

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELETRICA  
DEPARTAMENTO DE TELEMATICA

*Este exemplar corresponde  
à redação final de Tese  
deportugheis e aprovada  
pelo Conselho Julgado  
em Março 1987*

*Ambrósio  
Julho 87*

UMA CONTRIBUIÇÃO AO ESTUDO  
DE SISTEMAS DE MODULAÇÃO  
M-PSK CODIFICADA.

JAIME PORTUGHEIS

ORIENTADOR: PROF.DR.WALTER DA CUNHA BORELLI

Tese apresentada à Faculdade de Engenharia  
Elétrica, da Universidade Estadual de  
Campinas - UNICAMP, como parte dos  
requisitos exigidos para a obtenção do  
Título de Mestre em ENGENHARIA ELETRICA

Março 1987

UNICAMP  
BIBLIOTECA CENTRAL

A memória do meu avô, Luiz Krutman.

Aos meus pais.

Este trabalho contou com o apoio financeiro da

CAPES, FAPESP e UNICAMP.

## AGRADECIMENTOS

-----

Ao meu Orientador, por ter me concedido a oportunidade de realização deste trabalho e pelo apoio recebido.

Ao Prof. Reginaldo Palazzo, pelo estímulo constante e pelas inúmeras e frutíferas discussões.

A Ivonete, pelas inúmeras discussões e sugestões.

Ao José Pedro, pelo apoio e pelas valiosas discussões.

Ao Edson, pela "força" e pelo suporte computacional.

Ao revisor anônimo, pelas sugestões dadas ao capítulo dois deste trabalho.

Aos Profs. Fábio Violaro e Amauri Lopes, por terem me introduzido na também fascinante área de Processamento Digital de Sinais.

Aos meus pais "adotivos", Natan e Regina, pela amizade e apoio, sem os quais este trabalho não se concretizaria.

A Graça, pela alegria transmitida e pela cessão de algumas referências.

Ao Carlão, pelo constante apoio.

Ao João Nunes, pela constante cobrança da finalização deste trabalho.

Ao Wilton, pelos desenhos.

A todos os amigos da Pós-Graduação, que colaboraram constantemente na elaboração deste trabalho.

E finalmente, um agradecimento comum a uma pessoa incomum: à minha musa inspiradora, Ting.

## I N D I C E

APRESENTAÇÃO .....	1
CAPITULO UM:	
CODIFICAÇÃO DE ALFABETO DE MODULAÇÃO REDUNDANTE.	
1.1 Introdução .....	4
1.2 O modelo discreto .....	5
1.3 As constelações M-PSK e sua faixa espectral .	8
1.4 As curvas do parâmetro exponencial $R_0$ .....	10
1.5 Códigos de alfabeto de modulação redundante ...	11
1.5.1 O codificador .....	13
1.5.2 O código .....	13
1.5.3 Códigos lineares .....	13
1.5.4 Teorema .....	14
1.5.5 Modelo do codificador .....	14
1.5.6 Decodificação .....	16
1.5.7 Evento de erro .....	17
1.5.8 Limitante inferior .....	19
1.5.9 Ganho assintótico de codificação .....	20
1.5.10 Regras de codificação .....	20
1.5.11 A técnica de mapeamento por partição em conjuntos .....	22
1.5.12 Códigos superlineares .....	24
1.5.13 A propriedade de invariância do ganho assintótico .....	27
6. Conclusão .....	31

## CAPÍTULO DOIS:

### CONSTELAÇÕES M-PSK CODIFICADAS: UM ESQUEMA DE RECEPÇÃO QUANTIZADA.

2.1	Introdução .....	33
2.2	Implementação do processo de decodificação ....	33
2.3	O parâmetro $R_0$ do limitante exponencial e a quantização do vetor recebido .....	36
2.4	Quantização do vetor recebido .....	38
2.5	As métricas dos ramos .....	40
2.5.1	Métricas ótimas .....	40
2.5.2	Métricas quantizadas .....	46
2.6	Considerações sobre o programa de simulação ...	48
2.6.1	Estimação da probabilidade de erro de bit	49
2.6.1.1	Intervalos de confiança .....	50
2.6.2	Fonte de símbolos de informação e fonte de ruído .....	51
2.6.3	Pré-processamento .....	52
2.6.4	Quantizador e associação das métricas ..	52
2.6.5	Algoritmo de Viterbi .....	56
2.6.5.1	Estado inicial do decodificador ..	56
2.6.5.2	Comprimento de decodificação ....	56
2.7	Resultados da simulação .....	57
2.7.1	A constelação 8-PSK .....	57
2.7.1.1	Quantização da componente de fase do vetor recebido .....	59
2.7.1.2	Quantização das componentes de fase e amplitude do vetor recebido	59
2.7.1.3	Um esquema de decisão abrupta ..	67
2.7.2	A constelação 16-PSK .....	71
2.7.2.1	Quantização da componente de fase do vetor recebido .....	71
2.8	Conclusão .....	74

## APENDICE 1:

LIMITANTES SUPERIORES E A PROCURA DO $d_{min}$ .....	76
--	----

APENDICE 2:

ALGUNS RESULTADOS COM METRICA QUANTIZADA .....	86
--	----

CAPITULO TRES:

DESEMPENHO EM CANAIS NAO LINEARES DE FAIXA LIMITADA:  
CANALIS DE SATELITE.

3.1 Introduçao .....	91
3.2 O modelo adotado .....	92
3.3 Representaçao complexa .....	94
3.3.1 Sinais M-PSK .....	94
3.3.2 Filtros .....	94
3.4 Teoremas .....	95
3.4.1 Teorema da amostragem para processos estocasticos .....	95
3.4.2 Teorema da simulacao para processos estocasticos .....	97
3.5 Simulacao dos filtros de transmissao e recepcao .....	102
3.5.1 Obtencao dos polos e coeficientes para os filtros Chebyshev .....	105
3.6 Geracao dos digitos de informacao .....	107
3.7 A simulacao do amplificador nao linear .....	109
3.8 Os filtros casados .....	113
3.8.1 Representaçao complexa .....	115
3.9 A escolha de $t_0$ .....	118
3.10 A simulacao do ruido .....	119
3.11 A relacao sinal ruido $E_b/N_0$ .....	121
3.12 A simulacao e os resultados .....	123

3.13 Uma nota importante .....	132
3.14 Conclusão .....	134
CONSIDERAÇÕES FINAIS .....	135
REFERENCIAS .....	138

## A P R E S E N T A Ç Ã O

---

Este trabalho se insere na área de Comunicação Digital. Mais especificamente, no problema da codificação de canal.

Nos últimos anos, a codificação de canal e o sistema de modulação utilizado têm sido considerados como uma entidade única [48]. Neste trabalho, o principal enfoque constitui-se dos sistemas de modulação M-PSK (do inglês: Phase Shift Keying) codificada.

O trabalho está dividido em três capítulos.

No capítulo um, são descritos alguns conceitos básicos, necessários ao entendimento dos códigos de alfabeto de modulação redundante. Na seção 1.5.12, é definido o conceito de um código superlinear, sendo obtida uma condição especial de suficiência para a superlinearidade. A seção 1.5.13 descreve a propriedade de invariância do ganho assintótico, de onde foi possível caracterizar três classes de códigos de alfabeto redundante.

O capítulo dois focaliza a atenção num problema de implementação dos sistemas de modulação M-PSK codificada: a quantização das coordenadas dos sinais no receptor. É feita uma análise detalhada. Resultados são obtidos para constelações 8-PSK e 16-PSK. Uma maneira simplificada, de obtenção de limitantes superiores do desempenho (recepção não quantizada), para os códigos de alfabeto redundante, é descrita no apêndice 1. Ao mesmo tempo, é feita uma sugestão para a procura do  $d_{min}$  de um código.

O capítulo três se preocupa com o desempenho das constelações M-PSK, num canal de satélite. É feita a descrição de um programa de simulação, desenvolvido com o intuito de se obter este desempenho. Resultados foram obtidos para a constelação 8-PSK. No desenvolvimento do programa, optou-se pela simulação dos filtros do canal, através de filtros digitais (ao invés de se utilizar um algoritmo de Transformada de Fourier Rápida). Isto nos levou à necessidade de demonstrarmos uma condição de suficiência para a simulação. Esta condição é a mesma utilizada no Teorema da simulação para sinais determinísticos [33]. Sendo assim, ela foi denominada "Teorema da simulação para processos estocásticos" (seção 3.4.2).

Finalmente, são feitas algumas considerações no final do trabalho, onde também é analisado um resultado obtido para o sistema de modulação 32-PSK.

C A P Í T U L O

U M:

CODIFICAÇÃO DE ALFABETO DE MODULAÇÃO REDUNDANTE

## 1.1 Introdução.

A figura 1.1 mostra os elementos principais de um sistema de comunicação digital. O propósito do sistema é enviar mensagens (sequências de símbolos) de uma fonte para um destino, com a maior taxa de transmissão e o melhor desempenho (taxa de erros) possíveis [1].

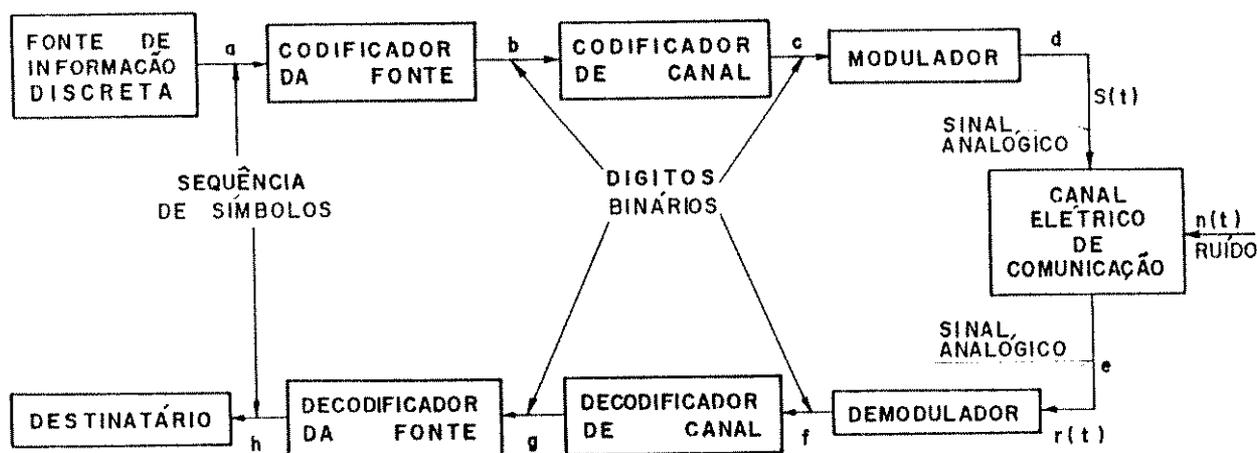


fig.1.1 Sistema de comunicação digital.

No nosso estudo estaremos focalizando o problema da codificação de canal. Os dígitos binários na entrada do codificador de canal serão por nós denominados bits de informação. Cada conjunto de  $k$  ( $k$  inteiro) bits de informação será considerado um símbolo de informação. Assim, estaremos interessados na transmissão de símbolos de informação do ponto  $b$  ao ponto  $g$ .

Com a crescente digitalização das comunicações e o aumento da utilização do espectro de frequências disponível, fez-

se necessário o desenvolvimento de sistemas mais eficientes de comunicação digital. A medida dessa eficiência é sempre um compromisso entre: complexidade, consumo de potência, largura de faixa e desempenho do sistema.

Uma técnica de codificação de canal, que melhora o desempenho dos sistemas de transmissão digital síncrona sem sacrificar a taxa de transmissão de informação ou a largura de faixa e com pequena complexidade adicionada, foi proposta por Ungerboeck [2].

A idéia considera a codificação de canal e a modulação como uma entidade única [3]. No decorrer do capítulo descreveremos detalhadamente esta idéia. Para isto, vamos antes obter um modelo discreto para o sistema de transmissão da figura 1.1, por nós focalizado (ponto b ao ponto g).

## 1.2 O modelo discreto.

O primeiro passo para a obtenção do modelo discreto é o teorema [4] descrito abaixo:

#  
Teorema 1. Qualquer conjunto de formas de onda fisicamente realizáveis com duração T,  $(s_i(t))$ ,  $0 \leq i \leq M-1$ , pode ser expresso como uma combinação linear de  $N \leq M$  formas de onda ortonormais,  $(\varphi_j(t))$ ,  $1 \leq j \leq N$ , ou seja:

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

onde  $s_{\lambda j} = \int_0^T s_{\lambda}(t) \varphi_j(t) dt$  ,  $\int_{-\infty}^{+\infty} \varphi_i(t) \varphi_j(t) dt = \delta_{ij}$  ,

$1 \leq i, j \leq N$  e  $\delta_{ij} = \begin{cases} 1, & i=j. \\ 0, & i \neq j. \end{cases}$

#

Um método de obtenção de um conjunto apropriado ortonormal para o conjunto  $\{s_i(t)\}$  é o procedimento conhecido como ortogonalização Gram-Schmidt [10].

