juntamente com sua métrica. Pode-se descrevê-lo sucintamente através dos seguintes passos:

- Passo 1 : A partir da seqüência recebida, computa a métrica parcial de cada caminho que chega a cada nó;
- Passo 2 : Armazena o caminho de maior métrica (sobrevivente) e sua respectiva métrica, para cada nó;
- Passo 3 : Para cada nó, calcula a métrica parcial para todos os caminhos que chegam em um nó, adicionando a métrica do ramo que chega naquele nó com a métrica do caminho sobrevivente na unidade de tempo anterior;
- Passo 4 : Armazena o caminho com maior métrica e sua respectiva métrica, para cada nó;
- Passo 5 : Retorna ao passo 3 até que a decodificação da seqüência recebida esteja completa.

Se mais de um caminho classificado como sobrevivente chegar a um mesmo nó, a escolha é arbitrária sobre qual caminho será efetivamente considerado como sobrevivente, já que a decodificação subsequente não ajudará a resolver o impasse.

A cada quadro de transições na treliça teremos para cada nó um sobrevivente e sua respectiva métrica armazenados. Assim, a decodificação de Viterbi é limitada na prática a códigos cujos codificadores têm comprimento total de memória (CTM) relativamente pequenos, uma vez que o número total de operações de decodificação de

cada bit cresce exponencialmente com o CTM, [1, 8, 11].

## 2.8 - OS LIMITANTES SUPERIORES DA PROBABILIDADE DE ERRO DE EVENTO E DE BIT

A obtenção da expressão para a probabilidade de erro é extremamente difícil, assim, limitantes passam a ter importância muito significativa, [16].

A derivação desses limitantes será feita a partir da representação do diagrama de treliça de um codificador convolucional, o que se ajusta muito bem aos propósitos que se seguem.

A Figura 2.9 mostra dois caminhos na treliça que constituem duas possíveis seqüências de estados no codificador.

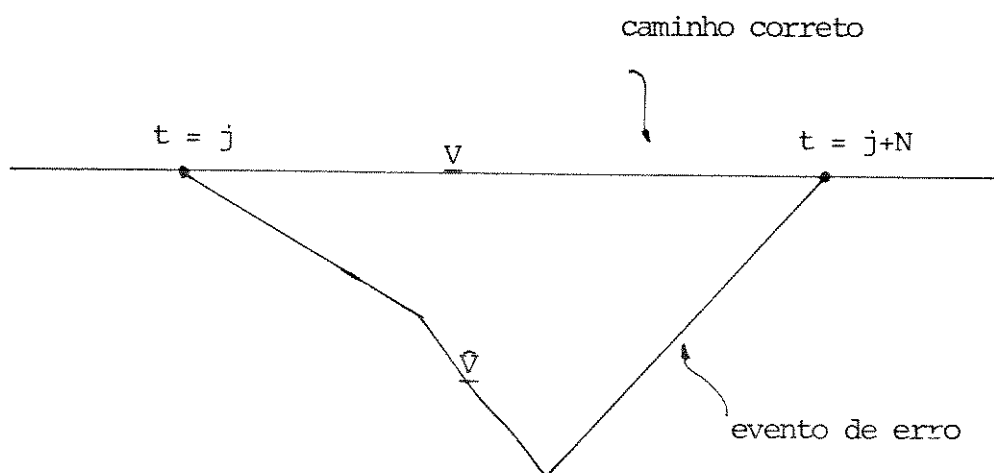


Figura 2.9 - Evento de erro

Por serem os códigos convolucionais lineares, podemos associar a  $v$  o caminho da seqüência codificada nula, o que corresponde à seqüência de informação também toda nula, como caminho correto; e

associar a  $\hat{\underline{V}}$  o caminho escolhido pelo decodificador de máxima verossimilhança. Assim, se  $\hat{\underline{V}}$  for escolhido, um evento de erro ocorrerá. Isso acontecerá se a métrica acumulada do caminho  $\underline{V}$  for menor do que a métrica acumulada do caminho  $\hat{\underline{V}}$ . Dessa forma, a condição necessária para a ocorrência de um evento de erro é que a métrica do caminho incorreto que divergiu do caminho correto no instante  $t=k$  acumule incrementos maiores do que os do caminho correto durante o período em que os mesmos divergem.

Então, a probabilidade de um evento de erro no instante  $t=k$  pode ser limitada superiormente pela probabilidade de que qualquer caminho que divirja do caminho correto no instante  $t=k$  acumule métrica total maior do que aquela acumulada pelo caminho correto, isto é, [16]:

$$P_e(k) \leq \left\{ \sum_{\hat{\underline{V}} \in \underline{V}} \Delta M(\hat{\underline{V}}, \underline{V}) \geq 0 \right\} \quad (2.34)$$

onde:

- $\hat{\underline{V}}$  é o caminho incorreto que divergiu do caminho correto em  $t=k$
- $\underline{V}$  é o conjunto de todos os possíveis caminhos incorretos na treliça em  $t=k$
- $\Delta M(\hat{\underline{V}}, \underline{V})$  é a diferença de métricas de  $\underline{V}$  e  $\hat{\underline{V}}$  durante o período em que os mesmos são distintos.

Usando o limitante da união em (2.34) teremos, [16]:

$$P_e(k) \leq \sum_{\hat{\underline{V}} \in \underline{V}} P_r \left\{ \Delta M(\hat{\underline{V}}, \underline{V}) \geq 0 \right\} \quad (2.35)$$



Supondo, agora, que  $\hat{\underline{V}}$  e  $\underline{V}$  são distintos durante um intervalo de comprimento  $N$  e que  $\hat{\underline{V}}$  e  $\underline{V}$  são os únicos dois possíveis caminhos na treliça, a probabilidade de erro deste par, representada por  $P(\underline{V} \rightarrow \hat{\underline{V}})$ , e fazendo-se uso do limitante de Chernoff com parâmetro  $\lambda \geq 0$ , tem-se que:

$$\begin{aligned}
 P(\underline{V} \rightarrow \hat{\underline{V}}) &= P_r \left\{ \sum_{i=j}^{j+N} M(r_i, v_i) \leq \sum_{i=j}^{j+N} M(r_i, \hat{v}_i / \underline{v}) \right\} \\
 &= P_r \left\{ \sum_{i=j}^{j+N} |M(r_i, \hat{v}_i) - M(r_i, v_i)| \geq 0 / \underline{v} \right\} \\
 &= E \left\{ \exp \left[ \sum_{i=j}^{j+N} \lambda |M(r_i, \hat{v}_i) - M(r_i, v_i)| \right] / \underline{v} \right\} \\
 &= \prod_{i=j}^{j+N} E \left\{ \exp \lambda \cdot |M(r_i, \hat{v}_i) - M(r_i, v_i)| \right\} / \underline{v} \quad (2.36)
 \end{aligned}$$

Definindo-se:

$$D(v, \hat{v}; \lambda) = E \left\{ | \lambda \cdot [M(r_i, \hat{v}_i) - M(r_i, v_i)] | / \underline{v} \right\} \quad (2.37)$$

então (2.36) fica:

$$P(\underline{V} \rightarrow \hat{\underline{V}}) \leq \prod_{i=j}^{j+N} D(v_i, \hat{v}_i; \lambda) \quad (2.38)$$

assumindo que



$$D(v, \lambda) = \begin{cases} 1, & \text{se } v = \hat{v} \\ D(\lambda) & \text{se } v \neq \hat{v} \end{cases}$$

e que o canal é do tipo entrada binária e saída simétrica (2.38) passa a ser:

$$P(\underline{v} \rightarrow \underline{\hat{v}}) \leq |D(\lambda)|^{d(\underline{v}, \underline{\hat{v}})} \quad (2.39)$$

onde:

-  $d(\underline{v}, \underline{\hat{v}})$  é a distância de Hamming entre  $\underline{v}$  e  $\underline{\hat{v}}$ .

Minimizando (2.39) em relação a  $\lambda > 0$ , teremos:

$$P(\underline{v} \rightarrow \underline{\hat{v}}) \leq D^{d(\underline{v}, \underline{\hat{v}})} \quad (2.40)$$

Se  $M(\underline{r}, \underline{v})$  é a métrica de máxima verossimilhança, então:

$$M(\underline{r}, \underline{v}) = a \ln[P(\underline{r}/\underline{v})] + b$$

com  $a \geq 0$ , então  $D(v, \hat{v}; \lambda)$  é a função de Bhattacharyya. Dessa forma, dado que existem  $a[d(\underline{v}, \underline{\hat{v}})]$  caminhos incorretos com distância de Hamming  $d(\underline{v}, \underline{\hat{v}})$  do caminho correto, então:

$$P_e(k) \leq \sum_{d(\underline{v}, \underline{\hat{v}}) \geq d} a[d(\underline{v}, \underline{\hat{v}})] \cdot D^{d(\underline{v}, \underline{\hat{v}})} \quad (2.41)$$

Observe que o número médio de bits em erro decorrentes do caminho  $\underline{\hat{v}}$  que difere do caminho  $\underline{v}$  pode ser limitado superiormente pelo

número de bits errôneos que ocorrem no caminho incorreto. Assim:

$$P_b \leq (1/k) \sum_{i=1}^{\infty} \sum_{d(\underline{v}, \underline{\hat{v}}) \geq d} i \cdot a[d(\underline{v}, \underline{\hat{v}}), i] \cdot d(\underline{v}, \underline{\hat{v}}) \quad (2.42)$$

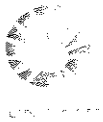
Assumindo que a seqüência de informação é toda nula (caminho correto) implicará que os eventos de erro corresponderão ao número de "1" (uns) na seqüência de informação durante o período que os caminhos divergem. Assim (2.42) fica:

$$P_b \leq (1/k) \sum_{i=1}^{\infty} \sum_{d(\underline{0}, \underline{\hat{v}}) \geq d} i \cdot a[d(\underline{0}, \underline{\hat{v}}), i] \cdot D^d(\underline{0}, \underline{\hat{v}}) \quad (2.43)$$

onde  $a[d(\underline{0}, \underline{\hat{v}}), i]$  é o número de caminhos que diferem da seqüência toda nula com  $i$  "1" (uns).

Da propriedade estrutural de códigos convolucionais, sabemos que derivando (2.20) com relação à  $Z$  observamos que esta função enumeradora fornecerá o número médio de bits errôneos. Como para taxas  $R = k/n$  cada ramo contém 'k' bits, a probabilidade de erro de bit é limitada superiormente por, [16]:

$$P_b \leq (1/k) \frac{d}{dZ} T(X, Z) \Big|_{Z=1} \quad (2.44)$$



## 2.9 - A REPRESENTAÇÃO DE UM CODIFICADOR CONVOLUCIONAL POR VETORES COEFICIENTES

A notação e a representação para os codificadores convolucionais apresentadas a seguir serão utilizadas para a apresentação dos resultados da procura dos melhores codificadores, dentro de cada classe simples no Capítulo 5.

Denota-se por  $\{G_{kn}; m_1, m_2, \dots, m_k\}$  a classe simples de codificadores com 'k' entradas e 'n' saídas, onde  $m_i$  com  $1 \leq i \leq k$  é o número de registradores de deslocamento contidos na  $i$ .ésima linha de entrada do codificador.

### 2.9.1 - Os Vetores Coeficientes

Além da representação matricial, os codificadores  $G_{kn}$  também podem ter uma representação única, através de um conjunto de 'n' vetores  $VC(j)$  com  $j = 1, 2, \dots, n$ , os quais são associados a cada classe simples  $\{G_{kn}; m_1, \dots, m_k\}$  de codificadores, através de sua estrutura de memória.

"Cada vetor  $VC(j)$  com  $j = 1, 2, \dots, n$  corresponde a um arranjo especial do vetor de todos os 'k+M' coeficientes da matriz geradora relativos à saída 'j', ou ainda, os coeficientes dos polinômios da coluna 'j' em  $G(D)$ ", [12].

Para o codificador da Figura 2.1 temos:

$$G(D) = [G_{11}(D) \quad G_{12}(D)]$$

$$\begin{cases} G_{11}(D) = g_{11}^0 + g_{11}^1 D + g_{11}^2 D^2 = 1 + D^2 \\ G_{12}(D) = g_{12}^0 + g_{12}^1 D + g_{12}^2 D^2 = 1 + D + D^2 \end{cases}$$

$$G(D) = [1 + D^2 \quad 1 + D + D^2]$$

$$\begin{cases} VC(1) = [g_{11}^2 \quad g_{11}^1 \quad g_{11}^0] = [1 \quad 0 \quad 1] \\ VC(2) = [g_{12}^2 \quad g_{12}^1 \quad g_{12}^0] = [1 \quad 1 \quad 1] \end{cases}$$

Cabe observar que a ordenação dos vetores  $VC(j)$ ,  $j=1, \dots, n$  é a mesma utilizada em [12], no sentido de cima para baixo e da direita para a esquerda na representação esquemática do codificador. Em [3], foi realizada esta ordenação no sentido de baixo para cima e da direita para a esquerda (Figura 2.11). Em [12], foi observado que a ordenação aqui apresentada acarreta a eliminação de um maior número de codificadores equivalentes quando da procura dos melhores códigos do que na realizada em [3].

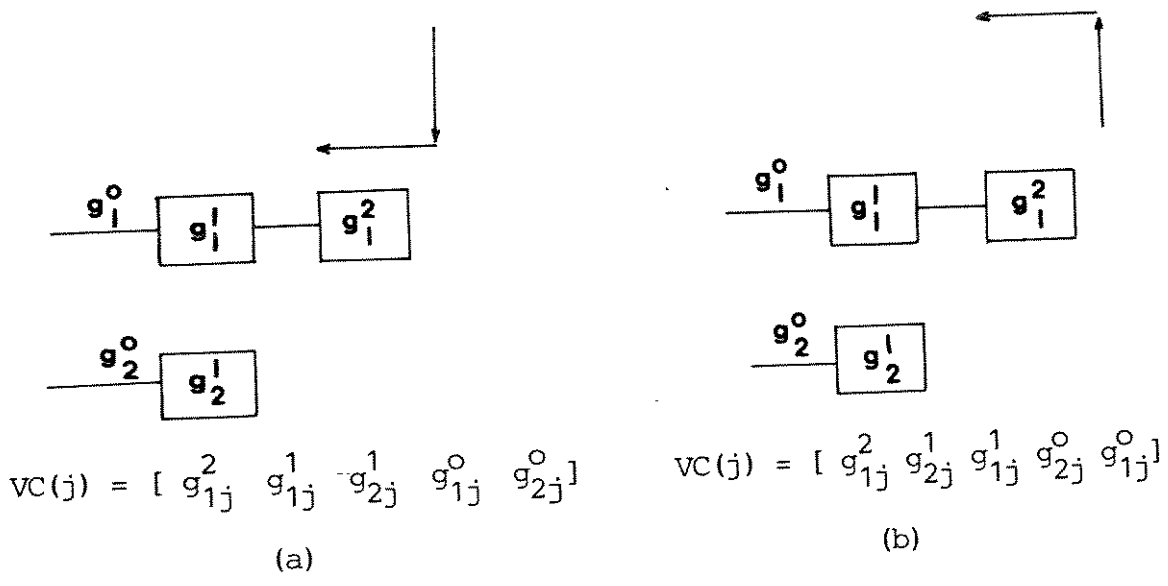


Figura 2.11 - Ordenação dos vetores coeficientes segundo:  
 (a) [12]; (b) [3]

2.10 - OS CÓDIGOS CONVOLUCIONAIS BINÁRIOS DE MEMÓRIA UNITÁRIA

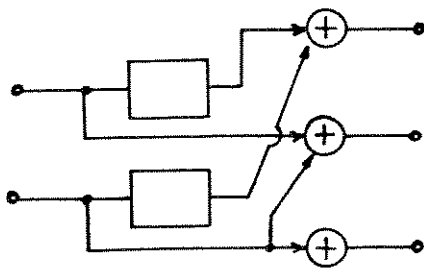
Um dos objetivos deste trabalho é a avaliação do desempenho de códigos TCM (definidos no Capítulo 3) com estrutura de memória unitária; assim, aqui se faz necessário a conceituação dos códigos convolucionais binários de memória unitária.