Assim, podemos representar o conjunto  $\{s_i(t)\}$ , pelo conjunto de N-uplas  $\{\underline{s}_i\}$  onde  $\underline{s}_i = (s_{i1}, s_{i2}, \dots, s_{iN})$ ,  $0 \leq i \leq M-1$ . Os N valores serão as projeções coordenadas do vetor  $\underline{s}_i$  num espaço N-dimensional euclidiano. Este espaço é comumente chamado de Espaço de Sinais.

A saída do modulador da figura 1.1 pode ser modelada pelo conjunto de sinais  $\{s_i(t)\}$  considerado no teorema 1.

Precisamos fazer mais algumas considerações:

- $n(t)$  é um processo estocástico estacionário gaussiano branco de média zero.
- o canal elétrico de comunicação será modelado por uma função  $h(s(t), n(t))$ . Estaremos considerando que  $r(t) = h(s(t), n(t)) = n(t) + s(t)$ . Ou seja, o ruído é também aditivo. Este ruído é comumente conhecido por AWGN (do inglês - Additive White Gaussian Noise).
- o demodulador está em perfeito sincronismo temporal com o modulador (ele sabe quando foi transmitido cada sinal).
- estaremos assumindo também que o demodulador conhece

perfeitamente as possíveis frequências e fases dos sinais na saída do modulador.

Multiplicando ambos os lados da equação  $r(t) = s(t) + n(t)$  por  $\varphi_j(t), 1 \leq j \leq N$ , e integrando, obtemos:

$$\int_0^T r(t) \varphi_j(t) dt = \int_0^T s_i(t) \varphi_j(t) dt + \int_0^T n(t) \varphi_j(t) dt$$

onde o primeiro termo à direita é  $s_{ij}$  e o segundo chamaremos de  $n_j$ . Ou seja,  $r_j = s_{ij} + n_j$ .

Considerando as hipóteses sobre  $n(t)$  pode-se demonstrar o seguinte teorema [4]:

#  
Teorema 2. Seja um conjunto ortonormal  $(\varphi_j(t)), 1 \leq j \leq N$ , que caracteriza o espaço de sinais e um ruído estacionário gaussiano branco de média zero, com densidade espectral de potência  $N_0/2$  watts/cps para todo  $f$  (positivo e negativo).

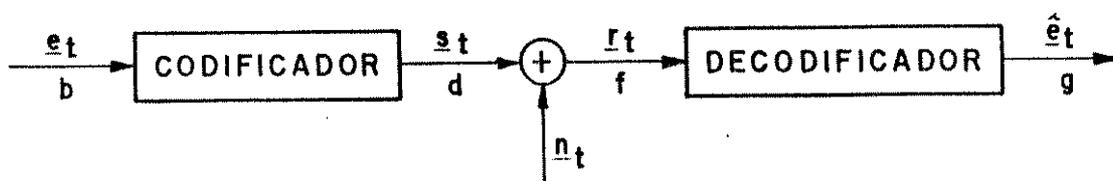
Este ruído pode ser decomposto em duas partes: uma primeira  $\sum_{j=1}^N n_j \varphi_j(t)$ , que é a projeção do ruído no espaço de sinais, e, uma segunda consistindo da porção de ruído que é ortogonal ao espaço de sinais.

Os  $n_j, 1 \leq j \leq N$ , serão variáveis aleatórias gaussianas independentes, com média zero e variância  $\sigma_j^2 = N_0/2, 1 \leq j \leq N$ . Além disto, amostras sucessivas dos  $n_j$  no tempo também serão independentes.

#

Então, podemos substituir o modelo da figura 1.1 por um modelo em termos de vetores  $N$ -dimensionais; o tempo agora é

uma variável discreta  $t$  (um inteiro),  $0 \leq t < +\infty$  (figura 1.2).



$e_t$  — Símbolo de informação no instante  $t$   
 $\hat{e}_t$  — Estimação do símbolo de informação

fig.1.2 Modelo discreto.

### 1.3 As constelações M-PSK e sua faixa espectral.

O conjunto de formas de onda M-PSK coerente é descrito por:

$$s_i(t) = \sqrt{2E_s/T} \cos(\omega_0 t + 2\pi i/M + \pi/M) g(t), \quad 0 \leq i \leq M-1,$$

onde  $g(t) = \begin{cases} 1 & ; 0 < t < T ; \\ 0 & ; \text{fora,} \end{cases}$

$E_s$  é a energia de  $s_i(t)$  e  $\omega_0 = 2\pi n_0/T$ ,  $n_0$  inteiro.

A representação no espaço de sinais é um conjunto de pontos sobre um círculo centrado na origem de raio  $\sqrt{E_s}$  e espaçados de ângulos iguais a  $2\pi/M$ .

Considere a representação complexa [5,6] em banda base dos sinais M-PSK na saída do modulador [figura 1.1]:

$$u(t) = \sum_{k=0}^{\infty} I_k g(t-kT) \quad (1.1)$$

onde  $I_k = \sqrt{2E_s/T} \exp(j\theta_k)$  ;

$\theta_k$  é uma das M fases da constelação

e  $g(t)$  é o pulso retangular definido anteriormente.

Pode-se mostrar [6] que a densidade espectral de potência de sinais representados pela equação 1.1, é dada por:

$$S_s(\omega) = 1/2 [S_u(\omega - \omega_0) + S_u(-\omega - \omega_0)] \quad (1.2)$$

onde  $S_u(\omega) = 1/T |G(\omega)|^2 S_k(\omega)$ , é a densidade espectral de potência de  $u(t)$ ,

$$S_k(\omega) = \sum_{m=-\infty}^{\infty} R_k(m) \exp(-j\omega m T),$$

$R_k(m) = 1/2 E(I_k^* I_{k+m})$ , "\*" denotando o conjugado,  $E(\cdot)$  a média estatística, e  $G(\omega)$  é a transformada de Fourier de  $g(t)$ .

Suponha agora que a transmissão não é codificada, e que os símbolos  $I_k$  são igualmente prováveis e estatisticamente independentes (o que é razoável para a maioria dos sistemas práticos).

Então podemos escrever:

para  $m \neq 0$ ,

$$\begin{aligned} R_k(m) &= 1/2 E(I_k^*) E(I_{k+m}) = \\ &= (E_s/T) (1/M^2) \left[ \sum_{k=0}^{M-1} \cos \theta_k + j \sum_{k=0}^{M-1} \text{sen} \theta_k \right] \left[ \sum_{l=0}^{M-1} \cos \theta_l + j \sum_{l=0}^{M-1} \text{sen} \theta_l \right] = \\ &= 0; \end{aligned}$$

para  $m=0$ ,

$$R_k(m) = 1/2 E(|I_k|^2) = E_s/T.$$

Logo, o espectro dos sinais M-PSK será obtido da equação 1.2, onde:

$$S_u(\omega) = E_s |G(\omega)|^2 / T^2 = E_s S_a^2(\omega T/2)$$

e  $S_a(.)$  é a função amostragem.

Este resultado mostra que, independentemente do número de sinais  $M$ , o pulso  $g(t)$  de banda base é o que determinará a faixa espectral dos sinais não codificados.

Para os sistemas codificados, simulações anteriores [7,8] têm mostrado que a correlação introduzida entre os símbolos  $I_k$  é pequena. Portanto, podemos considerar a faixa espectral dos sistemas codificados como sendo aproximadamente a mesma dos sistemas não codificados.

#### 1.4 As curvas do parâmetro exponencial $R_0$ .

Ungerboeck [2] mostrou que se aumentarmos o tamanho do alfabeto de modulação mantendo a taxa de transmissão de bits de informação inalterada, podemos diminuir o consumo de potência sem modificarmos o desempenho do sistema. Para isso ele utilizou as curvas de capacidade de canal.

O mesmo efeito pode ser observado nas curvas do parâmetro exponencial  $R_0$ , a taxa de corte do canal [10,3,9].

Para o AWGN, a expressão do  $R_0$  é extremamente simples [3]:

$$R_0 = -\log_2 \left( \min_{D(.)} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \exp(-|s_i - s_j|^2 / 4 N_0) D(i)D(j) \right)$$

onde  $D(.)$  é a distribuição dos símbolos de informação.

Estaremos considerando sempre  $D(i)=1/M$ ,  $0 \leq i \leq M-1$ , uma situação na maioria das vezes encontrada na prática.

Utilizando a expressão acima foi possível obter as

curvas do parametro  $R_0$  para as constelações M-PSK, como mostra a figura 1.3.  $E_b$  é a energia média por bit de informação, definida como:  $E_b = E_s / \log M$ .

Por exemplo: se quisermos transmitir 4 bits de informação por intervalo T de duração do símbolo de canal, a uma taxa de erro de símbolo de  $10^{-5}$ , e usarmos um alfabeto com 32 símbolos, podemos diminuir o consumo de potência em aproximadamente 6 dB (supondo um esforço de codificação infinito). O mesmo se pode afirmar para 5 e 6 bits de informação, se usarmos alfabetos de 64 e 128 símbolos respectivamente.

Outro fato importante, a ser observado nas curvas, é que o uso de modulação com um alfabeto quatro ou mais vezes maior que o alfabeto original, traz uma diminuição adicional no consumo de potência de alguma coisa em torno de 2 dB para uma taxa de 2 bits/T, e ainda menos para taxas maiores. Sendo assim, talvez não seja compensador a utilização de alfabetos de modulação maiores que duas vezes o alfabeto original.

### 1.5 Códigos de alfabeto de modulação redundante.

Como vimos no item anterior a redundância requerida no processo de codificação foi introduzida através do aumento do alfabeto de modulação. Por isso mesmo, esta classe de códigos é denominada Códigos de Alfabeto Redundante (AR) [9].

Vamos agora definir alguns conceitos básicos, ver teoremas, e descrever características dos códigos AR. Os itens 1.5.1 a 1.5.4 foram obtidos da referência [11].

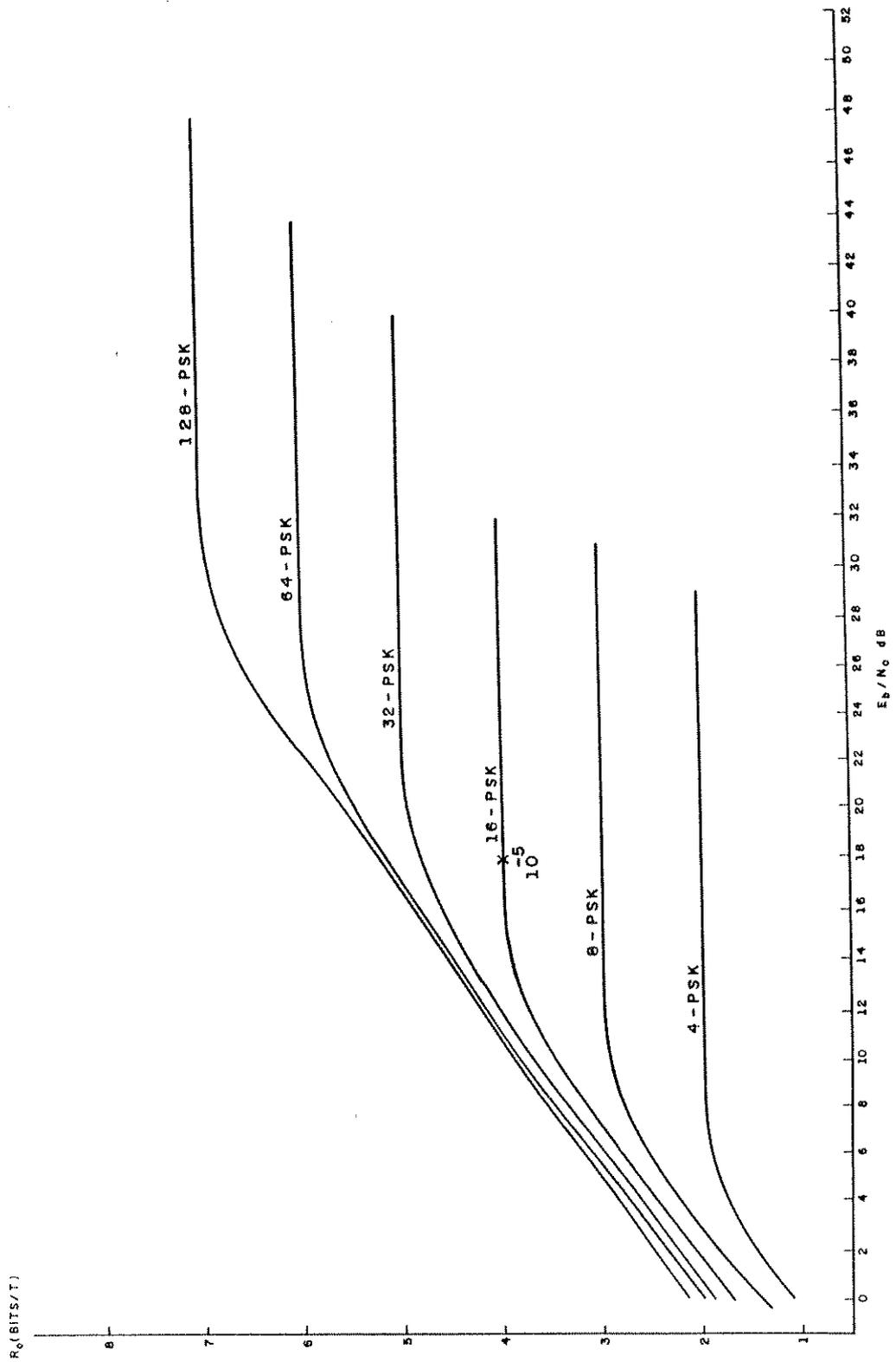


fig. 1.3 Curvas do parâmetro exponencial  $R_0$  para as constelações M-PSK.