**Definição 2.10.1** - Um codificador convolucional binário  $\{G_{kn}; m_1, \dots, m_k\}$  de taxa  $R = k/n$  é dito de memória unitária quando:

$$m_1 = m_2 = \dots = m_k = 1$$

Conceitualmente podemos entender os codificadores convolucionais binários de memória unitária como sendo aqueles para os quais todos os 'k' bits de entrada sofrem uma e apenas uma unidade de tempo de atraso até atingirem as saídas do codificador. Assim, podemos dizer que estes codificadores apresentam apenas um estágio de registradores de deslocamento, porém, todas as entradas são codificadas.

A Figura 2.12 mostra um codificador convolucional  $(G_{23}; 1,1)$



(a)

$$\begin{cases} VC(1) = (1100) \\ VC(2) = (0011) \\ VC(3) = (0001) \end{cases}$$

(b)

Figura 2.12 - Codificador convolucional de memória unitária  $(G_{23}; 1,1)$

(a) representação esquemática

(b) vetores coeficientes

Os códigos gerados por esta classe de codificadores são aqui referidos como códigos convolucionais de memória unitária.

Segundo Lee, [10], qualquer código convolucional  $(k,n)$  com memória  $M$  pode ser considerado um código convolucional de memória unitária com  $k'=Mk$  e  $n'=Mn$ , mantendo-se a taxa  $R=k/n = k'/n'$  e a complexidade do codificador, ou em outras palavras, para todo

codificador convolucional binário  $(k,n)$  com memória  $M$ , existe um código de memória unitária com  $k'=Mk$  entradas e  $n'=Mn$  saídas equivalentes, podendo-se assim restringir a procura dos melhores códigos para uma dada taxa e mantendo-se a complexidade aos códigos de memória unitária.

Então, podemos restringir a procura dos melhores códigos para uma dada complexidade e taxa aos códigos de memória unitária.

Tomemos um exemplo: para a classe simples  $\{G_{12}; 3\}$  temos um codificador equivalente de memória unitária na classe  $\{G_{36}; 1,1,1\}$ . A Figura 2.13 mostra a estrutura de memória para cada uma dessas classes.

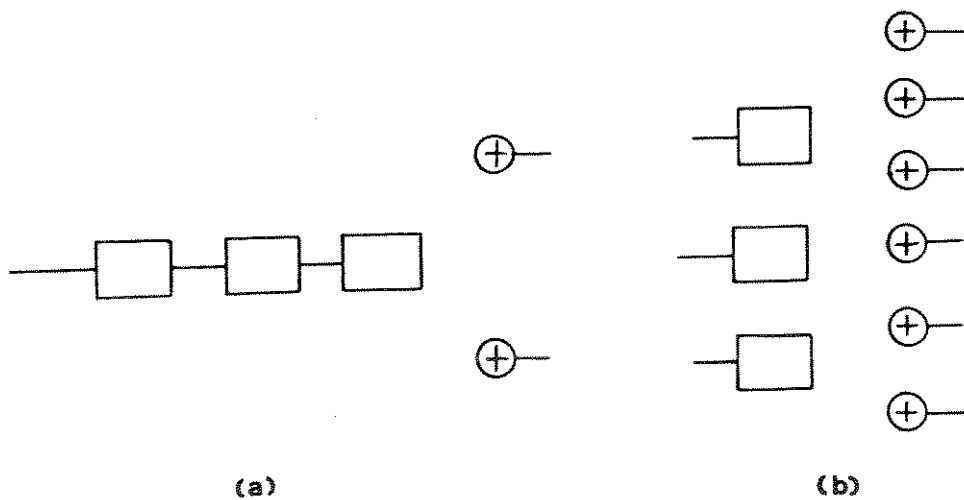


Figura 2.13 - Estrutura de memória para as classes simples

(a)  $\{G_{12}; 3\}$

(b)  $\{G_{36}; 1,1,1\}$



Ainda segundo Lee, [10], a distância livre  $d_{free}$  de um código  $(k, n)$  de memória unitária é limitada superiormente pela maior distância mínima de um  $n'=2n, k'=k$  código de bloco.

A Tabela 2.1 [10] ilustra essa afirmação e ainda mostra que o  $d_{free}$  de um código de memória unitária é sempre maior ou igual que o  $d_{free}$  de qualquer código convolucional de mesma taxa e mesma complexidade com  $\gcd(n, k) = 1$ , onde  $\gcd(n, k)$  denota o máximo divisor comum (greatest common divisor) entre  $n$  e  $k$ .

TABELA 2.1

taxa $k/n$	complexidade de estado	limitante sup. sobre $d_{free}$ (Cod. de bloco)	máximo $d_{free}$ (Mem. Unit.)	máximo $d_{free}$ (cod. gcd=1)
1/4	1	8	7	7
	2	10	10	10
	3	13	13	13
	4	16	16	16
	5	20	20	18
	6	24	24	20
1/3	1	6	5	5
	2	8	8	8
	3	10	10	10
	4	12	12	12
	5	15	15	13
	6	16	16	15
1/2	1	4	3	3
	2	5	5	5
	3	6	6	6
	4	8	8	8
	5	9	9	8
	6	10	10	10
2 2/3	1	4	3	3
	2	6	6	5
	3	8	7	7

Este trabalho procura avaliar a validade desta equivalência entre uma classe simples qualquer  $\{G_{kn}\}$  e a respectiva classe de memória unitária  $\{G_{kn}, 1, 1, 1, \dots\}$  [ $k'=Mk$  e  $n'=Mn$ ] para os códigos TCM definidos no Capítulo 3.

## CAPÍTULO 3

OS CÓDIGOS TCM



### 3.1 - INTRODUÇÃO

Os primeiros sistemas de comunicações digitais desenvolvidos foram do tipo não codificado, onde para cada mensagem (conjunto de 'k' bits que caracterizam um símbolo) existe uma correspondência biunívoca com os pontos de sinal na constelação.

Já em sistemas codificados, as seqüências codificadas são interrelacionadas através do processo de codificação dos bits de informação.

Através de um codificador de taxa  $R = k/n$ , 'k' bits de informação por intervalo de modulação T, são codificados em 'n' bits. Posteriormente os 'n' bits codificados são transmitidos por uma constelação de  $2^n$  sinais (ou símbolos de canal), através do mapeamento entre os bits codificados e o conjunto de sinais.

Considerando a codificação convolucional e a modulação no espaço de sinais como entidade única, Ungerboeck introduziu um novo conceito em Teoria de Comunicação que é aqui referido como Modulação por Codificação em Treliça (TCM - trellis-Coded Modulation).

A importância dessa contribuição de Ungerboeck, é devido ao salto quantitativo obtido pelos sistemas de comunicações com relação ao seu desempenho. O uso de códigos TCM permite alcançar ganhos de codificação assintóticos de 3 dB para esquemas simples, e até 6 dB para esquemas mais complexos em relação aos sistemas convencionais não codificados, sem alterar a taxa de transmissão de bits de informação, além de manter a largura de faixa inalterada.

As suposições geralmente admitidas no estudo dos sistemas codificados e que serão consideradas neste trabalho são as seguintes:

- as constelações serão em uma ou duas dimensões com energia média unitária e sinais equiprováveis.
- o canal é do tipo Gaussiano branco limitado em faixa.
- há um perfeito sincronismo entre transmissor e receptor.
- não é considerado o efeito da interferência inter-simbólica.
- a decodificação é suave (soft-decision) por máxima verossimilhança utilizando o algoritmo de Viterbi.

### 3.2 - O PROCESSO DE CODIFICAÇÃO PARA CÓDIGOS TCM

#### 3.2.1 - O Codificador TCM

O codificador TCM pode ser visualizado como um codificador convolucional binário em que suas saídas são mapeadas num conjunto de pontos no espaço de sinais (constelação) como mostra a Figura 3.1:

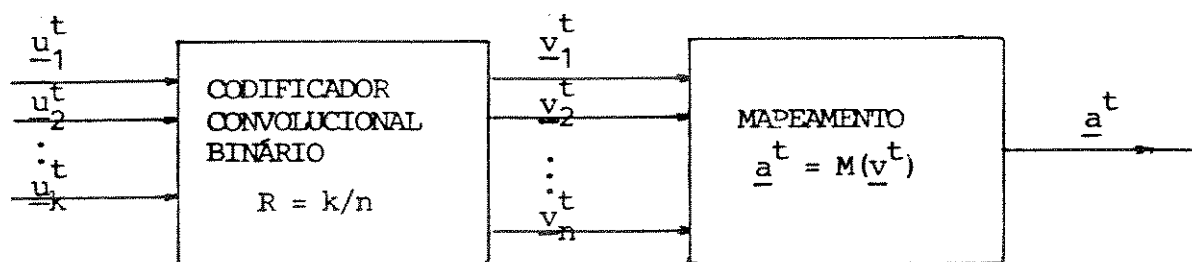


Figura 3.1 - Diagrama em blocos de um codificador TCM

Através do codificador convolucional binário de taxa  $R=k/n$ , 'k' bits de informação são codificados em 'n' bits. Posteriormente esses 'n' bits são mapeados em uma constelação de  $2^n$  pontos de sinais a cada intervalo de modulação T.



Assim, como os códigos convolucionais, os códigos TCM podem ser descritos pelo diagrama de estados e também pelo diagrama de treliça, uma vez que sua estrutura está relacionada com a do codificador convolucionar interno. Então tem-se palavras código ramo do código convolucionar formando  $n$ -úplas,  $q$ -árias e as palavras código ramo do código TCM que são  $n$ -úpla reais. Essas palavras código ramo do código TCM estão associadas aos subconjuntos terminais resultantes do mapeamento aplicado às palavras código ramo do código convolucionar. Este mapeamento é obtido por sucessivas partições da constelação de sinais em subconjuntos e é chamado mapeamento por partição de conjunto (set partitioning) que é descrito a seguir.

### 3.3 - A PARTIÇÃO DO CONJUNTO DE SINAIS E O MAPEAMENTO

#### 3.3.1 - A Partição do Conjunto de Sinais (Set Partitioning)

Ungerboeck [19] estabeleceu as seguintes regras para a partição do conjunto de sinais:

1. O conjunto de sinais (constelação) é dividido sucessivamente em dois subconjuntos de forma que as distâncias mínimas Euclidianas entre os sinais sejam maiores do que ou no mínimo iguais àquelas do conjunto que lhes deu origem ( $\Delta_0 \leq \Delta_1 \leq \Delta_2 \dots \leq \Delta_n$ );
2. Repete-se a partição até se obter  $2k'$  subconjuntos terminais, onde  $k'$  é o número de saídas "n" menos o número de entradas não codificadas

A Figura 3.2 mostra a partição do conjunto de sinais da constelação 8-PSK.

Observa-se que a partição não determina uma única forma de subdivisão de conjuntos, apenas estabelece uma relação de distâncias mínimas crescentes a cada subdivisão. Assim, a distância mínima entre dois subconjuntos será a distância mínima entre os pontos do subconjunto que os originou.

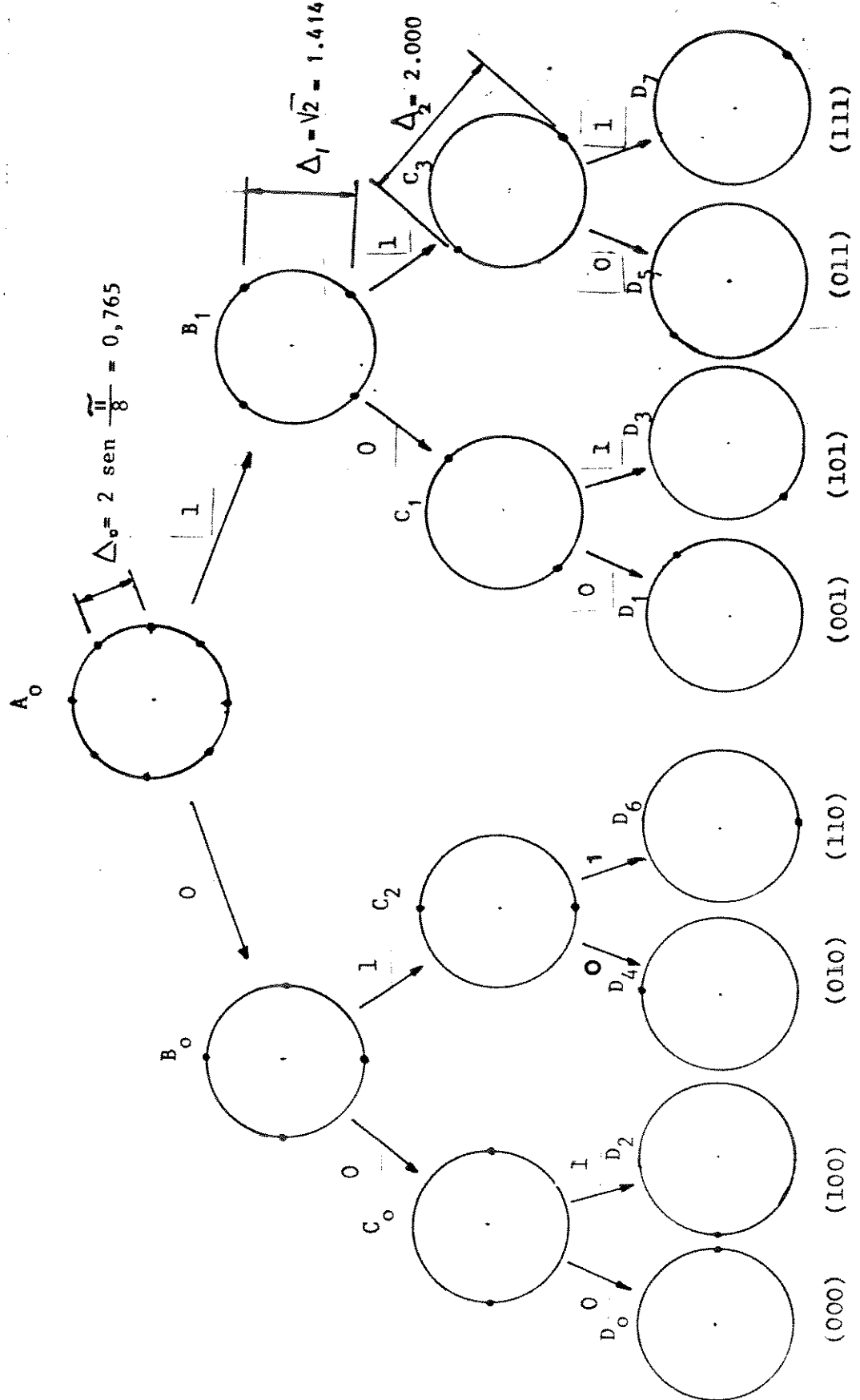


Figura 3.2 - Partição de conjunto de sinais (set partitioning) para a constelação 8-PSK



### 3.3.2 - O mapeamento

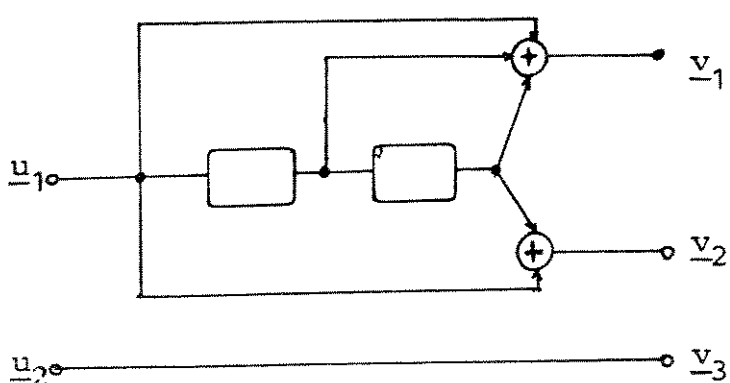
Com objetivo de apresentar uma maior distância Euclidiana entre os sinais dos subconjuntos terminais da partição do que aquela entre os sinais do conjunto original, faz-se o mapeamento, que consiste em associar às transições entre estados (nós) que ocorrem na treliça os subconjuntos terminais obtidos através das regras de partição.

Ungerboeck [19], por construção heurística, chegou a considerações que, mesmo sem prova formal de que resultarão nos códigos de desempenho ótimo, são comuns a todos os melhores códigos encontrados.

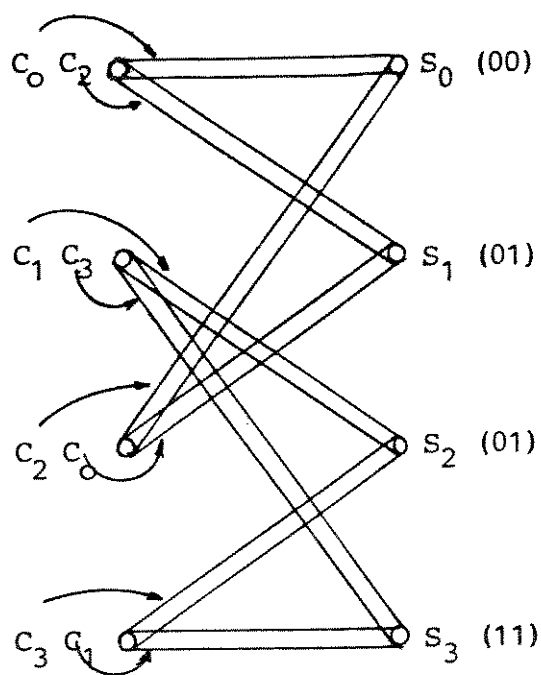
Após selecionar a estrutura de treliça, associa-se os  $2^n$  sinais da constelação às transições de treliça de modo a obter a máxima distância mínima Euclidiana (equação 2.18) para o código, obedecendo-se às seguintes regras:

1. Todos os sinais da constelação devem ocorrer com a mesma frequência, determinando a simetria da constelação;
2. Os subconjuntos associados às transições paralelas na treliça (transições entre nós consecutivos, que começam e terminam nos mesmos estados) devem ter a maior distância Euclidiana possível entre os sinais que os compõe;
3. Os subconjuntos terminais são associados aos ramos da treliça do codificador TCM de modo a garantir a maior distância mínima Euclidiana entre todos os ramos que chegam ou partem de um mesmo nó, guardando-se a restrição de que esta seja inferior à do item 2.

Para ilustrar as regras estabelecidas por Ungerboeck, a Figura 3.3b mostra um quadro de transições de um código TCM, com taxa  $R=2/3$  e  $m=2$ , com transições paralelas, referente ao codificador da Figura 3.3a associado à constelação 8-PSK, cuja partição é mostrada na Figura 3.2.



(a)



(b)

Figura 3.3 - (a) representação esquemática de um codificador convolucional  $\{G_{23}; 2\}$

(b) célula de treliça para este codificador associado à constelação 8-PSK

O mapeamento é feito associando-se o conjunto de pontos do subconjunto  $C_0$  (Figura 3.2) às transições paralelas que interligam os estados  $S_0$ . O subconjunto  $C_2$  é o que apresenta a maior distância Euclidiana em relação à  $C_0$ , por isso é escolhido para ser associado às transições paralelas que partem do estado  $S_0$  para o estado  $S_1$ . Seguindo o mesmo raciocínio, associa-se às transições de  $S_1$  a  $S_2$  o subconjunto  $C_1$  e portanto o subconjunto  $C_3$  às transições de  $S_1$  a  $S_3$ .

Mantendo-se a mesma frequência e a máxima distância Euclidiana entre os subconjuntos que partem e chegam a um mesmo nó, de  $S_2$  para  $S_0$  associa-se  $C_2$  e de  $S_2$  para  $S_1$   $C_0$ ; da mesma forma de  $S_3$  para  $S_1$  associa-se  $C_3$ , e de  $S_3$  para  $S_2$  o subconjunto  $C_1$  é associado.

Observa-se que a regra 2 está sendo obedecida, pois a distância entre as transições paralelas é a distância entre os sinais dos subconjuntos  $C_0$ ,  $C_1$ ,  $C_2$  ou  $C_3$  que é  $\Delta_2 = 2.000$ , que é sempre maior que as distâncias entre os subconjuntos  $C_0$  e  $C_2$  ou  $C_1$  e  $C_3$ , que é  $\Delta_1 = \sqrt{2}$ .

### 3.4 - A DISTÂNCIA MÍNIMA DO CÓDIGO TCM

Pode-se definir o peso Euclidiano, de um caminho  $V$ , de comprimento  $L_V$ , como [12]

$$w_E(V, L_V) = \left\{ \sum_{t=1}^{L_V} w_E^2(\underline{v}^t) \right\}^{1/2} \quad (3.1a)$$

$$w_E(\underline{v}^t) = \Delta_q(\underline{v}^t) \quad (3.1b)$$

onde:

$q(\underline{v}^t)$  é o número de zeros seguidos, a partir do bit de mais baixa ordem, no byte codificado não-nulo  $\underline{v}^t$  do caminho  $V$

$w_E(\underline{v}^t) = \Delta_q(\underline{v}^t)$  representa o peso Euclidiano do byte codificado não-nulo  $\underline{v}^t$  de  $V$ , o qual é dado pela distância mínima do subconjunto cujo índice é igual a  $q(\underline{v}^t)$  [por definição  $w_E(\phi) = \phi$ ]

"O peso Euclidiano do byte codificado  $\underline{v}^t$  nada mais é do que um limitante inferior da menor distância relacionada a um padrão de erro igual a  $\underline{v}^t$ ".

Ungerboeck [20] derivou um limitante inferior para a distância mínima Euclidiana, em termos dos pesos Euclidianos (equação 3.1) de forma que:

$$d_{\min} \geq \Delta_{\min} = \min_{V \neq \phi} w_E(V, L_V) \quad (3.2)$$

A procura de códigos de desempenho ótimo, que é objetivo deste trabalho, será feita segundo o parâmetro de máximo  $\Delta_{\min}$  (equação 3.2) conforme exposto no Capítulo 4.

Mostramos a seguir o processo para determinação do  $\Delta_{\min}$  para o código TCM da Figura 3.3.

Devemos procurar na estrutura de treliça deste código os caminhos que divergem e retornam ao estado  $S_0$  (00) que é tomado como referência e após calculadas as distâncias entre cada um desses caminhos e a seqüência toda nula, teremos que um possível  $\Delta_{\min}$  será a menor dessas distâncias. Esta distância mínima só será o  $\Delta_{\min}$  se for

inferior à distância entre as transições paralelas, caso contrário esta será o  $\Delta_{\min}$ .

No exemplo temos:

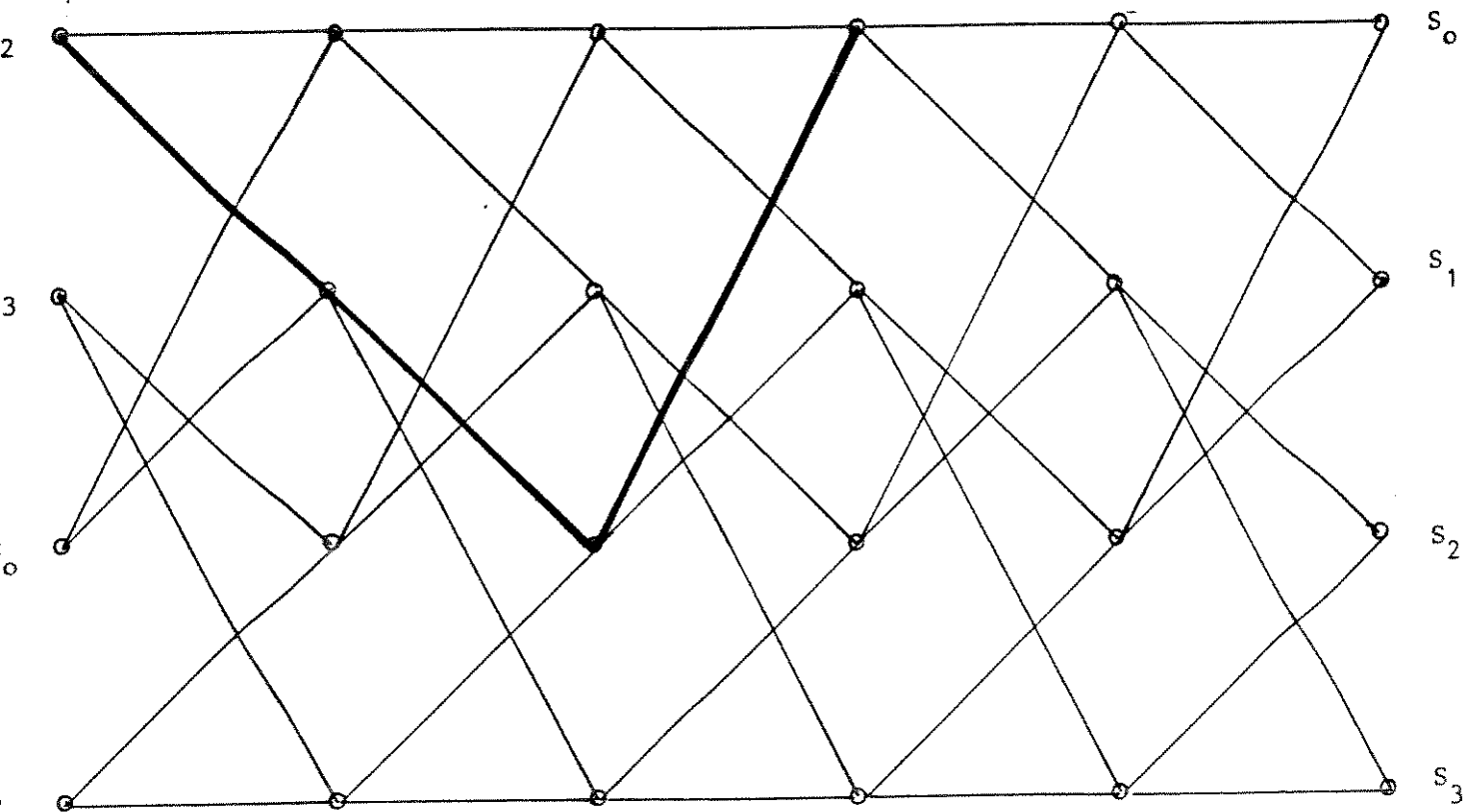


Figura 3.4 - Cálculo de  $\Delta_{\min}$  para o código da Figura 3.3

O caminho em destaque é o que apresenta a menor distância em relação à seqüência toda nula e essa distância é:  $d^2 = d^2(C_0 C_2) + d^2(C_0 C_1) + d^2(C_0 C_2)$ , então:  $d^2 = \Delta_1^2 + \Delta_0^2 + \Delta_1^2 = 2\Delta_1^2 + \Delta_0^2$  de acordo com o conjunto particionado da Figura 3.2.

Assim:

$$d^2 = 4.585$$

Comparando-se esse valor com a distância entre as transições paralelas que é  $\Delta_2^2 = 4.000$  temos que  $d^2$  é maior que  $\Delta_2^2$  e então  $\Delta_{\min} = \Delta_2$  portanto,  $\Delta_{\min} = 2.000$ , e dizemos que este é limitado pelas transições paralelas.

### 3.5 - A ESTRUTURA DOS CÓDIGOS TCM

Para a determinação da distância mínima dos códigos TCM é necessária a comparação aos pares de todas as seqüências, o que demandaria um grande esforço computacional. No intento de solucionar essa questão, revisaremos aqui alguns conceitos básicos sobre a estrutura dos códigos TCM, estabelecidos por Marsan et alii, [15].

**Definição 3.5.1** - "Um código C é dito linear se:

- (a) existe uma operação chamada de 'soma' e descrita pelo símbolo '+', em relação a qual o alfabeto de saída ou de canal, {a}, formando um grupo abeliano;
- (b) Existe um homomorfismo entre o alfabeto de entrada {u} e o alfabeto de canal {a}. Isso implica que o conteúdo satisfaz a propriedade:

$$f(\underline{u}_i + \underline{u}_j) = f(\underline{u}_i) + f(\underline{u}_j)" \quad [12]$$

onde f(.) é a transformação associada ao codificador.

**Teorema 3.5.1** - Dados 'n', 'k' e 'M', o conjunto de todos os códigos lineares de taxa  $R = k/n$  e 'M', o comprimento total de memória pode

ser obtido cascadeando-se um codificador convolucional  $(G_{kn}; M)$ , com um mapeamento biunívoco do alfabeto de saída do código convolucional  $\underline{v}$ , ao alfabeto de modulação  $\{a\}$ .

Pelo teorema 3.5.1, segundo a definição 3.5.1 e da descrição dos códigos TCM, verifica-se que estes são lineares.

**Definição 3.5.2** - Um código  $C$  é dito superlinear se é linear, segundo a definição 3.5.1 e:

(c) é possível associar a cada símbolo de canal 'a' um número real positivo  $w(a)$ , chamado peso de 'a', tal que  $d(a_i, a_j) = w(a_i + a_j)$ . Estes pesos são uma extensão do peso de Hamming e  $d(a_i, a_j)$  é a distância Euclidiana entre os símbolos  $a_i$  e  $a_j$ .

**Teorema 3.5.2** - "Um código  $C$  é dito superlinear se e somente se símbolos iguais na matriz adição do alfabeto de canal  $\{a\}$  corresponde a distâncias iguais nas posições respectivas da matriz distância, ou seja:

$$\text{se } a_i + a_j = a_u + a_v \rightarrow d(a_i, a_j) = d(a_u, a_v)''$$

Da definição 3.5.2 e do Teorema 3.5.2 verifica-se que a distância mínima dos códigos TCM superlineares pode ser calculada tomando-se uma seqüência como referência (geralmente a seqüência toda nula) ao invés de se comparar duas a duas, o que diminui em muito o esforço computacional.

É importante observar que os códigos TCM, devido à métrica utilizada (métrica Euclidiana), nem sempre obedecem ao Teorema 3.5.2.

No entanto, verifica-se que a partir da definição do peso Euclidiano (equação 3.1) e de sua utilização, em lugar de distâncias Euclidianas reais entre os sinais, os códigos TCM passam a satisfazer esse Teorema, ou seja, podem ser analisados como se fossem superlineares, podendo-se assim usar uma seqüência de referência na determinação da distância mínima, que na verdade não estará sendo calculada e sim um limitante inferior:  $\Delta_{\min}$  (equação 3.2) para a mesma.

### 3.6 - O DESEMPENHO DE SISTEMAS DE COMUNICAÇÕES CODIFICADOS

O desempenho dos sistemas de comunicações que utilizam códigos TCM, em relação ao dos sistemas não codificados, pode ser determinado pelo ganho de codificação assintótico (GCA), ou ainda, através da análise da probabilidade de evento de erro em função da relação sinal-ruído.