### 1.5.1 O codificador.

É um dispositivo que aceita uma sequência de símbolos de informação ( $\underline{e}_t$ ) pertencentes a um alfabeto E e envia para o canal uma sequência correspondente de símbolos ( $\underline{s}_t$ ) pertencentes a um alfabeto S. Cada símbolo  $\underline{e}_t$  corresponde a uma k-upla ( $e_1, e_2, \dots, e_k$ ) de dígitos binários. A cardinalidade do alfabeto S também será considerada uma potência de 2.

### 1.5.2 O código.

Seja  $\underline{a}$  o vetor correspondente à l-upla ( $a_1, a_2, \dots, a_l$ ) e  $A^l$  o conjunto de todas as l-uplas. Um código C é uma função de  $A^l$  em S,  $S=C(\underline{a})$ .

Se k bits entram no codificador para cada símbolo de saída  $\underline{g}$ , teremos sempre  $l \geq k$ .

Seja  $m = l - k$ . Se  $m = 0$ , teremos um código de bloco. Se  $m > 0$ , um código de treliça.

Para o código de treliça, m representa a memória do codificador. Os símbolos  $\underline{g} \in S$  representam formas de onda do canal.

### 1.5.3 Códigos lineares.

O conjunto de todas as l-uplas binárias  $A^l$ , forma um grupo comutativo em relação à soma " $\oplus$ " no corpo de Galois GF(2),  $A^l = (A^l; \oplus)$ . Um código é linear se:

1) Existe uma operação chamada "soma", "+", sob a qual o alfabeto de saída  $S$  forma um grupo comutativo,  $\mathcal{S} = (S; +)$ .

2)  $C$  é um homomorfismo de  $A$  em  $\mathcal{S}$ . Então,  $C(\underline{a}_i \oplus \underline{a}_j) = C(\underline{a}_i) + C(\underline{a}_j)$ .

Na definição acima, é assumido que o codificador é invariante com o tempo. Assim, podemos falar em "soma" de sequências, tanto na entrada quanto na saída.

#### 1.5.4 Teorema.

#  
Teorema 3. Para um dado valor de  $k$ , de  $m$  e de  $n$ , o conjunto de todos os códigos lineares podem ser obtidos através de um codificador convolucional  $(k, n)$  de memória  $m$ , seguido de uma função bijetora dos  $n$  bits de saída do codificador convolucional em  $S$ .

#

É fácil mostrar que se associamos aos símbolos do alfabeto de saída, o número binário correspondente aos  $n$  bits da saída do codificador convolucional, a operação soma da saída é a mesma soma " $\oplus$ " do corpo de Galois  $GF(2)$ , definida para o grupo  $A^l$ .

#### 1.5.5 Modelo do codificador [9].

É mostrado na figura 1.4. A memória do codificador em relação ao  $i$ -ésimo bit do símbolo  $\underline{e}_t$ , é representada por  $m_i$ , de tal maneira que  $\sum_{i=1}^k m_i = m$ .

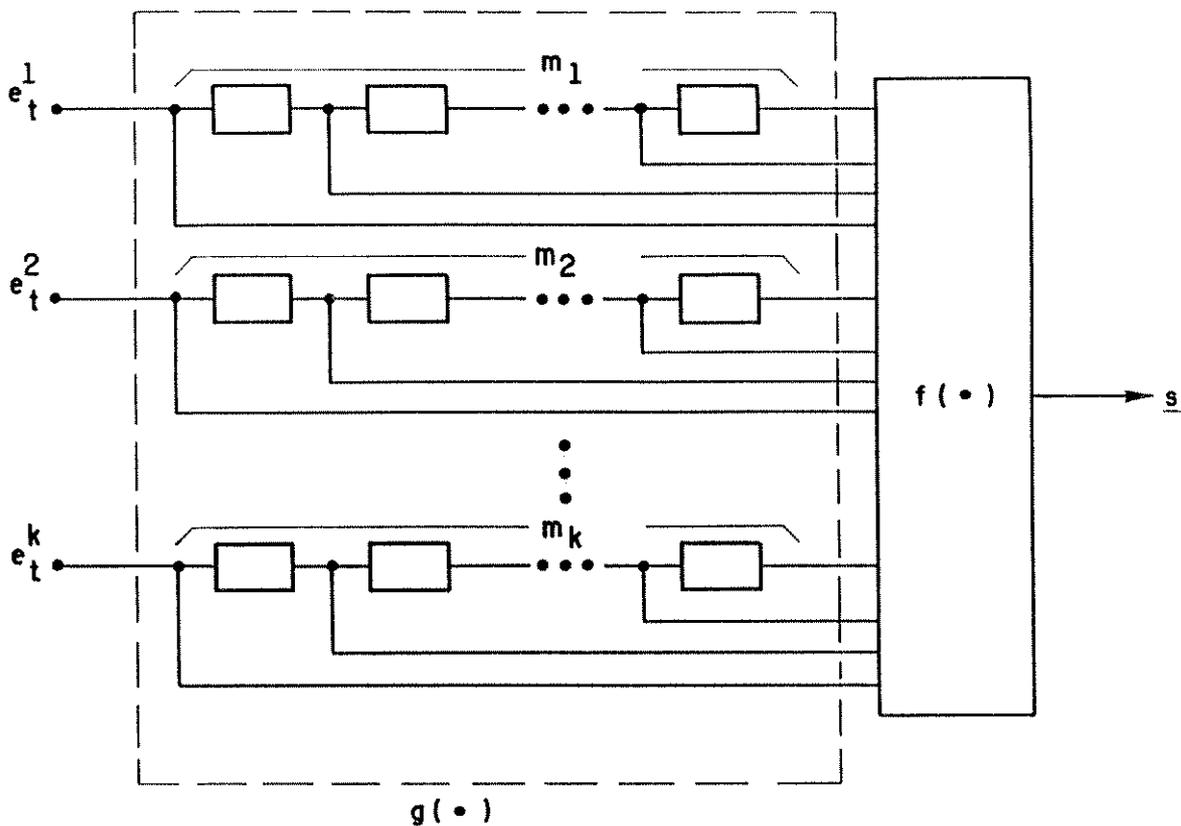


fig.1.4 Modelo do codificador (  $\square$  representa elementos de memória).

A relação entrada-saída pode ser escrita da seguinte forma:  $\underline{s}_t = f(\underline{e}_t, \underline{e}_{t-\tau}, \dots, \underline{e}_{t-\gamma\tau})$ , onde  $\tau$  é o tempo de duração do símbolo de canal, e  $\gamma$  é o maior dos  $m_i$ :  $\gamma = \max_i (m_i)$ .

O estado do codificador é definido como:

$$\underline{\sigma}_t = (\underline{e}_{t-\tau}, \dots, \underline{e}_{t-\gamma\tau}).$$

$$\text{Assim: } \underline{s}_t = f(\underline{e}_t, \underline{\sigma}_t), \quad (1.3.1)$$

$$\underline{\sigma}_{t+\tau} = g(\underline{e}_t, \underline{\sigma}_t). \quad (1.3.2)$$

Códigos que são dependentes de estados podem ser

representados convenientemente por uma treliça [13,12]. Uma treliça é uma estrutura periódica onde podemos acompanhar as transições (ramos) dos estados (nós) do codificador que ocorrem devido aos símbolos de entrada que influenciam a sua memória. Codificar consiste então, em associar a cada transição entre pares de estados, um símbolo  $\underline{s} \in S$  [figura 1.5].

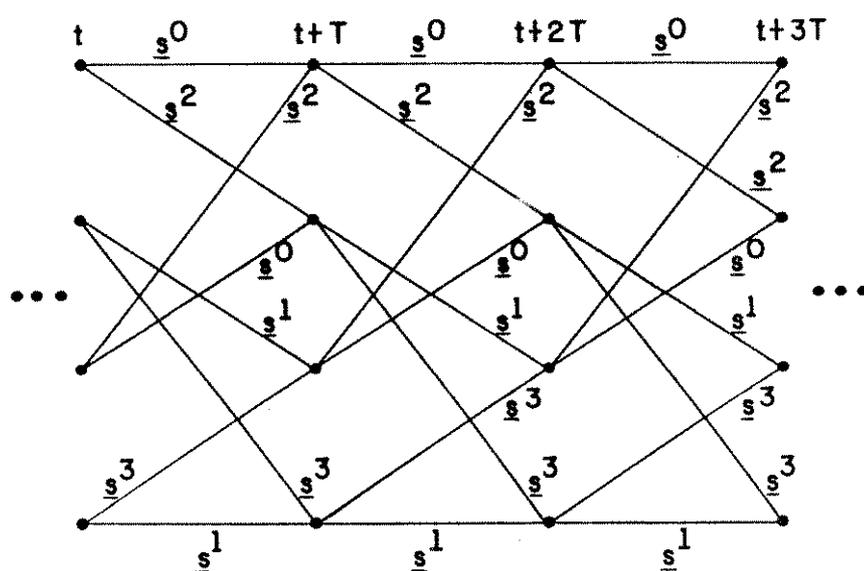


fig.1.5 Um exemplo de treliça e símbolos associados.  
( $k=1, n=2, m=m =2$ )

### 1.5.6 Decodificação.

A decodificação é feita pelo critério da Máxima verossimilhança [5,13].

Seja  $\underline{R} = (r_1, r_2, \dots, r_N)$  uma sequência de símbolos recebida, e  $\underline{s}^i = (s_1^i, s_2^i, \dots, s_N^i)$  uma das possíveis sequências

enviadas. O decodificador escolhe  $\underline{S}^k$  se:

$$P[\underline{R}/\underline{S}^k] > P[\underline{R}/\underline{S}^j] \text{ para todo } j \neq k.$$

Definindo a distância euclidiana entre duas sequências como:

$$d^2(\underline{X}, \underline{Y}) = \sum_t d^2(x_t, y_t)$$

onde  $d( , )$  é distância euclidiana e

$x_t(y_t)$  é um símbolo pertencente a sequência  $\underline{X}(\underline{Y})$  no instante  $t$ .

Então para o AWGN a regra de decisão acima é equivalente

a: Escolha  $\underline{S}^k$  se,

$$d^2(\underline{R}, \underline{S}^k) < d^2(\underline{R}, \underline{S}^j) \text{ para todo } j \neq k.$$

A demonstração é simples, e é uma extensão da aplicação do critério de máxima verossimilhança para o AWGN em sistemas não codificados [10,14].

### 1.5.7 Evento de erro.

Considere uma sequência enviada e uma outra sequência qualquer do código, diferente da enviada. Se elas estão próximas no sentido de distância euclidiana, um pequeno ruído poderá levar o receptor a decidir por uma sequência diferente da enviada. Daí vem a definição de evento de erro. Considere a figura 1.6. Se  $\underline{S}$  é a sequência enviada e  $\hat{\underline{S}}$  é a sequência decodificada, durante um determinado tempo  $\underline{S}$  e  $\hat{\underline{S}}$  irão divergir e se reencontrar várias vezes. Cada separação distinta é chamada de evento de erro. Geralmente a probabilidade de ocorrer qual-

quer evento de erro num intervalo de tempo  $NT$ , tende à unidade quando  $N$  tende a infinito [13]. Então, quando nos referirmos à probabilidade de evento de erro, estará implícito que esta probabilidade é condicionada a um determinado instante  $jT$  da treliça, cujo estado inicial do evento de erro é correto (esta probabilidade é também chamada probabilidade de primeiro evento de erro [12]).