#### 3.6.1 - O Ganho de Codificação Assintótico (GCA)

O ganho de codificação assintótico para sistemas codificados, usando constelações com energia média unitária e sinais equiprováveis em relação a sistemas não codificados nas mesmas condições, e para altos valores de relação sinal-ruído, pode ser definido como:

$$\text{GCA} = 20 \log \left( \frac{d_{\min}}{d_o} \right) \quad (\text{dB}) \quad (3.3)$$



onde:

$d_{\min}$  é a distância mínima Euclidiana do sistema codificado  
 $d_o$  é a distância mínima do sistema não codificado tomado como referência

### 3.6.2 - A Probabilidade de Evento de Erro [P(e)]

Uma outra maneira de se determinar o desempenho de um sistema de comunicações codificado é através da análise da probabilidade de evento de erro em função da relação sinal-ruído. Para altos valores dessa relação, a probabilidade de evento de erro aproxima-se assintoticamente do limitante inferior dado por, [8]:

$$P(e) \geq N_{d_{\min}} \cdot Q(d_{\min} / \sigma) \quad (3.4)$$

onde:

$N_{d_{\min}}$  é o número médio de seqüências com  $d_{\min}$

$\sigma^2$  é a variância do ruído

$Q(\cdot)$  é a função de probabilidade de erro Gaussiana dada por:

$$Q(x) = (1/\sqrt{2\pi}) \int_x^{\infty} \exp(-x^2/2) dx$$

Para valores práticos de relação sinal-ruído, o ganho de codificação é geralmente menor que o GCA, e para melhor estimação do

desempenho do sistema a probabilidade de evento de erro deve ser obtida por meio de limitantes ou simulação, [2].

## CAPÍTULO 4

A PROCURA DOS CÓDIGOS TCM DE DESEMPENHO ÓTIMO

## 4.1 - INTRODUÇÃO

Neste capítulo, descrevemos o procedimento [3, 12] para a procura dos códigos TCM de desempenho ótimo, para um dado esquema modulação/codificação e para classes simples de codificadores  $(G_{kn} \dots; m_1, m_2, \dots, m_k)$ , bem como os objetivos dessa procura. Portanto, é necessário aqui se definir e esclarecer alguns elementos que serão abordados neste processo.

**Definição 4.1** - "Um codificador  $G_{kn}$ , com comprimento total de memória 'M', é um codificador ótimo se o código a ele relacionado possui o máximo  $\Delta_{min}$  (equação 3.2), obtido entre os codificadores do conjunto total de classes  $\{G_{kn}; M\}$ , ao qual pertence", [3].

É importante salientar que este trabalho utiliza o pacote de software desenvolvido em [12] o qual realiza a procura dos códigos ótimos, sob o ponto de vista do máximo  $\Delta_{min}$  (equação 3.2) que é um limitante inferior para  $d_{min}$  (equação 2.18).

Para a eliminação das representações redundantes de um mesmo codificador (codificadores que possuem suas matrizes geradoras iguais a menos de permutações de linhas), a procura limita-se à estrutura de memória  $m_1 \geq m_2 \geq \dots m_k \geq 0$ , de acordo com o algoritmo proposto em [12].

## 4.2 - OS OBJETIVOS DA PROCURA

Os objetivos da busca dos códigos TCM de desempenho ótimo (sempre sob o ponto de vista de máximo  $\Delta_{min}$ ), estabelecem diretamente

a finalidade deste trabalho.

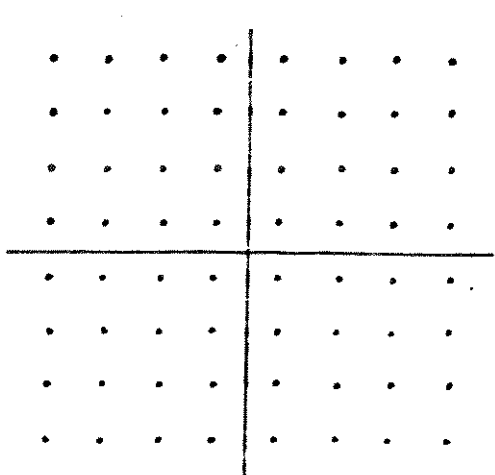
Basicamente podemos resumir esse objetivo em dois tópicos:

1. Estabelecer comparações entre diferentes constelações dentro do sistema combinado (modulação/codificação);
2. Investigar o desempenho dos códigos TCM, cuja estrutura do codificador é a de memória unitária (aqui referidos como códigos TCM de memória unitária), em comparação com outras estruturas de codificadores com mesmo número de entradas 'k', mesmo número de saídas 'n' e mesmo comprimento total de memória 'M'.

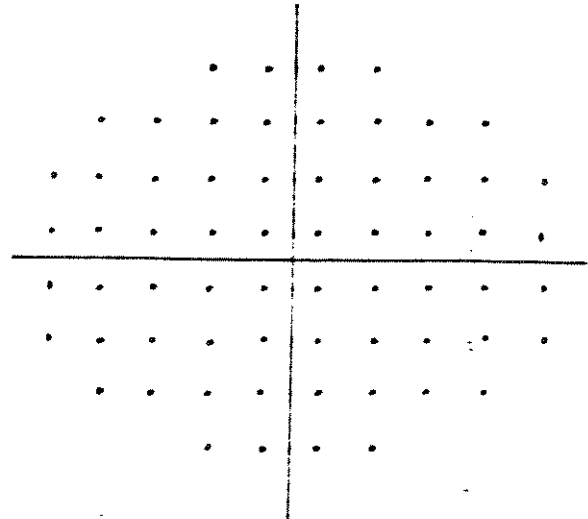
#### 4.2.1 - A Comparação Entre Diferentes Constelações

Forney [9] estabeleceu que a eficiência de um sistema de modulação (constelação) será tanto melhor quanto mais esta constelação se aproximar de um círculo (veja Seção 1.2.2). Assim a procura aqui é feita no sentido de se verificar a validade desta afirmação para os sistemas combinados (modulação/codificação). Então, para a mesma classe simples de codificadores  $\{G_{kn}; m_1, m_2, \dots, m_k\}$  buscar-se-á determinar códigos TCM ótimos para diferentes constelações, estabelecendo-se posteriormente uma comparação entre os desempenhos de cada sistema combinado.

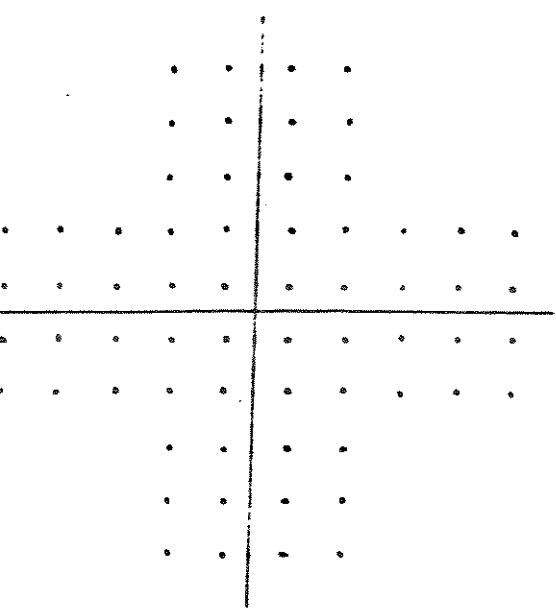
A Figura 4.1 mostra algumas constelações analisadas neste trabalho, bem como a nomenclatura aqui usada para cada uma:



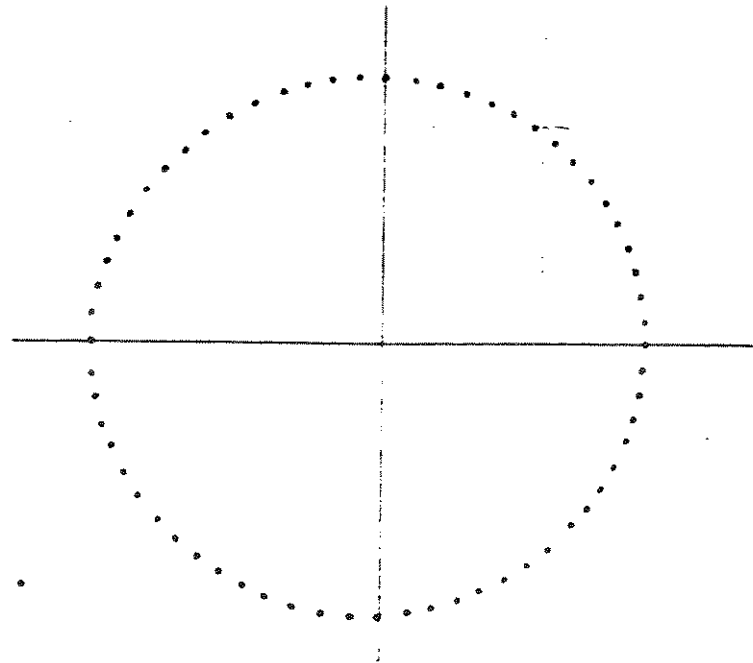
(a) 64-QAM



(b) 64-CROSS(1)



(c) 64-CROSS (2)



(d) 64-PSK

Figura 4.1 - Algumas constelações analisadas neste trabalho

Além das constelações da Figura 4.1, serão utilizadas outras a saber: 4-PSK (4-QAM), 8-PSK, 8-QAM, 16-PSK, 16-QAM, 32-CROSS e 32-PSK, exatamente com o intuito de se traçar comparações entre o desempenho de diferentes constelações para cada classe simples de codificadores, procurando avaliar-se a validade da proposição de Forney [9], de que o desempenho será tanto melhor quanto mais a constelação se aproximar de um círculo para os sistemas combinados. É importante salientar que neste trabalho todas essas constelações estão sujeitas à restrição de energia média unitária, o que implica no fato de que quanto maior o número de pontos na constelação menor será a distância mínima entre esses pontos, o que poderá acarretar uma certa limitação no desempenho do sistema. Essa característica será avaliada no Capítulo 5.

#### 4.2.2 - A Investigação Sobre os Códigos TCM de Memória Unitária

Estabelecer o desempenho dos códigos TCM de memória unitária é o objetivo primeiro desse trabalho.

Lin-Nan Lee [10] estabelece que para qualquer codificador convolucional binário existe um codificador equivalente com estrutura de memória unitária, (veja Seção 2.10), e que assim a procura dos códigos ótimos para classe  $(G_{kn}; M)$  pode se restringir aos códigos de memória unitária; em outras palavras, para cada classe simples de codificadores convolucionais binários existe sempre um codificador com estrutura de memória unitária que é ótimo no sentido de máximo  $d_{free}$ .

A busca realizada neste trabalho tenta investigar se a característica apresentada pelos códigos convolucionais de

memória unitária continua válida para os códigos TCM, ou seja, investigar se a característica se verifica para os sistemas combinados.

Assim, iremos assumir, para cada uma das constelações da Figura 4.1, que a classe simples de codificadores  $\{G_{kn}, M\}$  será utilizada neste processo comparativo. Como resultado teremos que a comparação entre os códigos TCM de memória unitária com as demais estruturas de codificadores estará fundamentada no critério de optimalidade de máximo  $\Delta_{\min}$ .

Por outro lado, para os mesmos códigos TCM de memória unitária, iremos avaliar o desempenho destes códigos quando um fator de multiplicidade de taxas é considerado, isto é, iremos comparar os códigos de memória unitária elementares, taxa  $R=k/n$  com os respectivos múltiplos, taxa  $R = \alpha k/\alpha n$  para  $\alpha \geq 2$ .

#### 4.3 - O PROCESSO DE PROCURA

O algoritmo proposto em [12] estabelece dois tipos de procura para códigos TCM ótimos em cada classe simples  $\{G_{kn}; m_1, \dots, m_k\}$ , denominados: Procura Exaustiva e Procura Não-Exaustiva, respectivamente. Esses dois tipos diferem entre si, substancialmente no número de códigos do CIP (conjunto inicial de procura), que, como o nome estabelece, é o conjunto de codificadores a partir do qual se iniciará a procura. Não discutiremos aqui os detalhes desse algoritmo por entender que o objetivo agora é mostrar suas aplicações a nível computacional para obtenção dos resultados, porém, o leitor poderá consultar [13] para maiores detalhes. Assim, a seguir, apresentaremos



o procedimento prático para cada tipo de procura, exemplificando com dados de entrada e os respectivos resultados de saída.

A procura foi realizada pelos programas elaborados em [12] em linguagem FORTRAN no sistema VAX/MS.

#### 4.3.1 - A Procura Exhaustiva

##### - Dados de Entrada:

- k : número de entradas do codificador
- n : número de saídas do codificador
- $m(i) ; 1 \leq i \leq k$  : número de unidades de memória (registradores de deslocamento na  $i$ -ésima linha de entrada do codificador
- $\Delta_j ; 0 \leq j \leq n-1$  : distância mínima Euclidiana entre os pontos do  $j$ -ésimo subconjunto no "set partitioning"
- $L_{max}$  : número máximo de janelas de tempo na treliça associada ao codificador em que se calculará o  $\Delta_{min}$ .
- DMDF : distância Euclidiana mínima a partir da qual se calculará o  $\Delta_{min}$  do código.

##### - Saída dos Resultados

Os resultados apresentarão os códigos ótimos da classe simples através dos seus vetores coeficientes [VC(j)] em número decimal

VC(j) :  $1 \leq j \leq n$  : vetores coeficientes

e

$$\Delta_{\min}$$

A seguir apresentamos, na forma de exemplo, a procura do tipo exaustiva.

**Exemplo 4.1** : codificador  $(G_{12}; 1)$  associado à constelação 4-PSK

- **Dados de Entrada:**

$$k = 1$$

$$n = 2$$

$$m(1) = 1$$

$$\Delta_o = 1.414$$

$$\Delta_i = 2.000$$

$$L_{\max} = 16$$

$$\text{DMDF} = 1.4$$

- **Saída de Resultados:**

$$VC(1) = 2 = [ 1 \quad 0 ]$$

$$VC(2) = 1 = [ 0 \quad 1 ]$$

$$\Delta_{\min} = 2.449$$

**OBSERVAÇÃO 4.1** - Se a estrutura do codificador apresentar entradas não

codificadas, o que acarretará o aparecimento de transições paralelas na treliça associada ao código, deve-se proceder da seguinte forma:

- Essas entradas e correspondentes saídas são excluídas na introdução dos dados de entradas nos programas;
- O  $\Delta_{\min}$  apresentado como resultado pelos programas deve ser comparado com a distância Euclidiana ( $\Delta$ ) entre os pontos do subconjunto do "set partitioning" associado às transições paralelas, e então:

se o  $\Delta_{\min} < \Delta$  a distância mínima do código é  $\Delta_{\min}$

se  $\Delta_{\min} > \Delta$  a distância mínima do código é  $\Delta$

Vejamos o exemplo 4.2:

Codificador  $[G_{24}; 3,0]$  associado à constelação 16-PSK

Observa-se que apenas uma das duas linhas de entrada do codificador é codificada, assim, os dados de entrada serão:

$$k = 1$$

$$n = 3$$

$$m(1) = 3$$

$$\Delta_0 = 0.390$$

$$\Delta_1 = 0.765$$

$$\Delta_2 = 1.414$$

$$L_{\max} = 10$$

$$\text{DMDF} = 0.390$$

**- Saída de Resultados:**

$$\text{VC}(1) = 4$$

$$\text{VC}(2) = 2$$

$$\text{V}(3) = 9$$

$$\Delta_{\min} = 1.543$$

Como a distância entre os pontos do subconjunto associado às transições paralelas para este código é  $\Delta_3 = 2.000$ , então o  $\Delta_{\min}$  deste código é 2.000. Desta forma, diz-se que este  $\Delta_{\min}$  está limitado pelas transições paralelas.

Para o estabelecimento dos vetores coeficientes do codificador deve-se considerar que sua dimensão é ' $k + M$ '. No exemplo  $k + M = 5$ . A entrada não codificada será representada pelo bit mais significativo (de acordo com a ordenação dos vetores coeficientes), assim teremos:

$$\text{VC}(1) = 16 = ( 1 0 0 0 0 )$$

$$\text{VC}(3) = 2 = ( 0 0 0 1 0 )$$

$$\text{VC}(2) = 4 = ( 0 0 1 0 0 )$$

$$\text{VC}(4) = 9 = ( 0 1 0 0 1 )$$

Logo a estrutura do codificador é a mostrada na Figura 4.2:

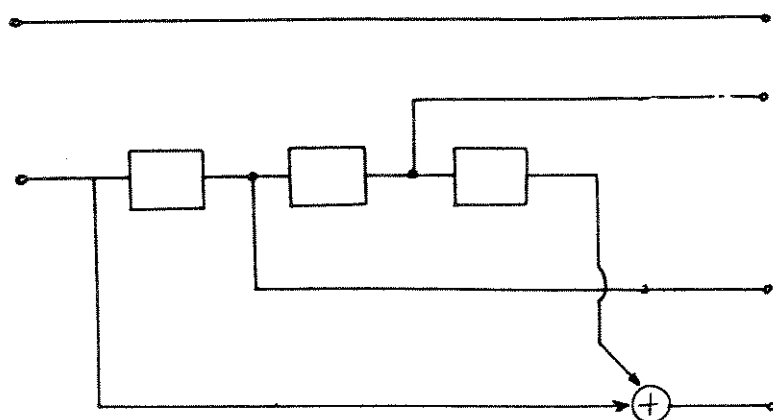


Figura 4.2 - Representação esquemática do codificador do exemplo 4.2

#### 4.3.2 - A Procura Não-Exaustiva

O Conjunto Inicial de Procura (CIP) para o caso da procura não exaustiva é limitado através da introdução nos dados de entrada de elementos denominados máscaras, que fixam pelo menos em parte a estrutura do codificador.

A elaboração dessas máscaras foi estabelecida em [13] e para as procuras realizadas neste trabalho este conceito foi utilizado.

#### - Dados de Entrada:

- Todos os dados de entrada para a procura exaustiva, e mais:

máscara  $VC(j)$  ;  $1 \leq j \leq n$ : máscara para o  $j$ -ésimo vetor coeficiente.

- Saída de Resultados:

- os mesmos da procura exaustiva.

Vejamos o exemplo 4.3:

Codificador  $(G_{40}; 1,1,1,1)$  associado à constelação 64-PSK. A Figura 4.3 mostra as máscaras para este exemplo e a estrutura do codificador com a devida ordenação para os vetores coeficientes:

	1	2	3	4	5	6	7	8
MASC VC(1)	0	0	0	0	1	0	0	0
MASC VC(2)	0	0	0	0	0	1	0	0
MASC VC(3)	0	0	0	1	0	0	0	0
MASC VC(4)	0	0	1	0	0	0	0	0
MASC VC(5)	0	1	0	0	0	0	1	0
MASC VC(6)	1	0	0	0	0	0	X	1

— 5 — 1

— 6 — 2

— 7 — 3

— 8 — 4

(b)

Figura 4.3 - (a) máscaras para o exemplo 4.3  
(b) estrutura do codificador ordenada

O elemento 'X' representa uma variável que o programa determinará ser um '0' ou um '1', e será introduzida nos dados de entrada representada pelo algarismo '2'.

**Dados de Entrada:**

$k = 4$	$\Delta_0 = 0.098$
$n = 6$	$\Delta_1 = 0.196$
$m(1) = 1$	$\Delta_2 = 0.390$
$m(2) = 1$	$\Delta_3 = 0.765$
$m(3) = 1$	$\Delta_4 = 1.414$
$m(4) = 1$	$\Delta_5 = 2.000$

MASC VC(1) = 00001000  
MASC VC(2) = 00000100  
MASC VC(3) = 00010000  
MASC VC(4) = 00100000  
MASC VC(5) = 01000010  
MASC VC(6) = 10000021

**Saída de Resultados:**

VC(1) = 8  
VC(2) = 4  
VC(3) = 16  
VC(4) = 32  
VC(5) = 66  
VC(6) = 129 ou 131

$\Delta_{\min} = 0.402$



Para a procura não-exaustiva também leva-se em consideração a observação 4.1.

Com estes instrumentos buscamos os códigos de desempenho ótimo, que são apresentados no próximo capítulo.

#### 4.4 - RESTRIÇÕES À PROCURA

Por questões de ordem prática, fomos limitados em alguns parâmetros na busca aos códigos TCM ótimos, restrições essas impostas pelos programas computacionais usados nesta procura.

As limitações são fundamentalmente duas:

1. número máximo de saídas do codificador -  $n_{\max} = 6$
2. número máximo de unidades de memória do codificador -  $M_{\max} = 5$

A limitação no número máximo de saídas do codificador em 6 impôs ao trabalho o número máximo de pontos das constelações estudadas em 64, além de restringir o número de entradas do codificador de 1 a 5.

Uma vez imposta a restrição, a segunda, ou seja,  $M_{\max} = 5$  não afeta o desenvolvimento do trabalho, já que como o objetivo principal é o de avaliar o desempenho dos códigos de memória unitária, para  $k=5$  (no máximo) necessitamos exatamente de 5 unidades de memória.

A extensão da busca para codificadores com  $n > 6$  e  $M > 5$  envolve necessariamente alterações na estrutura dos programas propostos em [12], o que foge ao escopo deste trabalho.





## CAPÍTULO 5

### APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS



## 5.1 - INTRODUÇÃO

Apresentamos neste capítulo os resultados do processo de procura dos códigos ótimos TCM, com intuito de se traçar comparações entre diferentes constelações dentro dos esquemas combinados (modulação/codificação) e investigar o desempenho dos códigos TCM de memória unitária, conforme descrito no Capítulo 4.

Seguem as tabelas contendo os códigos ótimos TCM para cada classe simples  $\{G_{kn}; m_1, \dots, m_k\}$  associadas às diversas constelações. Cada tabela especifica a classe simples  $\{G_{kn}; m_1, \dots, m_k\}$ , o codificador ótimo, em termos de seus vetores coeficientes  $[VC(1), VC(2), \dots, VC(n)]$ , em números decimais, o  $\Delta_{\min}$  (equação 3.2) deste código e o ganho de codificação assintótico (GCA) em dB calculado em relação a uma constelação de  $2^k$  pontos ('k' = número de entradas do codificador) não codificada e especificada em cada caso.

Finalmente, fazemos uma análise comparativa entre esses resultados, no sentido de esclarecer as questões levantadas neste trabalho.

TABELA 5.1

	constelação	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=5	64-QAM	$\{G_{16} : 5\}$	16, 8, 4, 2, 1, 32	2.453	1.77
	64-CROSS-1	$\{G_{16} : 5\}$	16, 8, 4, 2, 1, 32	1.930	-0.31
	64-CROSS 2	$\{G_{16} : 5\}$	16, 8, 4, 2, 1, 32	1.861	-0.63
	64-PSK	$\{G_{16} : 5\}$	16, 8, 4, 2, 1, 32	2.605	2.29

Resultados da procura dos códigos ótimos TCM para as constelações indicadas com GCA obtido em relação à constelação de 2-PSK não codificadas.

TABELA 5.2

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=4	$\{G_{26} : 2, 2\}$	8, 4, 2, 1, 16, 32	1.416	0.01
	$\{G_{26} : 3, 1\}$	16, 8, 2, 1, 4, 32	1.482	0.41
	$\{G_{26} : 4, 0\}$	32, 8, 4, 2, 1, 16	1.720	1.70
CTM=5	$\{G_{26} : 3, 2\}$	32, 8, 4, 2, 16, 65	1.606	1.11
	$\{G_{26} : 4, 1\}$	32, 16, 8, 2, 4, 64	1.720	1.70
	$\{G_{26} : 5, 0\}$	64, 16, 8, 4, 2, 33	1.748 <sup>*</sup>	1.84

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-QAM com GCA obtido em relação a constelação 4-PSK não codificada.



TABELA 5.3

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=4	$\{G_{26} : 2, 2\}$	8, 4, 2, 1, 16, 32	1.114	-2.07
	$\{G_{26} : 3, 1\}$	16, 8, 2, 1, 4, 32	1.197	-1.45
	$\{G_{26} : 4, 0\}$	32, 8, 4, 2, 1, 16	1.236	-1.17
CTM=5	$\{G_{26} : 3, 2\}$	32, 8, 4, 2, 16, 65	1.347	-0.42
	$\{G_{26} : 4, 1\}$	32, 16, 8, 2, 4, 64	1.482	0.41
	$\{G_{26} : 5, 0\}$	64, 16, 8, 4, 2, 33	1.236 <sup>*</sup>	-1.17

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-CROSS(1) com CGA obtido em relação a constelação 4-PSK não-codificada.

TABELA 5.4

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=4	$\{G_{26} : 2, 2\}$	8, 4, 2, 1, 16, 32	1.075	- 2.38
	$\{G_{26} : 3, 1\}$	16, 8, 2, 1, 4, 32	1.154	-1.76
	$\{G_{26} : 4, 0\}$	32, 8, 4, 2, 1, 16	1.192 <sup>*</sup>	-1.48
CTM=5	$\{G_{26} : 3, 2\}$	32, 8, 4, 2, 16, 65	1.299	-0.74
	$\{G_{26} : 4, 1\}$	32, 16, 8, 2, 4, 64	1.492	0.09
	$\{G_{26} : 5, 0\}$	64, 16, 8, 4, 2, 33	1.192 <sup>*</sup>	-1.48

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos TCM para a constelação 64-CROSS(2), com GCA obtido em relação à constelação 4-PSK não codificada.

TABELA 5.5

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=4	$\{G_{26} : 2, 2\}$	8, 4, 2, 1, 16, 32	1.470	0.34
	$\{G_{26} : 3, 1\}$	16, 8, 2, 1, 4, 32	1.483	0.41
	$\{G_{26} : 4, 0\}$	32, 8, 4, 2, 1, 16	1.669	1.44
CTM=5	$\{G_{26} : 3, 2\}$	32, 8, 4, 2, 16, 65	1.623	1.20
	$\{G_{26} : 4, 1\}$	32, 16, 8, 2, 4, 64	1.669	1.44
	$\{G_{26} : 5, 0\}$	64, 16, 8, 4, 2, 33	2.000*	3.01

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-PSK com GCA obtido em relação à constelação 4-PSK não codificada.

TABELA 5.6

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=1	$\{G_{12} : 1\}$	2, 1	2.449	1.76

Resultado da procura dos códigos TCM ótimos para a constelação

4-PSK(4-QAM) com GCA obtido em relação à constelação de 2-PSK não codificados

TABELA 5.7

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=2	$(G_{12}; 2)$	2, 5	3.162	3.98

Resultado da procura dos códigos TCM ótimos para a constelação 4-PSK(4-QAM) com GCA obtido em relação à constelação de 2-PSK não codificada.

TABELA 5.8

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=3	$(G_{12}; 3)$	4, 11	3.464	4.77

Resultado da procura dos códigos TCM ótimos para a constelação 4-PSK(4-QAM) com GCA obtido em relação à constelação de 2-PSK não codificada.

TABELA 5.9

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=2	$\{G_{24} : 1, 1\}$	8, 4, 2, 1	1.415	0.00
	$\{G_{24} : 2, 0\}$	8, 4, 2, 1	1.675	1.47
CTM=3	$\{G_{24} : 2, 1\}$	16, 8, 6, 1	1.675	1.47
	$\{G_{24} : 3, 0\}$	16, 4, 2, 9	1.790*	2.05

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 16-QAM com CGA obtido em relação à constelação 4-PSK não codificada.

TABELA 5.10

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=2	$\{G_{24} : 1, 1\}$	8, 4, 2, 1	1.467	0.32
	$\{G_{24} : 2, 0\}$	8, 4, 2, 1	1.654	1.36
CTM=3	$\{G_{24} : 2, 1\}$	16, 8, 6, 1	1.654	1.36
	$\{G_{24} : 3, 0\}$	16, 4, 2, 9	2.000*	3.01

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 16-PSK com GCA obtido em relação à constelação 4-PSK não codificada.

TABELA 5.11

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA (dB)
CTM=3	$(G_{3\sigma} : 1, 1, 1)$	4, 2, 1, 8, 16, 32	0.927	1.67
	$(G_{3\sigma} : 2, 1, 0)$	32, 8, 2, 1, 4, 16	1.025	2.54
	$(G_{3\sigma} : 3, 0, 0)$	32, 16, 8, 4, 2, 1	1.197	3.89

Resultados da procura dos códigos ótimos TCM para a constelação 64-QAM com GCA obtido em relação à constelação 8-PSK não codificada.

TABELA 5.12

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA (dB)
CTM=3	$(G_{3\sigma} : 1, 1, 1)$	4, 2, 1, 8, 16, 32	0.927	1.67
	$(G_{3\sigma} : 2, 1, 0)$	32, 8, 2, 1, 4, 16	1.025	2.54
	$(G_{3\sigma} : 3, 0, 0)$	32, 16, 8, 4, 2, 1	1.874 <sup>*</sup>	1.16

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-CROSS(1) com GCA obtido em relação à constelação 8-PSK não codificada.



TABELA 5.13

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=3	$(G_{36} : 1, 1, 1)$	4, 2, 1, 8, 16, 32	0.894	1.35
	$(G_{36} : 2, 1, 0)$	32, 8, 2, 1, 4, 16	0.988	2.22
	$(G_{36} : 3, 0, 0)$	32, 16, 8, 4, 2, 1	0.843	0.84

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-CROSS(2) com GCA obtido em relação à constelação 8-PSK não codificada.

TABELA 5.14

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=3	$(G_{36} : 1, 1, 1)$	4, 2, 1, 8, 16, 32	0.771	0.07
	$(G_{36} : 2, 1, 0)$	32, 8, 2, 1, 4, 16	0.796	0.35
	$(G_{36} : 3, 0, 0)$	32, 16, 8, 4, 3, 1	0.886	1.27

Resultados da procura dos códigos ótimos TCM para a constelação 64-PSK com GCA obtido em relação à constelação 8-PSK não codificada.

TABELA 5.15

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=2	$\{G_{29} : 1, 1\}$	12, 3, 1	1.413	0.00
	$\{G_{29} : 2, 0\}$	8, 2, 5	1.632*	1.25

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 8-QAM com GCA obtido em relação à constelação 4-PSK não codificada.

TABELA 5.16

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=2	$\{G_{29} : 1, 1\}$	12, 3, 1	1.608	1.12
	$\{G_{29} : 2, 0\}$	8, 2, 5	2.000*	3.01

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 8-PSK com GCA obtido em relação à constelação 4-PSK não codificada.

TABELA 5.17

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=4	$\{G_{40} : 1, 1, 1, 1\}$	16, 32, 8, 4, 66, 129	0.691	0.76
	$\{G_{40} : 2, 1, 1, 0\}$	128, 32, 4, 8, 18, 67	0.818	2.23
	$\{G_{40} : 2, 2, 0, 0\}$	128, 64, 8, 4, 18, 33	0.927	3.31
	$\{G_{40} : 3, 1, 0, 0\}$	128, 64, 8, 4, 18, 33	0.927	3.31
	$\{G_{40} : 4, 0, 0, 0\}$	128, 64, 32, 4, 10, 17	0.874 <sup>*</sup>	2.80

\*valor limitado pelas transições

Resultados da procura dos códigos ótimos TCM para a constelação 64-QAM com GCA obtido em relação à constelação 16-QAM não codificada.