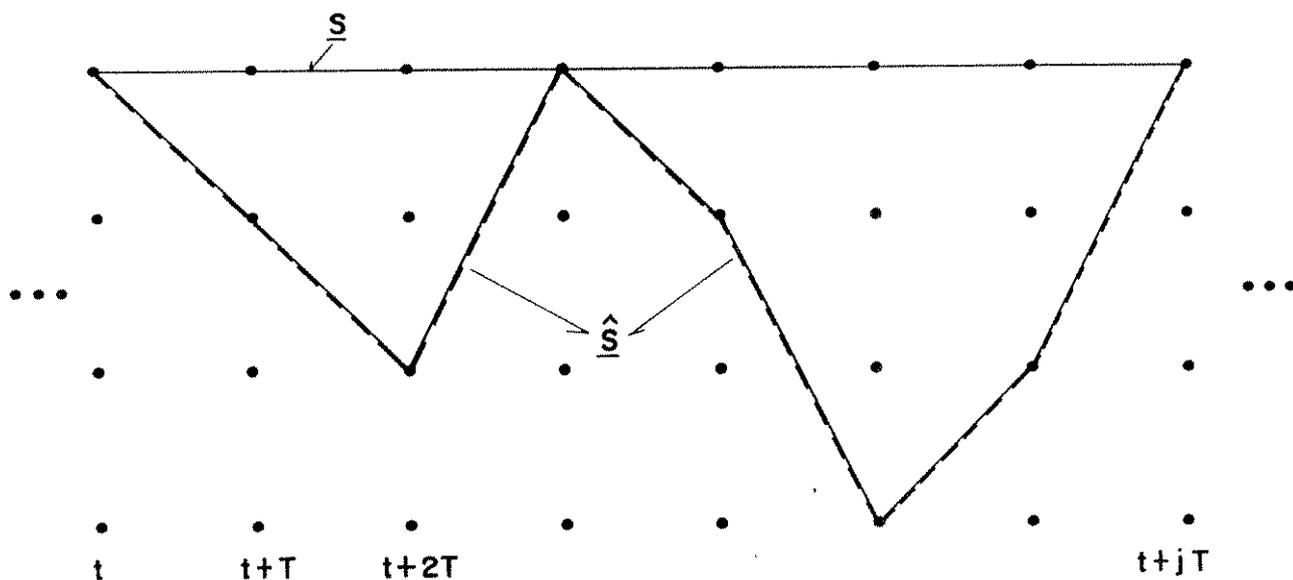


fig.1.6 Eventos de erro.  $S$  é a sequência transmitida e  $\hat{S}$  é a decodificada.

### 1.5.8 Limitante inferior.

A menor das distâncias entre todas as possíveis sequências de símbolos do alfabeto de modulação, é denominada [9] distância mínima do código, ou seja,

$$d_{\min}^2 = \min_{\underline{X} \neq \underline{Y}} d^2(\underline{X}, \underline{Y}) = \min_{\underline{X} \neq \underline{Y}} \sum_t d^2(x_t, y_t) .$$

Para relações sinal ruído altas, pode-se mostrar [13], que a probabilidade de evento de erro atinge assintoticamente, o limitante inferior:

$$\Pr(e) \geq N(d_{\min}) Q(d_{\min}/2\sigma), \quad (1.4)$$

onde  $N(d_{\min})$  é o número médio de sequências com distância  $d_{\min}$ ,

$\sigma^2$  é a variância do ruído em cada dimensão,

e  $Q(\cdot)$  é a função gaussiana de probabilidade de erro,

que é monotonicamente decrescente:

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

Então, para um determinada estrutura de memória, maximizar esta distância mínima é o nosso objetivo no sentido de otimizar o desempenho do sistema codificado.

É interessante notar a analogia que há entre o sistema codificado e o não codificado. Assim como a distância mínima da constelação não codificada era o parâmetro relevante em relações sinal ruído altas, o  $d_{\min}$  será agora o parâmetro relevante do sistema codificado.

### 1.5.9 Ganho assintótico de codificação.

Da equação 1.4 podemos definir um ganho assintótico de codificação [2] como sendo a relação:

$$G = d_{\min}^2 / d_0^2$$

onde  $d_0$  é distância mínima do sistema não codificado.

O sistema não codificado pode ser visto como uma treliça de um só estado cujos eventos de erros serão erros entre símbolos da constelação não codificada [2]. Sendo assim, a probabilidade de evento de erro que agora se confunde com a probabilidade de erro de símbolo, atinge assintoticamente o valor  $[N(d_0) Q(d_0/2\sigma)]$ , onde  $N(\cdot)$ ,  $Q(\cdot)$  e  $\sigma$  têm o mesmo significado da equação 1.4.

### 1.5.10 Regras de codificação.

Voltando agora a atenção para a estrutura de treliça descrita anteriormente, podemos observar que de cada nó da treliça no instante  $t$ , se origina um ramo para cada símbolo de entrada, que atinge um outro nó correspondente ao estado do codificador no instante  $t+T$ .

Cada nó deve ter  $2^k$  ramos divergindo dele e  $2^k$  ramos convergindo para ele. A estrutura de treliça é determinada pela função  $g(\cdot)$  (equação 1.3.2), e a função  $f(\cdot)$  (equação 1.3.1) nos diz como os símbolos de canal são associados a cada ramo da treliça. Está claro que, dependendo de  $g(\cdot)$ , haverá situações em

que dois ou mais ramos conectam os mesmos nós. Teremos então, transições paralelas como mostra a figura 1.7 [9]. Isto permitirá a ocorrência de eventos de erro de comprimento unitário.

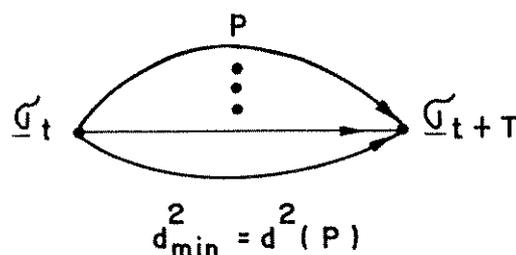


fig.1.7 Subconjunto P de transições paralelas.  $d(P)$  representa a menor distância entre símbolos do subconjunto P.

Nesta situação, como primeira regra de codificação, devemos associar ao conjunto de transições paralelas um subconjunto de símbolos do canal cuja distância mínima entre eles seja a maior possível.

Para eventos de erro com comprimento diferente do unitário, obtemos uma segunda regra. Como mostra a figura 1.8 [9], eles se originam num estado inicial e após  $j$  ( $j > 1$ ) instantes dois ramos irão convergir para um único estado. A distância entre duas possíveis sequências de símbolos irá incluir os termos  $d^{\lambda}(P1, P2)$  e  $d^{\lambda}(P3, P4)$ . Então, como segunda regra devemos associar aos subconjuntos que se originam num mesmo estado ou convergem para um mesmo estado, a maior distância possível.

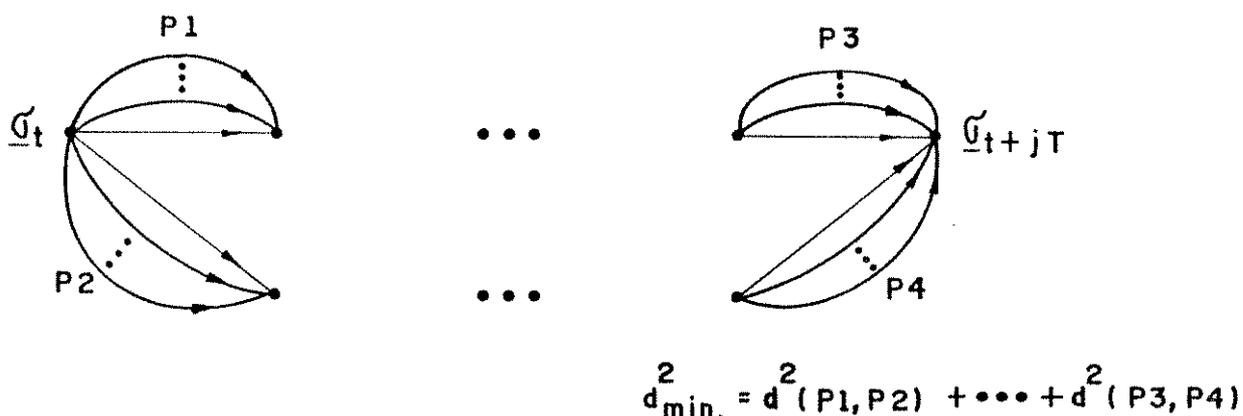


fig.1.8 Eventos de erro de comprimento diferente do unitário.  $d(x,y)$  representa a menor distância entre símbolos do subconjunto  $x$  e do subconjunto  $y$ .

Como terceira regra (geral em qualquer processo de codificação), devemos fazer com que os símbolos de canal ocorram com igual frequência e razoável regularidade e simetria.

Estas regras foram propostas por Ungerboeck [2], e o método por ele proposto para alcançá-las é o particionamento do conjunto de sinais em subconjuntos com a distância mínima entre sinais de um mesmo subconjunto, aumentando a cada partição.

#### 1.5.11 A técnica de mapeamento por partição em conjuntos.

Considere a figura 1.9. A constelação de  $M$  sinais não particionada tem uma distância mínima  $d(A)$ . A cada nível da árvore formada pela partição, o número de sinais em cada subconjunto é sempre duas vezes menor que o número de sinais do nível

anterior. Os subconjuntos são obtidos de tal maneira que a cada partição a distância mínima intra-subconjunto (a menor das distâncias entre os símbolos de cada subconjunto) é cada vez maior ( $d(A) > d(B_i) > d(C_i) > d(D_i) > \dots$ ).

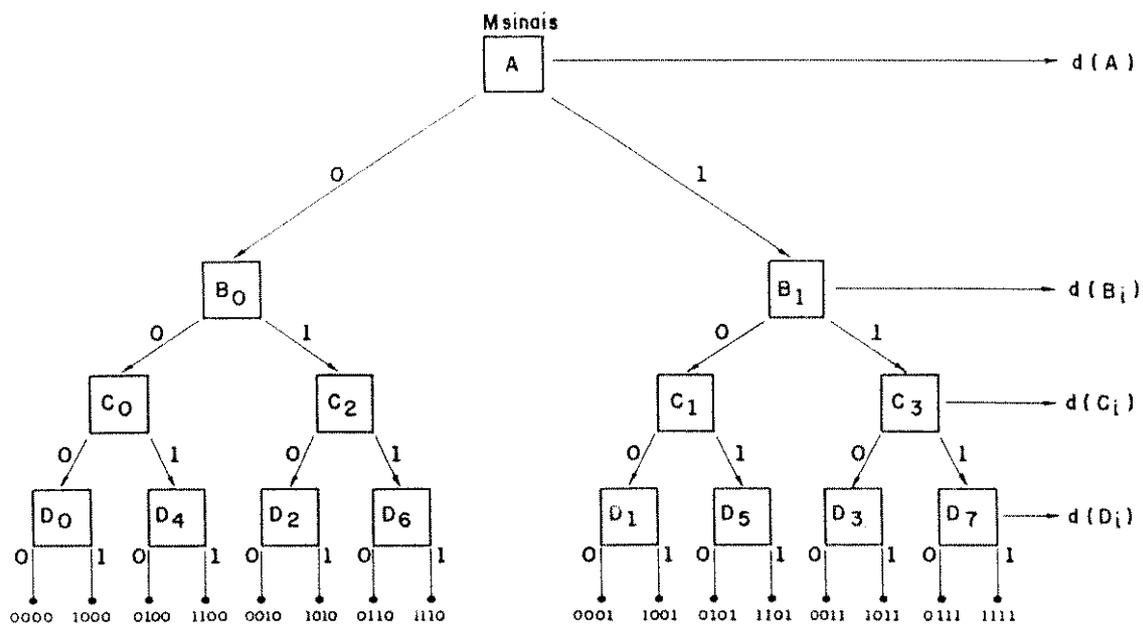


fig.1.9 Partição de uma constelação de  $M(=16)$  sinais.

Para as constelações analisadas por Ungerboeck (AM,PSK,QAM), existe uma simetria que faz com que todos os subconjuntos obtidos em qualquer nível da árvore sejam idênticos (a menos de uma rotação e/ou deslocamento dos eixos coordenados), e portanto, possuam as mesmas distâncias mínimas intra-subconjunto (naturalmente não tem sentido falar em distância mínima intra-subconjunto para o último nível da árvore se o código não contém

transições paralelas).