TABELA 5.18

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=4	$\{G_{40} : 1, 1, 1, 1\}$	16, 32, 8, 4, 66, 129	0.691	0.76
	$\{G_{40} : 2, 1, 1, 0\}$	128, 32, 4, 8, 18, 67	0.818	2.23
	$\{G_{40} : 2, 2, 0, 0\}$	128, 64, 8, 4, 18, 33	0.874 <sup>*</sup>	2.80
	$\{G_{40} : 3, 1, 0, 0\}$	128, 64, 8, 4, 18, 33	0.874 <sup>*</sup>	2.80
	$\{G_{40} : 4, 0, 0, 0\}$	128, 64, 32, 4, 10, 17	0.874 <sup>*</sup>	2.80

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-CROSS(1) com GCA obtido em relação à constelação 16-QAM não codificada

TABELA 5.19

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=4	{G <sub>40</sub> : 1, 1, 1, 1}	16, 32, 8, 4, 66, 12	0.666	0.44
	{G <sub>40</sub> : 2, 1, 1, 0}	128, 32, 4, 8, 18, 67	0.788	1.90
	{G <sub>40</sub> : 2, 2, 0, 0}	128, 64, 8, 4, 18, 33	0.843*	2.49
	{G <sub>40</sub> : 3, 1, 0, 0}	128, 64, 8, 4, 18, 33	0.843*	2.49
	{G <sub>40</sub> : 4, 0, 0, 0}	128, 64, 32, 4, 10, 17	0.843*	2.49

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-CROSS(2) com GCA obtido em relação à constelação 16-QAM não codificada

TABELA 5.20

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=4	{G <sub>40</sub> : 1, 1, 1, 1}	16, 32, 8, 4, 66, 129	0.402	-3.94
	{G <sub>40</sub> : 2, 1, 1, 0}	128, 32, 4, 8, 18, 67	0.447	-3.02
	{G <sub>40</sub> : 2, 2, 0, 0}	128, 64, 8, 4, 18, 33	0.560	-1.06
	{G <sub>40</sub> : 3, 1, 0, 0}	128, 64, 8, 4, 18, 33	0.593	-0.57
	{G <sub>40</sub> : 4, 0, 0, 0}	128, 64, 32, 4, 10, 17	0.625	-0.11

Resultados da procura dos códigos ótimos TCM para a constelação 64-PSK, com GCA obtido em relação à constelação 16-QAM não codificada.

TABELA 5.21

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=3	$\{G_{34} : 1, 1, 1\}$	4, 8, 18, 35	1.096	3.12
	$\{G_{34} : 2, 1, 0\}$	32, 8, 6, 17	1.415	5.34
	$\{G_{34} : 3, 0, 0\}$	32, 16, 4, 11	1.266*	4.38

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 16-QAM com GCA obtido em relação à constelação 8-PSK não codificada.

TABELA 5.22

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=3	$\{G_{34} : 1, 1, 1\}$	4, 8, 18, 35	1.859	1.01
	$\{G_{34} : 2, 1, 0\}$	32, 8, 6, 17	1.150	3.54
	$\{G_{34} : 3, 0, 0\}$	32, 16, 4, 11	1.214	4.01

Resultados da procura dos códigos ótimos TCM para a constelação 16-PSK com GCA obtido em relação à constelação 8-PSK não codificada.

TABELA 5.23

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=3	$(G_{35} : 1, 1, 1)$	16, 8, 4, 2, 33	1.000	2.37
	$(G_{35} : 3, 0, 0)$	32, 16, 4, 2, 9	1.264*	4.36

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 32-CROSS com CGA obtido em relação à constelação 8-PSK não codificada.

TABELA 5.24

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=3	$(G_{35} : 1, 1, 1)$	16, 8, 4, 2, 33	0.790	0.28
	$(G_{35} : 3, 0, 0)$	32, 16, 4, 2, 9	1.167	3.67

Resultados da procura dos códigos ótimos TCM para a constelação 32-PSK com GCA obtido em relação à constelação 8-PSK não codificada.

TABELA 5.25

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=4	{G <sub>45</sub> : 1, 1, 1, 1}	8, 16, 36, 66, 131	0.774	1.75
	{G <sub>45</sub> : 2, 1, 1, 0}	128, 32, 12, 18, 67	0.999	3.96
	{G <sub>45</sub> : 2, 2, 0, 0}	128, 64, 12, 22, 33	1.095	4.76
	{G <sub>45</sub> : 3, 1, 0, 0}	128, 64, 24, 14, 33	1.095	4.76
	{G <sub>45</sub> : 4, 0, 0, 0}	128, 64, 32, 4, 19	0.894*	2.30

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 32 CROSS com GCA obtido em relação à constelação 16-QAM não codificada.

TABELA 5.26

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=4	{G <sub>45</sub> : 1, 1, 1, 1}	8, 16, 36, 66, 131	0.436	-3.24
	{G <sub>45</sub> : 2, 1, 1, 0}	128, 32, 12, 18, 67	0.585	-0.68
	{G <sub>45</sub> : 2, 2, 0, 0}	128, 64, 12, 22, 33	1.617	-0.22
	{G <sub>45</sub> : 3, 1, 0, 0}	128, 64, 24, 14, 33	1.617	-0.22
	{G <sub>45</sub> : 4, 0, 0, 0}	128, 64, 32, 4, 19	0.648	0.20

Resultados da procura dos códigos ótimos TCM para a constelação 32-PSK com GCA obtido em relação à constelação 16-QAM não codificada.



TABELA 5.27

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=5	$\{G_{56} : 1, 1, 1, 1, 1\}$	32, 16, 72, 132, 262, 515	0.535	1.56
	$\{G_{56} : 2, 1, 1, 1, 0\}$	512, 128, 24, 36, 66, 259	0.691	3.78
	$\{G_{56} : 2, 2, 1, 0, 0\}$	512, 256, 32, 28, 66, 129	0.691	3.78
	$\{G_{56} : 3, 2, 0, 0, 0\}$	512, 256, 128, 32, 22, 65	0.757	4.57
	$\{G_{56} : 4, 1, 0, 0, 0\}$	512, 256, 128, 56, 6, 65	0.757	4.57
	$\{G_{56} : 5, 0, 0, 0, 0\}$	512, 256, 128, 64, 54, 11	0.618*	2.81

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-QAM com GCA obtido em relação à constelação 32-CROSS não codificada.

TABELA 5.28

	classe simples	VC(1), ..., VC(n)	$\Delta_{min}$	GCA(dB)
CTM=5	$\{G_{56} : 1, 1, 1, 1, 1\}$	32, 16, 72, 132, 262, 515	0.535	1.56
	$\{G_{56} : 2, 1, 1, 1, 0\}$	512, 128, 24, 36, 66, 259	0.691	3.78
	$\{G_{56} : 2, 2, 1, 0, 0\}$	512, 256, 32, 28, 66, 129	0.691	3.78
	$\{G_{56} : 3, 2, 0, 0, 0\}$	512, 256, 128, 32, 22, 65	0.757	4.57
	$\{G_{56} : 4, 1, 0, 0, 0\}$	512, 256, 128, 56, 6, 65	0.757	4.57
	$\{G_{56} : 5, 0, 0, 0, 0\}$	512, 256, 128, 64, 54, 11	0.618*	2.81

\*valor limitado pelas transições paralelas



Resultados da procura dos códigos ótimos TCM para a constelação 64-CROSS(1) com GCA obtido em relação à constelação 32-CROSS não codificada.

TABELA 5.29

	classe simples	VC(1), ..., VC(n)	$\Delta_{\min}$	GCA(dB)
CTM=5	$(G_{56} : 1, 1, 1, 1, 1)$	32, 16, 72, 132, 282, 515	0.515	1.23
	$(G_{56} : 2, 1, 1, 1, 0)$	512, 128, 24, 36, 66, 259	0.666	3.46
	$(G_{56} : 2, 2, 1, 0, 0)$	512, 256, 32, 28, 66, 129	0.666	3.46
	$(G_{56} : 3, 2, 0, 0, 0)$	512, 256, 128, 32, 22, 65	0.729	4.25
	$(G_{56} : 4, 1, 0, 0, 0)$	512, 256, 128, 56, 6, 65	0.729	4.25
	$(G_{56} : 5, 0, 0, 0, 0)$	512, 256, 128, 64, 54, 11	0.596*	2.50

\*valor limitado pelas transições paralelas

Resultados da procura dos códigos ótimos TCM para a constelação 64-CROSS(2) com GCA obtido em relação à constelação 32-CROSS não codificada.

## 5.2 - ANÁLISE COMPARATIVA

A Tabela 5.1 é por si só conclusiva. Neste caso, vemos que a modulação 64-PSK com o código  $(G_{16}; 5)$  é a que apresenta o maior ganho assintótico. A justificativa é que apesar do relativamente grande número de pontos da constelação, a modulação PSK apresentou melhor desempenho. Isso pelo fato de a taxa ser pequena, e conseqüentemente grande a redundância, tem-se um pequeno número de bits/símbolo, fazendo com que para as constelações 64-QAM, 64-CROSS(1) e 64-CROSS(2) se acumule um maior número de pontos-centro do círculo de energia unitária, enquanto que para a constelação 64-PSK todos os pontos se encontram no limite deste círculo.

As Tabelas 5.2 - 5.5 apresentam resultados para as classes  $G_{26}$  com CTM = 4 e 5, levando-se em consideração as constelações 64-QAM, 64-CROSS(1), 64-CROSS(2) e 64-PSK quando comparadas com a constelação 4-PSK, respectivamente. Observa-se que a constelação 64-PSK com o código  $(G_{26}; 5,0)$  apresenta o maior GCA dentre todas as constelações e classes simples de códigos com CTM = 5, seguindo pela constelação 64-QAM com a classe  $(G_{26}, 5,0)$ . Por outro lado, quando o CTM é 4 a constelação 64-QAM com  $(G_{26}; 4,0)$  apresenta o maior GCA dentro de todas as classes e constelações seguida de perto pela constelação 64-PSK e classe  $(G_{26}; 4,0)$ .

Para CTMs menores que 4, a constelações 64-QAM apresenta GCAs superiores aos da constelação 64-PSK; uma vez que as distâncias Euclidianas entre os pontos nos subconjuntos terminais são maiores que aquelas apresentadas pela constelação 64-PSK.

Acontece, todavia, que estas distâncias crescem mais rapidamente do que aquelas associadas às transições paralelas.

Conseqüentemente, estas últimas começarão a limitar o desempenho do sistema combinado. Isto se verifica já para o  $CTM=5$  tanto para o 64-QAM como para o 64-PSK. Entretanto, como as distâncias Euclidianas entre as transições paralelas no 64-PSK são maiores que as do 64-QAM temos portanto um GCA maior para a constelação 64-PSK quando comparada com a constelação 64-QAM.

As Tabelas 5.6, 5.7 e 5.8 por si só relacionam os GCAs da constelação 4-PSK ou 4-QAM quando comparados com a constelação 2-PSK. Aqui mostra-se o crescimento do GCA obtido, pela classe  $\{G_{12}, 2\}$  comparada com a classe  $\{G_{12}, 1\}$  que é de 2.22 dBs e do GCA obtido pela classe  $\{G_{12}, 3\}$  comparada com a classe  $\{G_{12}, 2\}$  que é de 0.79 dBs.

Note que os GCAs das Tabelas 5.6, 5.7 e 5.8 são bastante altos, visto que a não existência de transições paralelas nas correspondentes treliças implica em distâncias Euclidianas infinitas, logo não oferecendo limitações quanto ao  $\Delta_{\min}$  máximo.

Agora, considere a classe  $\{G_{k, 2k}, M\}$  com  $M \geq 1$  e  $k \geq 1$ . Das Tabelas 5.9 a 5.14 podemos inferir que conjetura partir de uma dada multiplicidade  $k=k_0$  e  $M$  fixo existirá um código  $\{G_{k_0, 2k_0}, M\}$  tal que o GCA correspondente será igual ou maior que aquele obtido pelo código  $\{G_{12}, M\}$ .

Das Tabelas 5.9 e 5.10 constata-se o fato de que para um  $CTM \leq 2$ , as constelações 16-QAM apresentam um GCA maior do que aquele obtido através das constelações 16-PSK. Porém, para  $CTM \geq 3$ , o reverso é verificado. As justificativas para tal são as mesmas do caso observado nas Tabelas 5.2 a 5.5.

Devido a uma limitação restritiva do programa desenvolvido em [12], não foi possível realizarmos um estudo comparativo levando em conta o fator de multiplicidade de taxa para um  $m$  fixo. Contudo,

através das Tabelas 5.9 e 5.10, bem como através das Tabelas 5.11 a 5.14, via processo de inferências, podemos dizer que a conjetura acima se aplica.

Destas mesmas Tabelas, um fato um tanto quanto surpreendente revela que os códigos de memória-unitária completos apresentam um baixo valor de GCA quando comparado com os demais códigos pertencentes às respectivas classes. Isto nos leva a concluir que as constelações ótimas conhecidas não são apropriadas ou não estão casadas com o codificador com estrutura de memória-unitária. Como uma confirmação desta observação, note que nas Tabelas 5.12 e 5.13 os códigos  $\{G_{36}; 1, 1, 1\}$  para as constelações 64-CROSS (1) e (2) apresentam GCAs maiores que os códigos  $\{G_{36}; 3, 0, 0\}$ .

Das Tabelas 5.15 e 5.16 podemos ver que os códigos de memória-unitária completa estão aquém dos códigos  $\{G_{23}; 2, 0\}$ . Novamente, a constelação 8PSK apresenta um GCA maior do que aquele obtido pela constelação 8-QAM.

Com as Tabelas 5.17 a 5.20 mostramos que a constelação 64-QAM com o código  $\{G_{46}; 2, 2, 0, 0\}$  apresenta maior GCA dentre as demais constelações e classes simples. O código  $\{G_{46}; 2, 2, 0, 0\}$  apresenta internamente um código convolucional de memória-duonária de taxa  $R=2/3$  preservando as mesmas características que são apresentadas por um código de memória unitária. Assim, temos um primeiro exemplo onde esta classe de códigos atinge um desempenho maior que os códigos convolucionais tradicionais. Estas afirmações valem para o caso mostrado na Tabela 5.25.

Note que este código é ótimo para as constelações 64-QAM, 64-CROSS(1) e 64-CROSS(2). Porém, o código  $\{G_{46}; 4, 0, 0, 0\}$  apresenta o melhor GCA para a constelação 64-PSK.

As Tabelas 5.21 e 5.22 mostram resultados para as classes  $G_{34}$  com CTM=3 associadas às constelações 16-QAM e 16-PSK, em que observamos o melhor desempenho obtido para a constelação 16-QAM com o código  $\{G_{34}; 2,1,0\}$ . Cabe salientar que o mesmo sendo limitado pelas transições paralelas, o desempenho da classe  $\{G_{34}; 3,0,0\}$  com a constelação 16-QAM, é superior ao desempenho dessa mesma classe com a constelação 16-PSK.


Essa vantagem no desempenho da constelação QAM em relação à PSK pode ser observada também nas Tabelas 5.23 e 5.24, que apresentam os resultados das classes  $G_{35}$  com CTM=3.

Nas Tabelas 5.25 e 5.26 os melhores códigos para as classes  $G_{45}$  com CTM=4, em que o melhor desempenho se dá para a classe  $\{G_{45}; 3,1,0,0\}$  com a constelação 32-CROSS. Neste caso, observamos que o melhor desempenho para a constelação 32-PSK (classe  $\{G_{45}; 4,0,0,0\}$ ) é inferior ao pior desempenho para a constelação 32-CROSS (classe  $\{G_{45}; 1,1,1,1\}$ ).

Finalmente, as Tabelas 5.27 a 5.29 apresentam resultados para as classes  $G_{56}$  com CTM=5, com as constelações 64-QAM, 64-CROSS(1) e 64-CROSS(2), onde podemos observar que para qualquer uma dessas constelações a classe que apresenta melhor desempenho é a  $\{G_{56}; 4,1,0,0,0\}$  equivalente à classe  $\{G_{56}; 3,2,0,0,0\}$ , sendo pois o maior desempenho para a constelação 64-QAM.

Traçaremos agora comparações entre diferentes constelações analisando o desempenho de cada arranjo dentro do esquema combinado (modulação/codificação).

De uma maneira geral observamos que para as classes simples  $\{G_{kn}; m_1, m_2, \dots, m_k\}$ , o que nos leva a constelações com mesmo número de pontos, as modulações do tipo QAM (quadradas para 'n' par e CROSS



para 'n' ímpar) apresentam desempenhos superiores ao apresentado pela constelação PSK.

Porém, isso não se verificou para as taxas 2/3 e 2/4 (Tabelas 5.15 e 5.16; 5.8 e 5.9 respectivamente) onde a modulação do tipo PSK teve desempenho superior às demais. Analisando esses resultados, nós chegamos à conclusão que a modulação PSK apresenta melhor desempenho para as constelações com menor número de pontos, exatamente por estarem os pontos mais afastados entre si, em relação às demais constelações com mesmo número de pontos, (maior  $\Delta_o$ ): à medida em que se for aumentando o número de pontos numa constelação do tipo PSK, estes irão ficando cada vez mais próximos entre si (diminui  $\Delta_o$ , uma vez que todas as constelações estudadas estão restritas à condição de energia média unitária, portanto no caso da modulação PSK todos os pontos estão sobre a circunferência de raio unitário. Podemos verificar facilmente essa afirmação comparando os resultados das Tabelas 5.19 e 5.20, onde vemos que até a constelação 64-CROSS(2), que é um arranjo de desempenho bem inferior ao do 64-QAM, apresenta melhor desempenho que a constelação 64-PSK.

Para a modulação do tipo QAM, esta característica também é verificada, ou seja, com aumento do número de pontos na constelação, temos uma diminuição do desempenho; porém esta é mais sensível para a modulação PSK.

Fazendo-se agora uma análise mais específica, observamos que para o mesmo CTM (comprimentos total de memória) e para a mesma modulação, o desempenho do código TCM diminui à medida que a taxa também diminui. Esta afirmação é comprovada comparando-se os resultados das Tabelas 5.21, 5.23 e 5.11, onde temos CTM=3 e

respectivamente: taxa 3/4 e constelação 16-QAM, taxa 3/5 e constelação 32-CROSS e taxa 3/6 e constelação 64-QAM. Essa característica também se verifica para a modulação PSK (Tabela 5.22 - taxa 3/4, 16-PSK; Tabela 5.24 - taxa 3/5, 32-PSK; Tabela 5.14 - taxa 3/6, 64-PSK).

Uma característica importante é a de que para determinadas condições, algumas constelações apresentam pequena vantagem em relação a outras de mesmo número de pontos. Isto acontece por exemplo para CTM=4 em que a constelação 64-QAM (Tabela 5.17) apresenta pequenas vantagens em relação à constelação 64-CROSS(1) (Tabela 5.18), e ainda para CTM=5, onde não existe diferença entre os desempenhos das constelações 64-QAM (Tabela 5.27) e 64-CROSS(1) (Tabela 5.28).

Traçamos agora comparações entre diferentes estruturas de memória do codificador TCM, evidenciando o desempenho dos códigos TCM de memória unitária.

Para um codificador convolucional binário com 'k' entradas, 'n' saídas e CTM=M, existe sempre um equivalente com estrutura de memória unitária, com k'=Mk entradas e n'=Mn saídas [10].

O objetivo principal desse trabalho foi o de avaliar se esta característica é válida para os codificadores TCM, e comparando-se os resultados da Tabela 5.7, onde tem-se a classe simples  $\{G_{12}; 2\}$  associada à constelação 4-PSK, com os resultados das Tabelas 5.7 e 5.8, onde temos os codificadores de memória unitária  $\{G_{24}; 1, 1\}$  associados às constelações 16-QAM e 16-PSK respectivamente, observamos que tal característica não é verificada para os codificadores TCM, pois o desempenho para a classe  $\{G_{12}; 2\}$  é muito superior ao da classe  $\{G_{24}; 1, 1\}$  qualquer que seja a constelação associada a esta última.

Essa conclusão é bastante lógica, pois ao passarmos da classe  $\{G_{12}; 2\}$  para a classe  $\{G_{24}; 1, 1\}$  estamos passando de uma

constelação de 4 pontos para uma constelação de 16 pontos, e como essas constelações estão restritas à condição de energia média unitária, a distância mínima ( $\Delta_0$ ) entre dois pontos na constelação de 16 pontos será bem menor que esta distância para a constelação de 4 pontos, o que acarreta a diminuição do desempenho.

Mesmo para cada classe simples  $\{G_{kn}; M\}$  e para a mesma constelação, os codificadores de memória unitária apresentam o pior desempenho em comparação com as demais estruturas de memória. Observamos, comparando os resultados de uma mesma tabela (qualquer), que à medida que se concentram as unidades de memória num menor número de linhas de entrada do codificador, o desempenho do código tende a melhorar, e só deixa de melhorar quando este passa a ser limitado pelas transições paralelas, comprovamos essa característica pelos resultados da Tabela 5.12, onde para a constelação 64-CROSS(1) temos a saturação para a classe  $G_{36}; 3, 0, 0$ . Porém, tal saturação, não acontece para a mesma classe associada à constelação 64-PSK (Tabela 5.14), isso pelo fato de que para a constelação 64-PSK não ocorre repetição entre sucessivas distâncias mínimas entre os pontos dos subconjuntos no "set partitioning", repetição esta que se dá para a constelação 64-CROSS(1).

Observamos, também, que algumas estruturas de memória do codificador não levam vantagens sobre outras. Isso acontece quando temos o mesmo número de unidades de memória concentradas no mesmo número de linhas de entrada do codificador, não importando sua distribuição. Por exemplo: na Tabela 5.17 temos o mesmo desempenho para as classes  $\{G_{46}; 2, 2, 0, 0\}$  e  $\{G_{46}; 3, 1, 0, 0\}$ .

Outra questão que se colocava como objetivo deste trabalho era a da avaliação do desempenho dos códigos TCM de memória



unitária quando um fator de multiplicidade de taxas é considerado. Então para a mesma taxa comparamos os desempenhos dos codificadores da classe  $\{G_{12}; 1\}$  (Tabela 5.6) com a classe  $\{G_{24}; 1, 1\}$  (Tabelas 5.8 e 5.9) e observamos que com o aumento da multiplicidade da taxa ocorre uma diminuição do desempenho, isso certamente pelo fato do aumento do número de pontos na constelação associada, e conseqüente diminuição da distância mínima entre esses pontos ( $\Delta_0$ ).

Se compararmos agora os resultados das classes  $\{G_{24}; 1, 1\}$  (constelação 16-QAM) da tabela 5.8 com  $\{G_{36}; 1, 1, 1\}$  (constelação 64-QAM) da tabela 5.11, ou ainda, das classes  $\{G_{24}; 1, 1\}$  (constelação 16-PSK) da Tabela 5.9, com  $\{G_{36}; 1, 1, 1\}$  (constelação 64-PSK) da Tabela 5.14, observamos uma melhora no desempenho, porém esta é devido ao aumento do CTM de dois para três.

Esta conclusão de que o aumento na multiplicidade da taxa implica numa diminuição do desempenho é confirmada pela comparação entre os resultados da Tabela 5.16 ( $\{G_{23}; 1, 1\}$  - 8-PSK) e Tabela 5.20 ( $\{G_{46}; 1, 1, 1, 1\}$  - 64-PSK).

Como os códigos TCM de memória unitária constituíram o objetivo central do estudo aqui realizado, apresentamos a seguir uma expressão que permite calcular o  $\Delta_{\min}$  para qualquer desses códigos. É de fundamental importância dizer que para esta expressão não foi conseguida demonstração formal e que foi obtida experimentalmente, verificando-se pois para todos os códigos TCM de memória unitária aqui apresentados. Assim fica sendo esta uma conjetura, ou seja, existindo o codificador TCM de memória unitária com 'k' entradas e 'n' saídas, o  $\Delta_{\min}$  para o código a ele associado será dado por:

$$\Delta_{\min}^2 = \Delta_{n-k}^2 + \Delta_0^2$$

(5.1)


onde:

$\Delta_0$  = distância mínima entre os pontos da constelação associada ao codificador

$\Delta_{n-k}$  = distância mínima entre os pontos do subconjunto de ordem 'n-k' no "set partitioning"

Com base na equação 5.1 concluímos que mesmo para códigos com valores relativamente grandes para 'k' e 'n', em que a constelação esteja restrita à condição de energia média unitária, o desempenho não deverá ser satisfatório, exatamente por conter a expressão de seu  $\Delta_{\min}^2$  o termo  $\Delta_0^2$ , que será tanto menor quanto maior for 'n', já que o número de pontos na constelação é  $2^n$ .

O que também se pode verificar é como degrada o desempenho do sistema combinado para taxas diferentes de  $\frac{k}{k+1}$ .



**CONCLUSÕES E SUGESTÕES  
PARA TRABALHOS FUTUROS**



Os Capítulos 1, 2 e 3 contêm o embasamento teórico para o desenvolvimento do estudo que foi realizado, ou seja, uma revisão à respeito dos tópicos em modulação digital (Capítulo 1), codificação convolucional (Capítulo 2) e códigos TCM (Capítulo 3). O desenvolvimento do trabalho em si é explicitado no Capítulo 4, onde técnicas e objetivos são apresentados para a obtenção dos resultados que comparecem no Capítulo 5. Destes resultados conseguimos muitas informações relevantes sobre a forma combinada codificação/modulação, salientando-se as seguintes:

1. Para os códigos TCM, associados a constelações em duas dimensões, a estrutura de memória unitária não apresenta melhor desempenho que as demais estruturas de memória dentro de cada classe simples  $(G_{kn}; M)$ , e assim, não se verifica a característica válida para os códigos convolucionais binários que há sempre um codificador ótimo com estrutura de memória unitária.
2. Dentro de cada classe simples de codificadores  $(G_{kn}; M)$ , o desempenho tende a melhorar à medida que se concentram as unidades de memória num menor número de linhas de entrada do codificador, sendo porém este desempenho limitado, em certos casos, pelas transições paralelas, ou em outras palavras, a melhor configuração do codificador, será a classe  $(G_{kn}; M, D, D, D, \dots)$  se não houver limitações pelas distâncias entre as transições paralelas.
3. Com relação ao desempenho quando um fator de multiplicidade de taxa é considerado, temos a observar que para os casos que


conseguimos avaliar, não há melhora no desempenho com esse fator de multiplicidade. Porém, devido às restrições impostas pelos programas de procura (Seção 4.4) não pudemos estender a busca para multiplicidades maiores que 3, assim não temos uma conclusão precisa à respeito deste tópico.

4. Para um mesmo CTM, observamos uma degradação no desempenho com a diminuição da taxa, por exemplo: para taxas  $3/4$ ,  $3/5$ ,  $3/6$  temos um desempenho cada vez menor para uma mesma constalação em função do aumento do número de pontos desta, dada à restrição de energia média unitária.
5. Também não se observou melhora no desempenho com a geometria da constelação aproximando-se de um círculo, em função do não casamento dessa com o codificador e também pelo processo de partição de conjunto, que em determinados casos leva a uma repetição da distância mínima entre pontos de subconjuntos consecutivos.

#### SUGESTÕES PARA FUTUROS TRABALHOS

Apresentamos algumas sugestões para futuros trabalhos neste assunto, como um aprofundamento deste estudo:

1. Extensão dos programas computacionais de busca elaborados por [12], no sentido de tornar essa procura mais rápida, computacionalmente, e permitir procuras para classes com maiores números de entradas e saídas, e maior número de unidades de memória, permitindo assim a avaliação para códigos com maiores multiplicidades



de taxa.

2. Realização da busca para constelações em maiores dimensões, para se avaliar a possível melhoria no desempenho dos códigos TCM de memória unitária para essas dimensões.
3. Realização da busca para constelações em reticulados diferentes do  $Z^2$ , com o mesmo intuito da proposição 2.
4. Estudo da possibilidade de se demonstrar formalmente a equação 5.1.



## REFERÊNCIAS BIBLIOGRÁFICAS



- [1] Batson, B.; Moorhead, R.W.: Simulation Results for the Viterbi Decoding Algorithm - Technical Report, Manned Space Craft Center, Houston, Texas/ NASA, Washington D.C., 1972.
- [2] Biglieri, E.: High Level Modulation and Coding for Nonlinear Satellite Channels - IEEE Transactions on Communications, vol. COM-32, May 1984, p. 616.
- [3] Borelli, W.C.: Convolutinal Codes for Multilevel Data Transmission - Ph.D Thesis, University of Kent at Canterbury, England - June 1983.
- [4] Campopiano, C.N.; Glazer, B.G.: A Coherent Digital Amplitude and Phase Modulation Scheme - IRE Transaction on Commun. Syst., March 1962, p. 90-95.
- [5] Clark, G.C. Jr.; Cain, J.B.: Error-Correction Coding for Digital Communications - Plenus Press - New York and London.
- [6] Costello, K.J.: Construction of Convolutional Codes for Sequential Decoding; Ph.D Thesis, University of Notre Dame, U.S.A., 1966.
- [7] Forney, G.D. Jr.: Convolutional Codes I, Algebraic Structure - IEEE Trans. on Inf. Theory, vol. IT-16, November 1970, p. 720.
- [8] Forney, G.D. Jr.: The Viterbi Algorithm; IEEE Proc., March 1976, p. 268.





- [9] Forney, G.K. Jr.; Gallanger, R.G.; Lang, G.R.; Longstaff, F.M.; Qureshi, S.U.: Efficient Modulation for Band-Limited Channels, IEEE Journal on Selected Areas in Communications - vol SAC2, September 1984, p. 632-647.
- [10] Lee, Lin-Nan: Short Unit-Memory Byte Oriented Binary Convolutional Codes Having Maximal Free Distance - IEEE Trans. on Information Theory, May 1976, p. 349-352.
- [11] Lin, Shu; Costello, K.J. Jr.: Error control Coding - Fundamentals and Applications - Prentice-Hall, 1983.
- [12] Markman, I.: esquemas de Modulação Codificada em Três e Quatro Dimensões - Tese de Mestrado - UNICAMP, Campinas 1987.
- [13] Markman, I.; Borelli, W.C.: Um Método de Procura de Códigos de Alfabeto redundante - Anais do 5.º SBT - Campinas, Setembro 1987.
- [14] Markman, I.; Borelli, W.C.: Subconjuntos Eficientes de Códigos de Alfabeto Redundante para a Procura Não-Exaustiva de Códigos ótimos - Anais do 6.º SBT - Campina Grande - Setembro 1988.
- [15] Marson, M.A.; Albertengo, G.; Benedetto, S.; Giachin, E.: Theoretical Aspects and Performance Evaluation of Channel Encoding with Redundant Modular Alphabets - Proc. GLOBECOM'84, Atlanta, U.S.A., November 1984, p. 688.



- [16] Palazzo, R. Jr.; Da Rocha, V.C. Jr.: Códigos Corretores de Erros - Monografia ISICT'87, Campinas, SP, Brasil.
- [17] Peterson, W.W.; Weldon, E.S.: Error-Correcting Codes - The M.I.T. Press, Cambridge, Mass, U.S.A., 1972.
- [18] Shanmugan, K.S.: Digital and Analogic Communication Systems, Wiley, New York, 1979.
- [19] Ungerboeck, G.: Channel Coding with Multilevel/Phase Signal, IEEE Trans. on Information Theory, vol. IT-28, January 1982.
- [20] Ungerboeck, K.G.: Trellis-Codes Modulation with Redundant Signal Sets. Part I: Introduction, Part II, State of the Art. IEEE Communications Magazine, vol. 25, February 1987.
- [21] Viterbi, A.J.; Omura, J.K.: Principles of Digital communication and Coding - McGraw-Hill. Kogakusha, LTD, 1979.
- [22] Wozencraft, J.M.; Jacobs, I.M.: Principles of Communications Engineering - Wiley, New York, 1965.
- [23] Massey, J.L.: Coding and Modulation in Digital Communications. Proc. 1974 Int. Zurich Seminar Digital Comm., Zurich, Switzerland, March 1974.