O mapeamento binário dos bits codificados também está ilustrado na figura 1.9. A função específica que relaciona a saída do codificador convolucional ao alfabeto de modulação não é importante. O que é relevante é que os subconjuntos contenham distâncias mínimas cada vez maiores.

#### 1.5.12. Códigos superlineares [11].

Vamos agora descrever a propriedade de superlinearidade para os códigos AR.

#

Definição. Um código  $C$  é superlinear se ele é linear e se também é possível associar a cada símbolo de saída  $\underline{s} \in S$  um número real  $T(\underline{s})$ , chamado peso de  $\underline{s}$ , de tal maneira que:

$$d(\underline{s}_i, \underline{s}_j) = T(\underline{s}_i + \underline{s}_j),$$

onde  $d(\underline{s}_i, \underline{s}_j)$  é a distância euclidiana entre  $\underline{s}_i$  e  $\underline{s}_j$ .

#

Considere que associamos aos símbolos do alfabeto de saída as  $n$ -uplas binárias da saída do codificador convolucional.

Portanto, a operação soma da saída é a soma " $\oplus$ ", ou seja,

"+"  $\equiv$  " $\oplus$ ".

Considere que o código é superlinear. Defina o peso entre duas sequências,  $T(\underline{s}_i \oplus \underline{s}_j)$ , como sendo a distância euclidiana entre as mesmas. Ou seja, o quadrado do peso entre duas sequências é a soma dos quadrados dos pesos entre os respectivos símbolos que compõem as sequências.

Para códigos lineares binários estamos sempre trabalhando com distâncias de Hamming. Na referência [14,pg.85]

mostra-se que o conjunto de distâncias de Hamming entre um dado vetor do código e os outros vetores do código é o mesmo para todos os vetores do código. Se substituirmos a palavra "vetor do código" por "sequência do código", "distância de Hamming" por "distância euclidiana" e "peso de Hamming" pelo peso entre sequências definido acima, a demonstração da referência nos leva a concluir que o conjunto de distâncias euclidianas entre uma dada sequência de símbolos do código e as outras sequências é o mesmo para todas as sequências do código.

Portanto, podemos encontrar o  $d_{min}$  de um código superlinear fixando uma sequência de referência, sem necessidade de compararmos todos os possíveis pares de sequências.

Uma maneira simples de verificarmos a superlinearidade de um código é a seguinte [11]:

1. Construa uma matriz H da soma, dois a dois, entre todos os símbolos do alfabeto de saída.
2. Construa uma matriz D das distâncias entre todos os símbolos do alfabeto de saída.
3. Compare as entradas de H e D. Se entradas iguais em H corresponderem a entradas iguais em D, o código é superlinear.

Em outras palavras, se  $\underline{s}_i + \underline{s}_j = \underline{s}_k + \underline{s}_l$ , implica em  $d(\underline{s}_i, \underline{s}_j) = d(\underline{s}_k, \underline{s}_l)$ , o código é superlinear.

Como tínhamos mencionado, o processo de codificação consiste em associar os símbolos do alfabeto de modulação às transições entre estados da treliça. Quando há transições paralelas, apenas a menor distância entre dois símbolos quaisquer

pertencentes a dois subconjuntos distintos de transições paralelas, é que é relevante na procura do  $d_{\min}$  de um código. Estas distâncias são na verdade as distâncias mínimas intra-subconjuntos de cada nível da partição. Por exemplo (figura 1.9):  $d(D_0, D_4) = d(C_0)$ ,  $d(D_0, D_2) = d(B_0)$ ,  $d(D_0, D_1) = d(A)$ , etc; onde  $d(X, Y)$  é a menor distância euclidiana entre símbolos pertencentes a X em relação aos símbolos de Y e  $d(X)$  é a menor distância entre os símbolos de X.

Dessa maneira, podemos utilizar a definição de superlinearidade descrita anteriormente, para uma treliça sem transições paralelas em que os elementos do alfabeto de saída serão os subconjuntos pertencentes ao último nível da partição. Todas as considerações feitas para os símbolos de saída são válidas agora para os subconjuntos do último nível.

Estamos agora em condições de mostrar que todos os códigos que contêm transições paralelas e têm a simetria mencionada no item 1.5.11, são superlineares.

Sem perda de generalidade, considere a figura 1.9. Associe aos subconjuntos  $D_i$  de índice ímpar o peso  $d(A)$ ,  $T(D_{\text{ímpar}}) = d(A)$ . Vá para o lado esquerdo da árvore. Retire o bit menos significativo (do índice) de cada subconjunto  $D_i$  de índice par. Associe aos novos subconjuntos de índice ímpar ( $D_1^1$  originário de  $D_2$  e  $D_3^1$  originário de  $D_6$ ), o peso  $T(D_{\text{ímpar}}^1) = d(B_0) = d(B_1)$ . Associe a  $D_2^1$  (originário de  $D_4$ ) o peso  $T(D_2^1) = d(C_0) = d(C_1)$ . E finalmente, associe a  $D_0^1$  o peso  $T(D_0^1) = 0$ . Como no  $GF(2)$  a soma de um binário par a um binário ímpar é sempre ímpar, e, a cada nível da partição todos os subconjuntos

são idênticos, a propriedade de superlinearidade fica estabelecida. Para uma partição diferente da figura 1.9, uma associação de pesos análoga estabelecerá a propriedade de superlinearidade.

Assim, a maioria dos códigos encontrados por Ungerboeck são superlineares, como também os das constelações M-PSK que será o enfoque do nosso trabalho.

### 1.5.13 A propriedade de invariância do ganho assintótico.

Seja um código  $C$  que contém transições paralelas.

O valor de  $d_{min}^2$  é uma função linear dos quadrados das distâncias intra-subconjuntos da árvore da partição:  $d^2(A)$ ,  $d^2(B1)$ ,  $d^2(C1)$ , etc. Adicionando-se transições paralelas ao código  $C$ , obtemos um código  $C'$  que agora irá utilizar uma constelação com número maior de sinais. Sob a mesma restrição de energia média da constelação, se as distâncias intra-subconjuntos diminuírem de um fator  $K$ , o  $d_{min}^2$  irá diminuir de um fator  $K^2$ .

Verificam-se as seguintes propriedades:

1. Se o código  $C$  não era catastrófico [12],  $C'$  também não será.
2. Se  $C$  era um código ótimo, no sentido de maximizar o valor de  $d_{min}^2$  para uma determinada configuração de memória,  $C'$  também será ótimo para a mesma configuração de memória [17].
3. Se a distância mínima da constelação não codificada (de referência) também diminuir do fator  $K$  citado acima, o ganho assintótico de codificação se mantém inalterado.

Exemplo 1 - Os mesmos códigos obtidos por Ungerboeck para

constelações 16-QAM e 64-QAM, podem ser utilizados com qualquer constelação  $2^m$ -QAM,  $m$  par. As constelações QAM desta classe de códigos possuem a propriedade:  $d(X_i) = \sqrt{2} d(X_{i-1})$ ,  $i=1,2,\dots$ , onde  $i$  representa o nível da árvore. Se  $X_0$  representa a constelação original QAM,  $X_1$  representará as constelações da primeira partição: constelações AM-PM. Por exemplo, se tínhamos um código para 64-QAM e adicionamos duas linhas de entrada não codificadas, obteremos um código para 256-QAM. Se  $d(X_0)$  para 256-QAM é igual a  $K$  vezes  $d(X_0)$  para 64-QAM,  $d(X_1)$  para 256 será também  $K$  vezes  $d(X_1)$  para 64. E assim por diante. O  $d_{\min}^2$  diminuirá de um fator  $K^2$ .

Se  $X_0$  tinha uma energia média  $E_m$ , as constelações  $X_1$  (AM-PM) terão também energia média  $E_m$ . Como a constelação não codificada de referência, é a constelação AM-PM sob a mesma restrição de energia, concluímos que a distância mínima das mesmas,  $d_0$ , é igual a  $d(X_1)$ . Portanto  $d_0^2$  diminuirá do mesmo fator que  $d_{\min}^2$ .

Exemplo 2 - As constelações M-AM possuem  $d(A)$  igual a :  $[12/M^2 - 1]^{1/2}$  e  $d(X_i) = 2 d(X_{i-1})$ ,  $i=1,2,\dots$ . Para  $M$  grande,  $d(A) \cong \sqrt{12}/M$ . Então, é de se esperar que os códigos obtidos por Ungerboeck para  $M = 16$ , possam ser utilizados para  $M > 16$  com ganhos assintóticos bem próximos.

Exemplo 3 - Códigos para Constelações M-PSK.

Considere agora uma partição em que o último nível da árvore contenha os conjuntos  $C_i$  da figura 1.9. Como existem quatro subconjuntos, isto é o correspondente a termos só um dos  $k$

bits de entrada codificado, ou só uma linha de entrada com elementos de memória. Se aplicarmos as regras de codificação a uma treliça de quatro estados, obteremos o código mostrado na figura 1.10.

É evidente que o maior  $d_{min}$  nunca poderá exceder a distância entre transições paralelas  $d(C1)$ , pois elas formam um evento de erro de comprimento unitário. Se o  $d_{min}$  do código não for  $d(C1)$  será função das distâncias mínimas entre subconjuntos  $C1$ , ou seja:  $d(A)$ ,  $d(B1)$ . As distâncias  $d(A)$ ,  $d(B1)$  e  $d(C1)$ , para o caso das constelações M-PSK, correspondem às distâncias mínimas das constelações não codificadas,  $d_{M-PSK} = 2 \text{ sen}(\pi/M)$ . Para M grande é válida a aproximação  $\text{sen}(\pi/M) \cong \pi/M$ . Então, é de se esperar que para  $M = 64$  obtenhamos aproximadamente os mesmos ganhos assintóticos encontrados em [11] para  $M = 32$ . O mesmo raciocínio será válido para  $M = 128$  em relação à  $M=64$  e assim por diante. Para uma treliça de 4 estados e  $M=8$ , como  $d_{min} = d(C1)$  [2], não compensaria aumentar o número de estados. Para  $M > 8$ ,  $d_{min}^2 = 2 d^2(B1) + d^2(A)$ , e portanto, um aumento no número de estados nos leva a obter maiores ganhos.

A Tabela 1 mostra os resultados obtidos em [11] até  $M=32$ . Sem usar a aproximação acima nós procuramos o  $d_{min}$  de códigos para taxas 5/6 (64-PSK) e 6/7 (128-PSK), e com memória  $m=2,3$ , obtendo ganhos assintóticos aproximadamente os mesmos dos obtidos para  $M = 32$ .

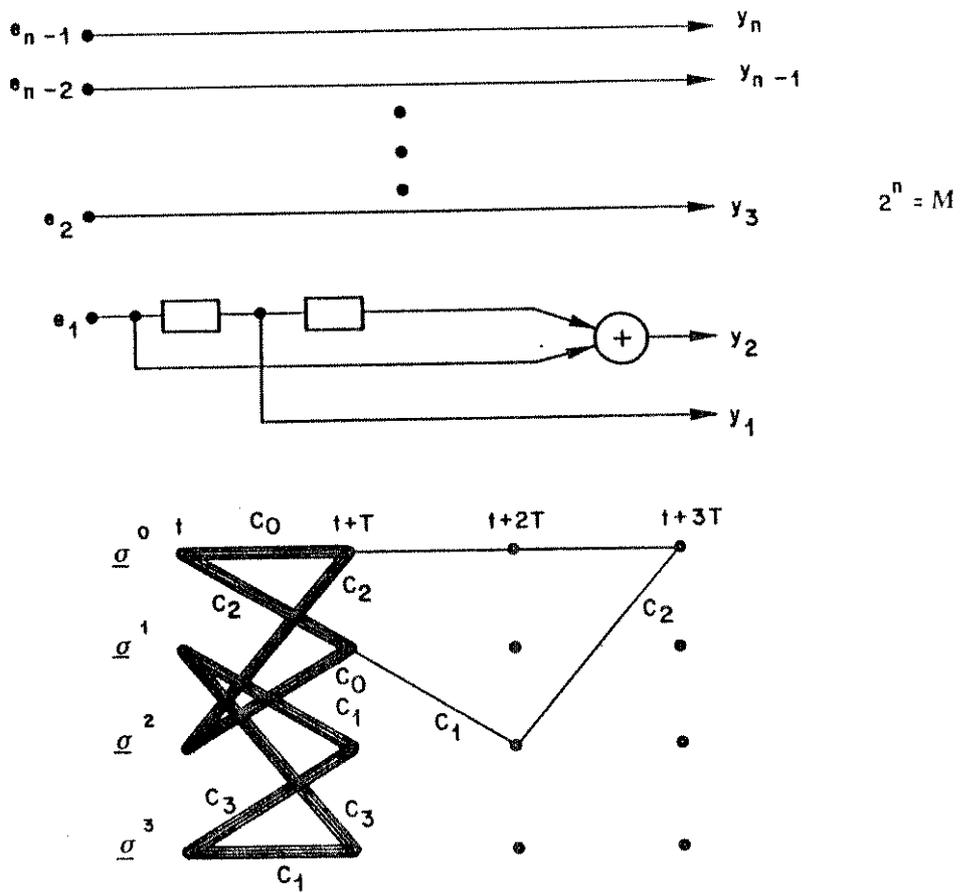


fig.1.10 Codificador convolucional ( $m=2$ ) e treliça de 4 estados ( $\sigma^i$ ) associada.