## APENDICE 1

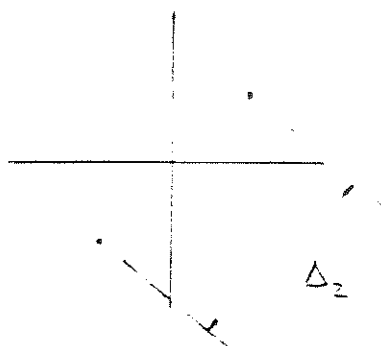
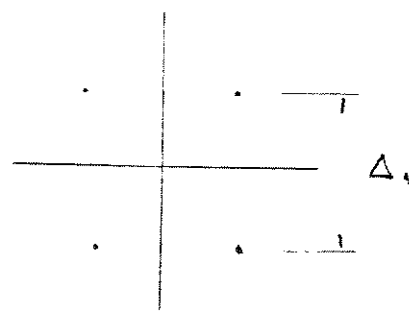
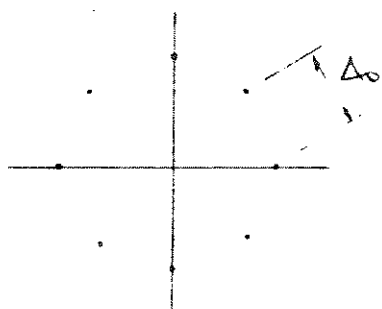
### "SET PARTITIONINGS"



Este apêndice apresenta a partição de conjunto ("set-partitioning") das constelações estudadas neste trabalho.

Devido à simetria dessas constelações, é desenvolvido apenas um "ramo" do set partitioning, mostrando os subconjuntos que apresentam a distância mínima entre os pontos.

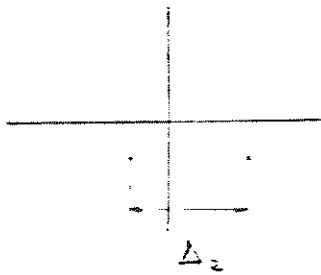
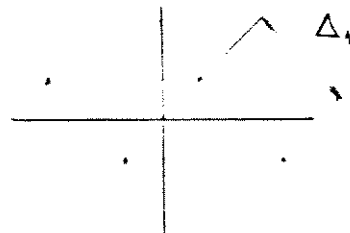
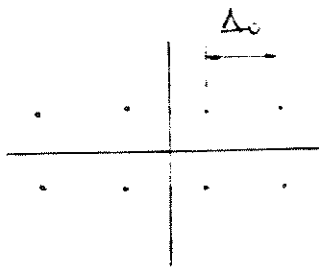
É importante lembrar que todas essas constelações estão restritas à condição de energia média unitária ( $E_m = 1$ ).



$$\Delta_0 = 0.765$$

$$\Delta_1 = 1.414$$

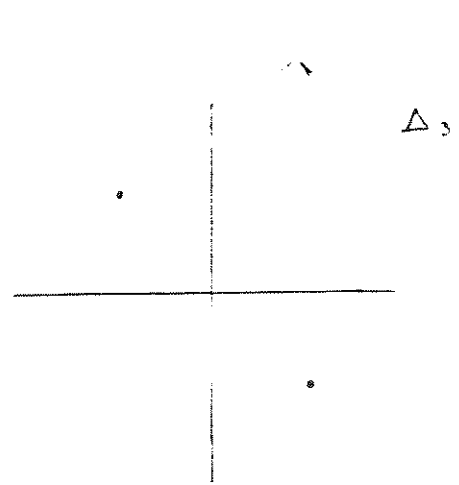
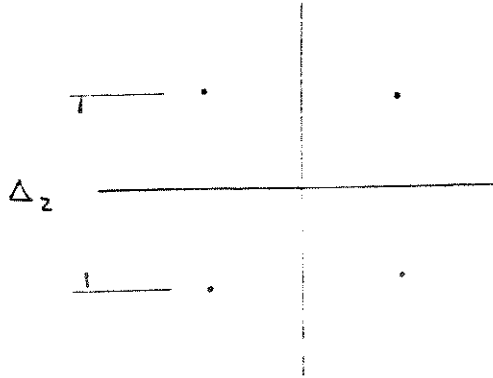
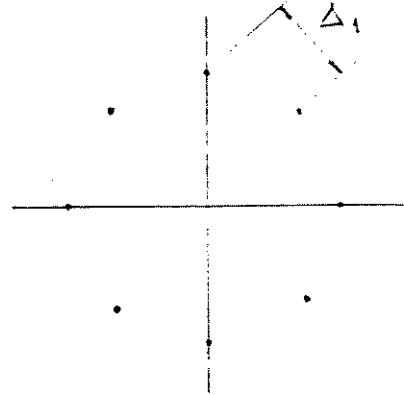
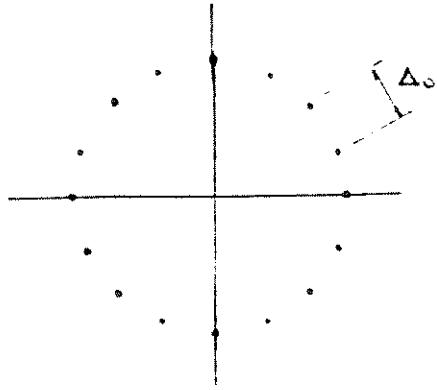
$$\Delta_2 = 2.000$$



$$\Delta_0 = 0.816$$

$$\Delta_1 = 1.154$$

$$\Delta_2 = 1.632$$

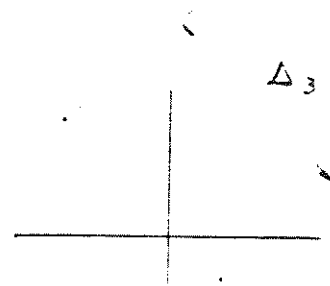
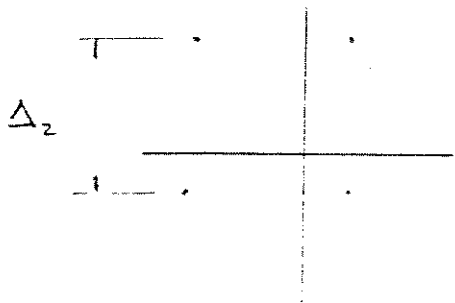
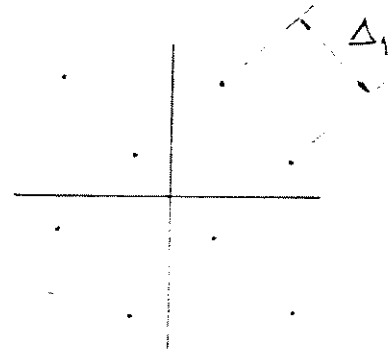
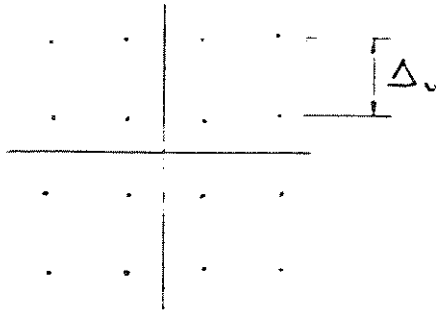


$$\Delta_0 = 0.390$$

$$\Delta_1 = 0.765$$

$$\Delta_2 = 1.414$$

$$\Delta_3 = 2.000$$



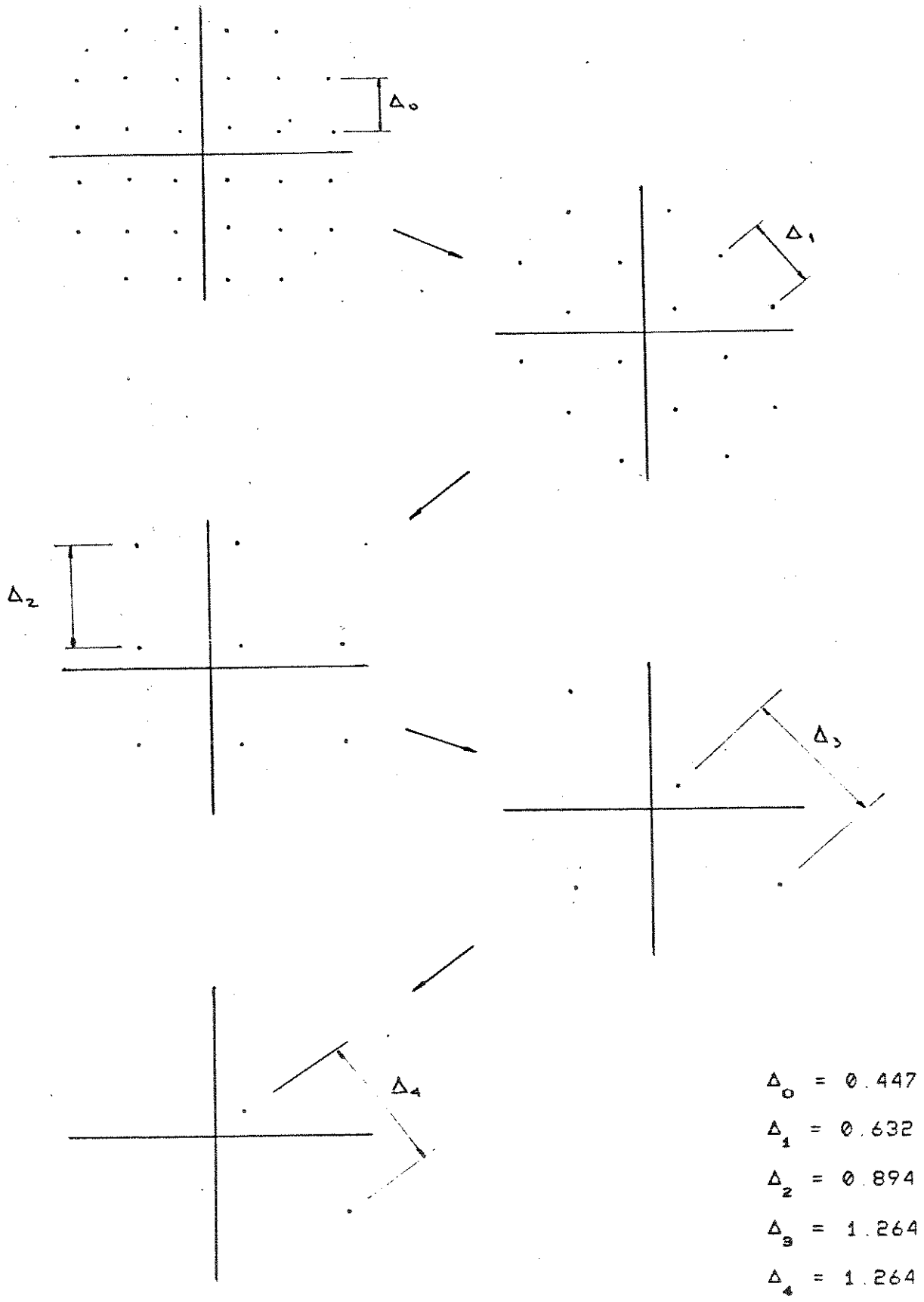
$$\Delta_0 = 0.633$$

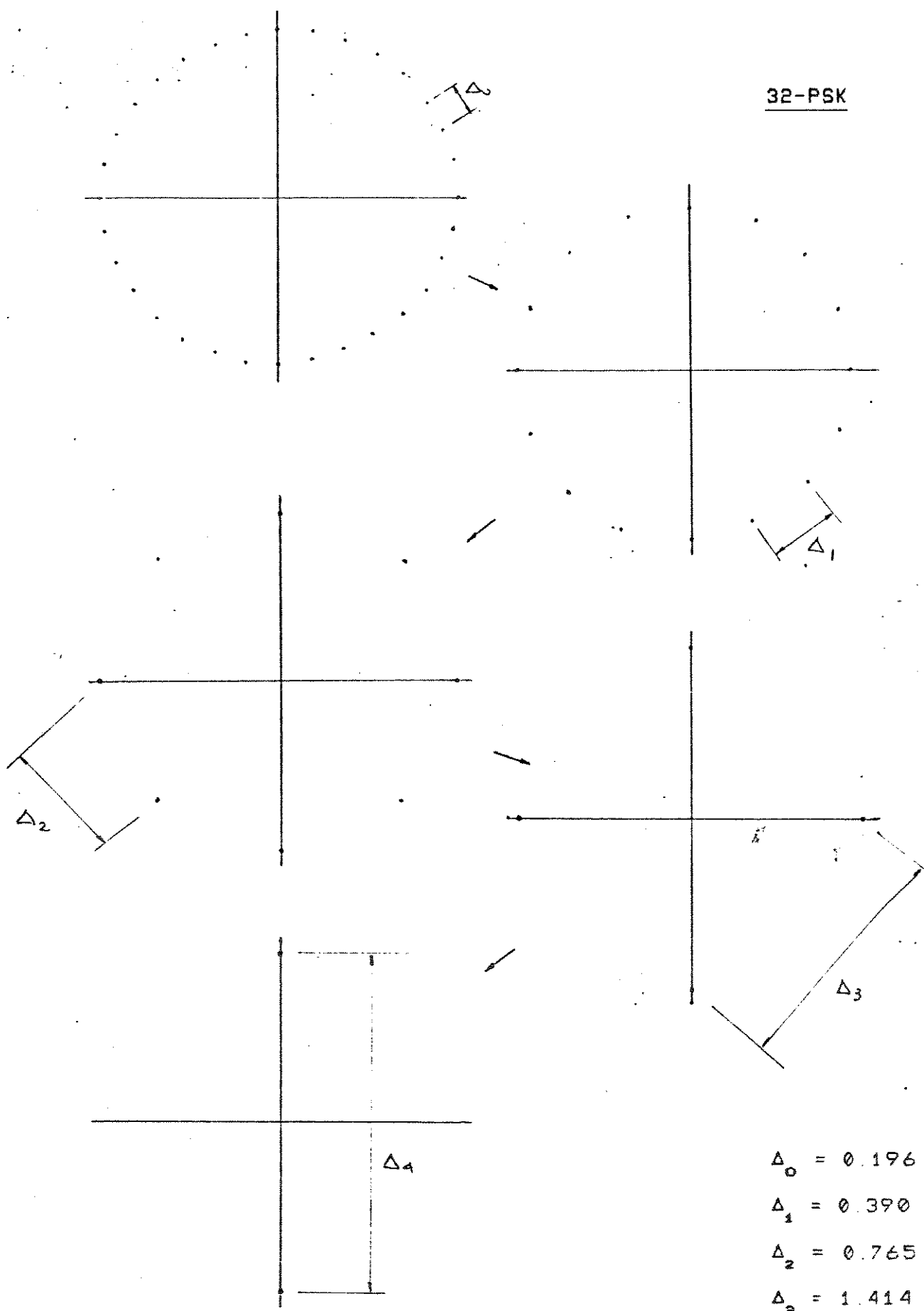
$$\Delta_1 = 0.895$$

$$\Delta_2 = 1.266$$

$$\Delta_3 = 1.790$$







$\Delta_0$	=	0.196
$\Delta_1$	=	0.390
$\Delta_2$	=	0.765
$\Delta_3$	=	1.414
$\Delta_4$	=	2.000