TABELA 1

m	$d_{\min}^2$	G(dB)	
		M = 16	32/64/128
2	$2\Delta_1 + \Delta_0$	3,54	3,52
3	$2\Delta_1 + 2\Delta_0$	4,01	3,98
4	$2\Delta_1 + 3\Delta_0$	4,44	4,40
5	$3\Delta_1 + \Delta_0$	5,13	5,12

#  $\Delta_0 = d^2(A)$  ;  $\Delta_1 = d^2(B_i)$ .

## 1.6 Conclusão.

O que podemos concluir é que, ficam assim caracterizadas três classes de códigos AR (M-AM, M-PSK e M-QAM), com simplicidade de implementação e razoáveis ganhos assintóticos, para diversas taxas de transmissão de informação.

C A P I T U L O

D O I S:

CONSTELAÇÕES M-PSK CODIFICADAS:

UM ESQUEMA DE RECEPÇÃO QUANTIZADA.

## 2.1 Introdução.

No capítulo um ficou caracterizada uma classe de códigos AR para modulação M-PSK com razoáveis ganhos assintóticos de codificação. Neste capítulo, nós focalizamos nossa atenção num problema prático da implementação do decodificador: a quantização das coordenadas do sinal recebido. Ao mesmo tempo, outros aspectos do processo de decodificação são analisados.

A obtenção analítica de curvas de desempenho para sistemas codificados, é praticamente impossível. Um caminho a seguir é a obtenção destas curvas (de onde se pode inferir ganhos não assintóticos de codificação) através de simulação computacional. Nós desenvolvemos um programa de simulação, com o intuito de obtermos o desempenho dos diversos esquemas de quantização analisados.

A estimação confiável de probabilidades de erro para valores abaixo de  $10^{-4}$ , iria requerer tempos computacionais excessivamente longos. O apêndice 1 descreve uma maneira de obtenção de limitantes superiores do desempenho, para recepção não quantizada. Estes limitantes se aproximam assintoticamente da curva de desempenho real.

## 2.2 Implementação do processo de decodificação.

A implementação do decodificador para códigos de treliça é feita pelo Algoritmo de Viterbi [12]. Como demonstrado por [13,15], o algoritmo é uma estimação da sequência

enviada pelo critério da máxima verossimilhança.

Nós aqui vamos nos ater a uma descrição rápida do algoritmo, como feita em [16], para podermos ressaltar as simplificações resultantes no algoritmo quando existem transições paralelas na treliça.

O algoritmo é um processo recursivo onde o caminho de menor métrica (sobrevivente) que chega a um nó (estado) da treliça, num instante  $jT$ , é armazenado. No instante  $(j+1)T$ , todos os sobreviventes anteriores são aumentados de um ramo, onde as métricas desses ramos dependem do vetor recebido e do símbolo de canal associado ao ramo. As métricas dos caminhos aumentados, que chegam a um determinado nó (no instante  $(j+1)T$ ), são comparadas, e, a menor métrica é então armazenada juntamente com o caminho sobrevivente. O processo se repete indefinidamente. O número de sobreviventes é sempre igual ao número de estados da treliça.

Sendo assim, precisamos:

- 1- Computar as métricas dos ramos a cada instante.
- 2- Adicioná-las às métricas dos sobreviventes.
- 3- Comparar as métricas destes caminhos aumentados, para determinar o novo sobrevivente em cada estado.

Para o canal AWGN, estas métricas são proporcionais a  $d^2(\underline{r}_t, \underline{s}^j)$ , onde  $\underline{r}_t$  é o vetor recebido no instante  $t$  e  $\underline{s}^j$  é o símbolo de canal considerado.

Sem perda de generalidade, considere a constelação B-PSK já particionada como mostra a figura 2.1. Qualquer vetor recebido no plano estará associado a um subconjunto de quatro sinais mais próximos, que inclui pelo menos um sinal de cada

subconjunto  $C_i$ :  $C_0, C_1, C_2$  ou  $C_3$ . Não há possibilidade de dois sinais de um mesmo subconjunto estarem contidos entre os quatro sinais mais próximos simultaneamente. Podemos observar isto da seguinte forma: para cada subconjunto de dois sinais, traçamos a mediatriz do segmento de reta que une os dois sinais. Cada mediatriz define qual dos dois sinais é o mais próximo do sinal recebido. As mediatrizes também definem setores circulares. Os quatro sinais mais próximos serão aqueles pertencentes ao setor do sinal recebido e aos setores adjacentes. Como sinais consecutivos na circunferência pertencem a subconjuntos distintos, a propriedade acima fica estabelecida.

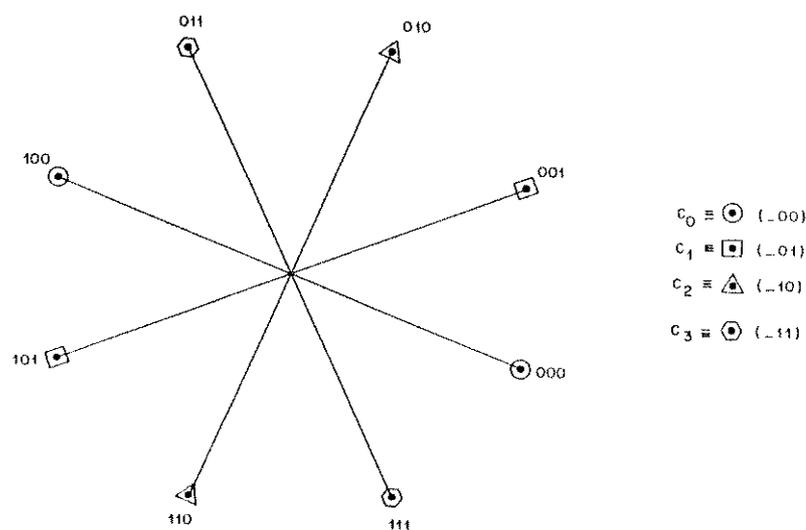


fig.2.1 Constelação B-PSK particionada e mapeamento binário.

Se estivermos num estado da treliça, com um determinado caminho sobrevivente e quisermos estender este caminho através de transições paralelas para outro estado posterior, apenas a

métrica do ramo cujo símbolo associado é um dos quatro sinais mais próximos, será relevante. Os ramos associados à sinais não próximos terão métricas maiores e seriam portanto eliminados.

O que podemos concluir, é que a presença de transições paralelas, possibilita a redução no processo de decodificação, tanto do número de métricas a serem computadas, quanto do número de adições e comparações a serem feitas. Todas as conclusões acima se generalizam facilmente para constelações M-PSK.

### 2.3 O parâmetro $R_0$ do limitante exponencial e a quantização do vetor recebido.

Uma limitação imposta pela implementação digital do decodificador é a necessidade da quantização das coordenadas do vetor recebido  $r_t$ . Cada coordenada será quantizada em um número finito de níveis se o processamento for feito através de multiplicações e adições digitais.

A quantização de  $r_t$ , converte o canal AWGN num canal discreto e sem memória ( canal DMC - do inglês: Discrete Memoryless Channel ), com  $M$  símbolos de entrada ( a constelação do codificador ) e  $J \geq M$  símbolos de saída. Cada símbolo de saída está associado a uma região ( região de decisão ) do espaço de sinais; no nosso caso, o plano.

Dizemos que o decodificador faz "decisões abruptas", se  $J = M$ , ou "decisões suaves" se  $J > M$  [19].

Em [20], mostrou-se que a probabilidade de erro média na decodificação de uma sequência de  $L$  símbolos do canal num DMC pode ser limitada por:

$$\overline{P_e} < C_r 2^{-NR_0} \quad \text{se } R < R_0,$$

onde  $N$  é função da memória  $m$  do codificador e da taxa do código ( $k/n$ ),

$C_r$  é um fator que independe de  $N$  e  $L$ , e,

$R$  é a taxa de transmissão em bits por intervalo  $T$  de símbolo de canal.

A expressão para  $R_0$  é dada por [3]:

$$R_0 = -\log_2 \left( \min_{D(\underline{g})} \sum_{j=1}^J \left[ \sum_{l=1}^M \sqrt{P(\underline{r}^j / \underline{g}^l)} D(\underline{g}^l) \right]^2 \right)$$

onde  $D(\cdot)$  é a distribuição do alfabeto de modulação ( $\underline{g}^l$ ) e  $\underline{r}^j$  é um dos  $J$  símbolos do alfabeto de saída do canal.

Baseado no argumento de que  $R_0$  é um razoável critério para o projeto de um sistema de modulação [21], Massey [3] desenvolveu um algoritmo iterativo para definir-se os níveis ótimos (no sentido de maximizar  $R_0$ ) de quantização de um DMC com  $M = 2$  e  $J \geq 2$ .

Lee [19], generalizou o algoritmo para  $M > 2$  sinais. O critério utilizado foi o de maximizar  $\tilde{R}_0$ , onde  $\tilde{R}_0$  é obtido de  $R_0$  fazendo  $D(i) = 1/M$  (para  $M=2$ , esta é a distribuição de probabilidades que maximiza  $R_0$ ). Para este critério, o limitante superior deve ter  $R_0$  substituído por  $\tilde{R}_0$  e os sinais devem ser igualmente prováveis. A condição obtida para maximização de  $R_0$  é uma condição necessária mas nem sempre suficiente. Ou seja, o algoritmo pode convergir para um ponto de máximo que não seja o máximo global. O algoritmo é baseado na transformação das regiões ótimas no espaço de sinais em regiões ótimas num "espaço

de máxima verossimilhança".

Como nós estamos interessados em definir regiões de quantização para uma constelação M-PSK, o algoritmo envolveria o cálculo de inúmeras integrações da função de densidade de probabilidade gaussiana bidimensional. Dependendo do número de símbolos de saída J, o esforço computacional seria muito grande. Mesmo assim, as regiões de decisão ótimas obtidas poderiam ser de difícil implementação real. Haveria necessidade, então, de aproximá-las.

O nosso intuito foi propor regiões de fácil implementação prática. Os resultados da simulação comprovaram mais uma vez que o problema da quantização não é tão crítico: com um número reduzido de regiões, podemos obter uma degradação de desempenho muito próxima à da quantização infinita.

#### 2.4 Quantização do vetor recebido.

Como vimos anteriormente, o setor do vetor recebido é que define os quatro sinais mais próximos. Se definirmos fases para cada sinal, por exemplo entre 0 e  $2\pi$  radianos, o vetor recebido sempre estará entre as fases de dois sinais consecutivos. Ou seja, existem M setores circulares idênticos (todos com abertura  $2\pi/M$  radianos) em que o vetor poderá ser recebido. A maneira mais conveniente de decidir qual o setor, é obter a fase do vetor recebido. Sendo assim, nos pareceu viável quantizar uniformemente as componentes de fase e amplitude do vetor recebido.

Seja  $Q_f$  o número de sub-setores produzidos em cada setor

definido pelas fases de dois sinais consecutivos. Seja  $Q_a$  o número de anéis circulares produzidos para quantizar a amplitude. Um esquema com  $Q_a = Q_f = 4$  é mostrado na figura 2.2. A região hachurada na figura é uma das  $4 \times 4 \times 8 = 128$  regiões de decisão que definem os símbolos do alfabeto de saída do DMC.

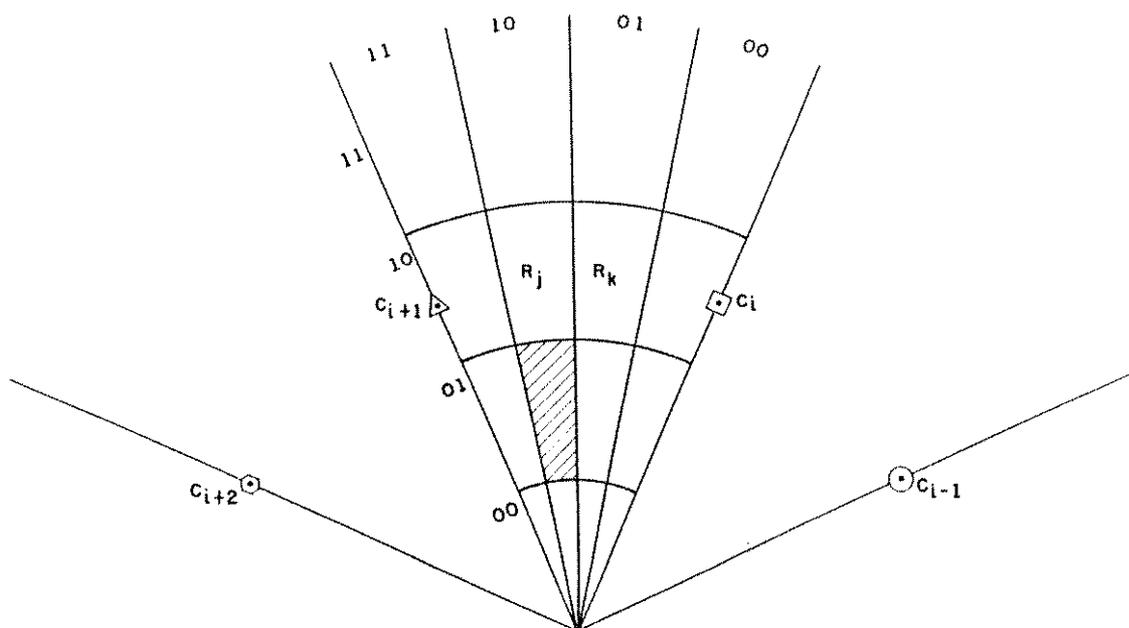


fig 2.2 Esquema de quantização  $Q_a = Q_f = 4$ .  
 A região hachurada é uma das regiões que define um dos símbolos do alfabeto de saída do DMC.  
 $R_j$  e  $R_k$  possuem métricas equivalentes (seção 2.5.1).

A quantização tão somente da componente de fase do vetor recebido, têm sido proposta na literatura [14,8]. Os níveis de limiar da componente de amplitude são efetivamente controlados pelo circuito de Controle Automático de Ganho (CAG) do receptor [22]. Os níveis de limiar da componente de fase não

sofrem influência do CAC. Sendo assim, um esquema em que somente a componente de fase fosse quantizada, seria insensível ao CAC.

Sabemos que [10] num sistema M-PSK não codificado, as regiões de decisão abrupta definidas por níveis de limiar de fase iguais a  $2\pi i/M$ ,  $0 \leq i < M$ , minimizam a probabilidade de erro de símbolo. Em [19], mostra-se que estas regiões também são ótimas no sentido de maximizar  $R_0$ . Portanto, um esquema prático de decisão suave que quantize a componente de fase do vetor recebido, parece ser um esquema conveniente, e que provavelmente necessite de um menor número de regiões de decisão para se aproximar do desempenho da quantização infinita.

## 2.5 As métricas dos ramos.

### 2.5.1 Métricas ótimas.

Como vimos na seção 2.3 a quantização do vetor recebido transformou o canal AWGN num DMC. Relembrando novamente o critério da máxima verossimilhança (seção 1.5.6),  $P(\underline{R}/\underline{S}_k)$  não é mais uma densidade de probabilidade, mas sim a probabilidade de que a sequência recebida seja  $\underline{R}$  condicionada ao envio da sequência  $\underline{S}_k$ , e pode ser expressa por:

$$P(\underline{R}/\underline{S}_k) = \prod_t P(r_t^j / s_t^i)$$

onde  $P(r_t^j / s_t^i) = P(r^j / s^i)$  são as probabilidades de transição do DMC (suposto invariante com o tempo),  $0 \leq j < J$  e  $0 \leq i < M$ .

Como a função logarítmica é monotonicamente crescente,

podemos evitar o uso de multiplicações e utilizar no algoritmo de Viterbi o logaritmo da probabilidade condicionada:

$$\log P(\underline{r}/\underline{s}_k) = \sum_t \log P(\underline{r}_t^j/\underline{s}_t^i).$$

Como maximizar a expressão acima é equivalente a minimizar o negativo da mesma, as métricas ótimas para o algoritmo de Viterbi serão:  $-\log P(\underline{r}^j/\underline{s}^i)$ .

Para o esquema de quantização polar considerado, as probabilidades de transição do DMC, podem ser expressas por:

$$P(\underline{r}^j/\underline{s}^i) = \iint_{R_j} (1/\pi N_0) \exp[-(x-s_1^i)^2 + (y-s_2^i)^2/N_0] dx dy$$

onde  $\underline{s}^i = (s_1^i, s_2^i)$  e  $R_j$  é a região de decisão do símbolo  $\underline{r}^j$ .

Fazendo uma transformação para coordenadas polares (mais adequadas ao esquema polar proposto):

$$P(\underline{r}^j/\underline{s}^i) = (G/\pi) \int_{r_{INF}/A}^{r_{SUP}/A} \int_{\theta_{INF}}^{\theta_{SUP}} \exp[-G(r^2 - 2r \cos(\theta - \theta_1) + 1)] r dr d\theta, \quad (2.1)$$

onde  $\underline{s}^i = (A, \theta_1)$ ,  $G = A^2/N_0 = E_s/N_0$  é proporcional a  $E_b/N_0$ , e,  $r_{SUP}$ ,  $r_{INF}$ ,  $\theta_{SUP}$ ,  $\theta_{INF}$  definem a região de decisão  $R_j$ .

Como sempre estaremos definindo  $r_{INF}$  e  $r_{SUP}$ , proporcionalmente a "A", podemos observar que as métricas ótimas serão função da relação sinal ruído e da localização das regiões de decisão.

Há um ponto importante a salientar. Para os esquemas de quantização propostos, o DMC resultante e suas métricas ótimas não alteram a maneira de obtenção dos quatro sinais relevantes no processo de decodificação. Todas as conclusões da seção 2.2

continuam válidas.

Passemos agora ao problema do armazenamento das métricas dos ramos. Como só há quatro sinais relevantes e um setor circular resultante da determinação desse conjunto de sinais mais "próximos", somente as métricas das regiões de decisão deste setor, relativas aos quatro sinais, serão armazenadas. A simetria das constelações M-PSK diminui ainda mais o número de métricas a serem armazenadas. Considere novamente um esquema com  $Q_a=Q_f=4$  e observe a figura 2.2. Será necessário armazenar apenas  $4 \times 8 = 32$  métricas correspondentes às regiões de decisão de metade (determinada pela bissetriz) do setor. As métricas da outra metade serão equivalentes. Isto será válido para todos os esquemas de quantização por decisão suave considerados.

Para podermos minimizar o número de bits que representarão estas métricas precisamos torná-las números inteiros. Para isto, precisamos escaloná-las e depois arredondá-las. Note que o escalonamento das métricas não altera o critério da máxima verossimilhança (maximizar  $x$  é equivalente a maximizar  $Kx$ ,  $K > 0$ ). O arredondamento das métricas (números reais) irá torná-las sub-ótimas.

Uma propriedade importante das métricas é que :

- Para uma determinada região de decisão, o desempenho do sistema somente é afetado pelo conjunto de diferenças entre os valores das métricas associadas àquela região de decisão. O

valor específico das métricas não é relevante.

Isto pode ser visto da seguinte forma:

- Ao compararmos duas sequências quaisquer do código, as métricas do instante  $t$  serão as métricas de uma região de decisão  $R_j$  em relação à dois sinais da constelação. Se mantivermos a diferença entre estas métricas, o resultado do critério da máxima verossimilhança não é alterado (estamos subtraindo um mesmo fator dos valores das métricas das sequências comparadas). Portanto, se mantivermos todas as diferenças entre as métricas de  $R_j$ , o desempenho do sistema não será alterado.

Esta propriedade é uma generalização da propriedade descrita em [23] para uma constelação unidimensional 2-PSK.

Vamos continuar agora com o exemplo do esquema  $Q_a=Q_f=4$  (figura 2.2). A figura 2.3 mostra as métricas ótimas para uma relação sinal ruído de 5dB e um espaçamento entre os limiares de amplitude igual a 0,4 ( $\lambda = 1$  na equação 2.1). As métricas com relação aos símbolos  $C_{i+1}$  e  $C_{i+2}$  são equivalentes. Suponha que quizéssomos utilizar apenas 5 bits para representar estas métricas. O primeiro passo é escaloná-las de maneira que ao arredondarmos as métricas escalonadas, os números inteiros obtidos possam se diferenciar proporcionalmente aos números reais das métricas ótimas. Por exemplo: podemos multiplicar por um fator de 10 e arredondar considerando a primeira casa decimal. O segundo passo é aplicar a propriedade descrita acima. E finalmente um terceiro passo seria associar aos inteiros obtidos que ultrapassem 5 bits para sua representação, o valor 31. Se seguirmos estes passos obtemos a tabela de métricas mostrada na figura 2.4. Nós simulamos este esquema para um código 8-PSK com

SÍMBOLO:  $C_{i-1}$

Qa \ Qf	00	01	10	11
00	3,4942	2,6783	2,7621	3,6402
01	3,7581	3,2225	3,5947	4,7791
10	4,0371	3,8044	4,5050	6,0397
11	4,3139	4,3959	5,4570	7,3757

SÍMBOLO:  $C_i$

Qa \ Qf	00	01	10	11
00	2,8757	1,4124	0,8700	1,0088
01	2,9431	1,5500	1,0737	1,3621
10	3,0742	1,8178	1,4712	1,8959
11	3,2615	2,2011	2,0431	2,6661

fig. 2.3 Métricas ótimas:  $-\log_{10} P(\underline{r}^i/\underline{g}^i)$   
 $Q_a = Q_f = 4$ . Espaçamento entre limiares de amplitude: 0,4. Relação sinal ruído  $E_b/N_0 = 5$  dB.

SÍMBOLO:  $C_{i-1}$

	Qa	00	01	10	11
Qf					
00		6	13	19	25
01		9	16	25	31
10		11	22	31	31
11		14	30	31	31

SÍMBOLO:  $C_i$

	Qa	00	01	10	11
Qf					
00		0	0	0	0
01		0	0	0	0
10		2	2	4	5
11		4	8	11	16

Fig. 2.4 Métricas sub-ótimas para o caso da figura 2.3. Cinco bits para representação.

memória  $m=2$ , utilizando 5 bits para representação das métricas. A degradação do desempenho, em relação às métricas ótimas, se mostrou desprezível. Este resultado concorda com resultados anteriores para códigos convolucionais binários que utilizam modulação 2-PSK [23,22]. As curvas da simulação serão apresentadas na seção 2.7.

Finalmente, convém lembrar a influência do número de bits das métricas dos ramos no número de bits necessários para armazenar as sequências sobreviventes no algoritmo de Viterbi. Se a cada instante o estado de menor métrica é levado a ter métrica zero (e os outros estados são normalizados na mesma proporção), o número de bits necessários para armazenar um sobrevivente não excede  $m \times M(R_j)$ , onde  $m$  é a memória do código e  $M(R_j)$  é a maior métrica (inteira) associada a um ramo [14,23]. Daí a importância da minimização do número de bits das métricas dos ramos (apesar de que, quase sempre, pode-se diminuir o valor do número de bits que armazenam um sobrevivente para um valor bem abaixo de  $m \times M(R_j)$  [23]).

#### 2.5.2 Métricas quantizadas.

Há duas desvantagens no uso de métricas ótimas [23]:

- 1) Elas necessitam do conhecimento das probabilidades de transição do canal.
- 2) Elas são dependentes da relação sinal ruído.

Para o AWGN, as métricas ótimas são obtidas da

expressão:

$$d^2(\underline{R}, \underline{S}_k) = \sum_t d^2(\underline{r}_t, \underline{s}_t^i),$$

$$\text{onde } d^2(\underline{r}_t, \underline{s}_t^i) = |\underline{r}_t|^2 - 2 \underline{r}_t \cdot \underline{s}_t^i + |\underline{s}_t^i|^2.$$

Como o valor de  $|\underline{r}_t|$ , é o mesmo para qualquer sequência  $\underline{S}_k$ , e todos os sinais tem mesma energia,  $|\underline{s}_t^i|^2 = E_s$ , as métricas dependerão apenas dos termos dos produtos internos:

$$\underline{r}_t \cdot \underline{s}_t^i = |\underline{r}_t| \cos(\theta_r - \theta_i), \text{ onde } \theta_r \text{ é a componente de fase de } \underline{r}_t \text{ e } \theta_i \text{ é a fase do sinal considerado.}$$

Assim sendo, podemos definir uma métrica quantizada, em função das componentes de fase e amplitude de  $r$ . Definiremos um raio médio e um ângulo médio. Isto é:

$$r_M^l = (r_l + r_{l+1})/2, \quad 0 \leq l < Qa, \text{ e,}$$

$$\theta_M^m = (\theta_n + \theta_{n+1})/2, \quad 0 \leq n < Qf.$$

Como mostra a figura 2.5,  $r_l$ ,  $r_{l+1}$ ,  $\theta_n$ ,  $\theta_{n+1}$ , são os raios e ângulos que definem uma dada região de decisão. Assumimos:  $r_{Qa} = r_{Qa-1} + \Delta r$ , onde  $\Delta r$  é o espaçamento entre os limiares de decisão da componente de amplitude. Então, associamos a cada região de decisão uma métrica em função do raio e ângulo médios:

$$r_M^l \cos(\theta_M^m - \theta_i),$$

onde  $\theta_i$  é a fase do sinal considerado.

Todas as considerações feitas na seção 2.5.1 para as métricas ótimas, continuam válidas para as métricas quantizadas propostas.

No apêndice 2, nós descrevemos simulações feitas para o código 8-PSK ( $m=2$ ), utilizando métricas quantizadas. No capítulo 3, nós descrevemos resultados obtidos para o canal de satélite

com a utilização de métricas quantizadas.

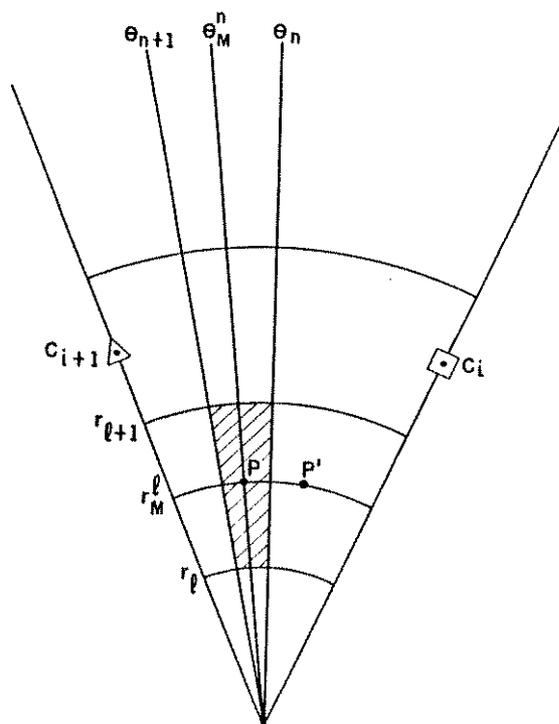


fig. 2.5 Definição do raio médio e do ângulo médio.  
 $Q_a = Q_f = 4$ .  $P$  e  $P'$  possuem métricas equivalentes.

## 2.6 Considerações sobre o programa de simulação.

A figura 2.6 mostra a representação, em diagrama de blocos, do programa de simulação computacional que avalia o desempenho dos esquemas de quantização propostos.

Os blocos de pré-processamento, quantização, associação de métricas e algoritmo de Viterbi fazem o papel do decodificador da figura 1.2. A subrotina que simula o algoritmo foi escrita

baseada em idéias contidas em [24].

Iremos nas seções que se seguem, fazer considerações sobre os diversos blocos.

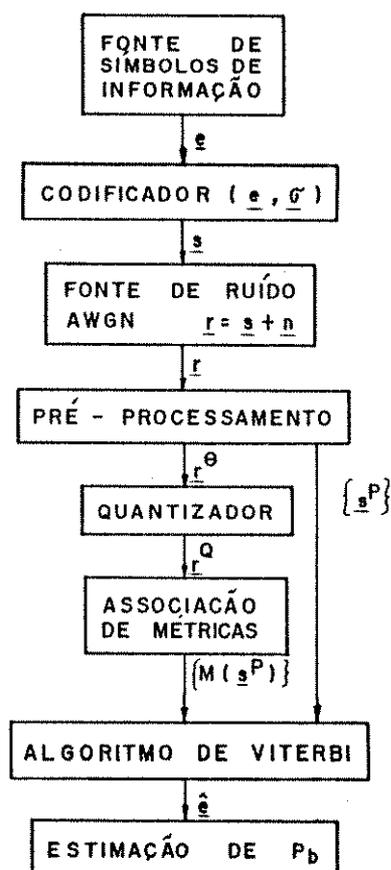


fig. 2.6 Diagrama de blocos do programa de simulação.

### 2.6.1 Estimação da probabilidade de erro de bit.

Até agora nós utilizamos dois conceitos de probabilidade de erro: probabilidade de erro de sequência, no critério da máxima verossimilhança; e, probabilidade de evento de erro, na procura dos códigos AR ótimos.

A probabilidade de erro de bit é definida como o número

de bits em erro numa dada sequência de bits de informação recebida, normalizado pelo número total de bits da sequência [14]. A probabilidade de erro de bit é geralmente escolhida como a medida de desempenho de um código.

### 2.6.1.1 Intervalos de Confiança.

A obtenção de uma estimativa da probabilidade de erro de bit, é feita pela contagem dos erros que ocorreram ao compararmos a sequência recebida com a enviada.

Se  $X_i$  é uma variável aleatória que assume o valor unitário quando ocorre um erro, e o valor nulo caso contrário, a probabilidade de erro de bit é estimada pela razão

$$\hat{P}_b = \sum_{i=1}^{kN} X_i / kN,$$

onde  $N$  é o número de símbolos de informação transmitidos, e,  $k$  é o número de bits de informação por símbolo.

Suponha que o sistema não é codificado e que assumimos um mapeamento Gray. Para relações sinal ruído altas, a cada erro de símbolo corresponderá aproximadamente um erro de bit. Então, podemos escrever:  $P_b \cong P_o/k$ , onde  $P_o$  é a probabilidade de erro de símbolo. Podemos considerar que a estimativa  $P_b$  terá uma distribuição binomial de média  $P_b$  e variância  $\hat{\sigma}_{P_b}^2 = P_b(1-P_b)/kN$ .

Poderíamos perguntar: Qual a probabilidade de que  $|\hat{P}_b - P_b|$  seja menor ou igual a um  $vP_b$  ?

Sabemos do Teorema Central do Limite [25] que,

$[(\hat{P}_b - P_b) \sqrt{kN} / \hat{\sigma}_{P_b}]$  tende a uma densidade de probabilidade gaussiana, com média zero e variância unitária, quando  $N$  tende a infinito. Assim,

$$P(|\hat{P}_b - P_b| \leq vP_b) \cong 1 - 2Q(b) = a,$$

onde,  $P(\cdot)$  denota probabilidade,  $b = vP_b/\hat{\sigma}_{P_b}$ , e,  $Q(\cdot)$  é a função gaussiana de probabilidade de erro.

Dizemos que  $[(1-v)P_b, (1+v)P_b]$  é um intervalo de  $(100a)\%$  de confiança [25]. Por exemplo: para  $P_b=10^{-4}$ , 90% de confiança e  $kN = 200.000$ ,  $v$  será aproximadamente 0,4.

Por outro lado, quando a transmissão é codificada a distribuição dos erros irá depender da densidade de probabilidade conjunta do ruído e das sequências particulares de símbolos. Fica difícil agora definir um intervalo de confiança. O critério adotado foi o de observar o valor de  $\hat{P}_b$  a cada  $N=20.000$  símbolos de canal. Os resultados das primeiras simulações mostraram que para  $N$  próximo de 100.000 o valor de  $\hat{P}_b$  não varia o seu primeiro dígito significativo. A região de probabilidades observadas foi de  $10^{-4} < P_b < 10^{-4}$ . Não foram verificados resultados inconsistentes à medida que pontos das curvas em relações sinal ruído mais baixas eram estimados.

Vale ressaltar que em simulação estamos sempre utilizando rotinas pseudo-aleatórias de ruído. É preciso, inicialmente, conhecer a precisão destas rotinas.

#### 2.6.2 Fonte de símbolos de informação e Fonte do ruído.

Para a obtenção dos símbolos de informação, utilizamos a função intrínseca RAN(X) (FORTRAN do sistema DEC-10 Digital) que gera números pseudo-aleatórios reais uniformemente distri-

buscados no intervalo (0,1). Através de uma transformação, podemos obter números inteiros num intervalo qualquer [A,B].

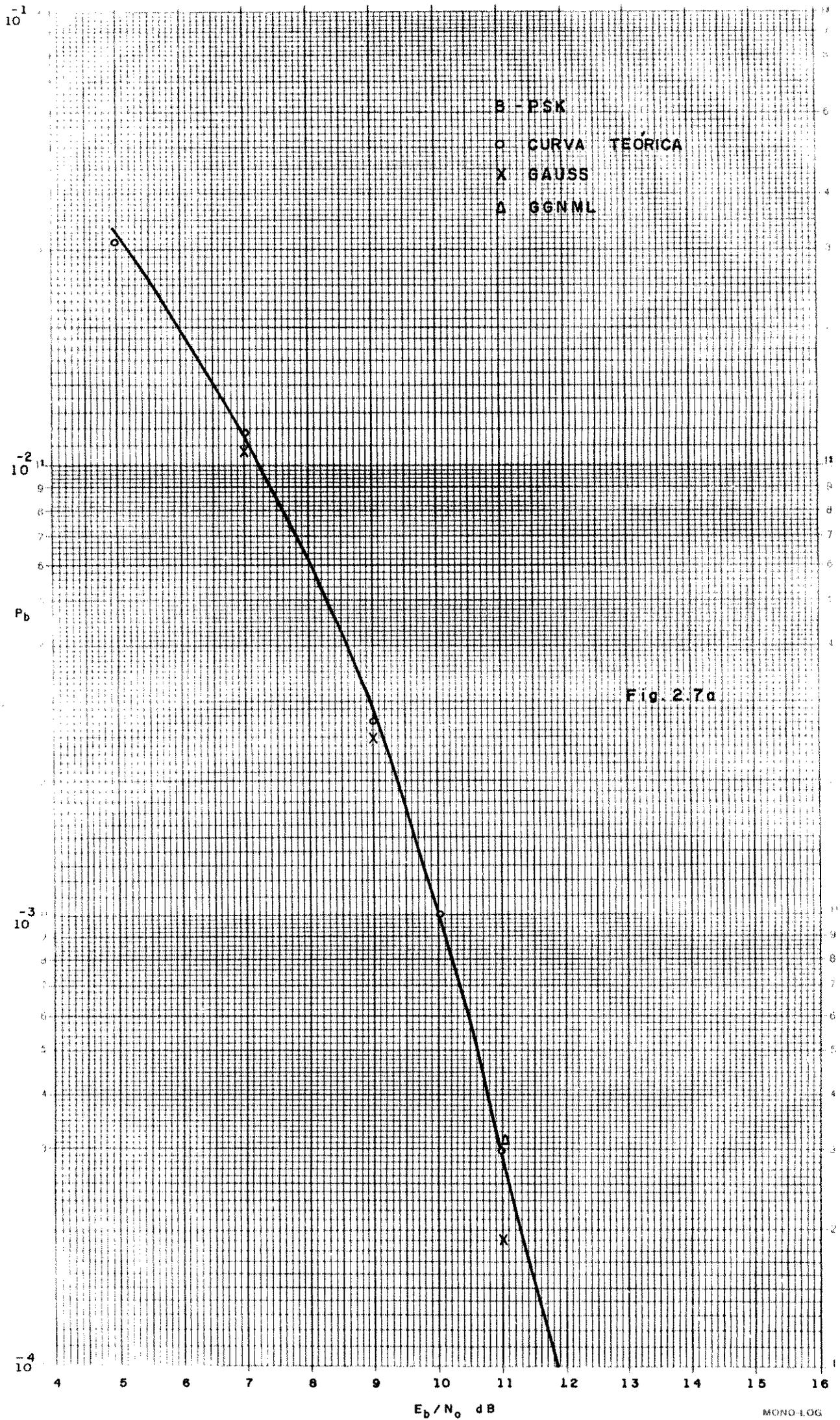
A geração do ruído foi feita, para as curvas B-PSK, através da subrotina GAUSS (Biblioteca IBM-FORTRAN), que gera amostras de uma densidade de probabilidade gaussiana. Esta subrotina baseia-se no Teorema Central do Limite, utilizando amostras da função RAN(X). Posteriormente (M=16 e 32), utilizamos a subrotina GGNML (Biblioteca IMSL-FORTRAN) para a geração do ruído. Os resultados de simulação dos sistemas não codificados são apresentados nas figuras 2.7a, 2.7b e 2.7c. A subrotina GGNML demonstrou uma precisão melhor em relações sinal ruído mais altas. As curvas teóricas das constelações não codificadas foram obtidas em [26].

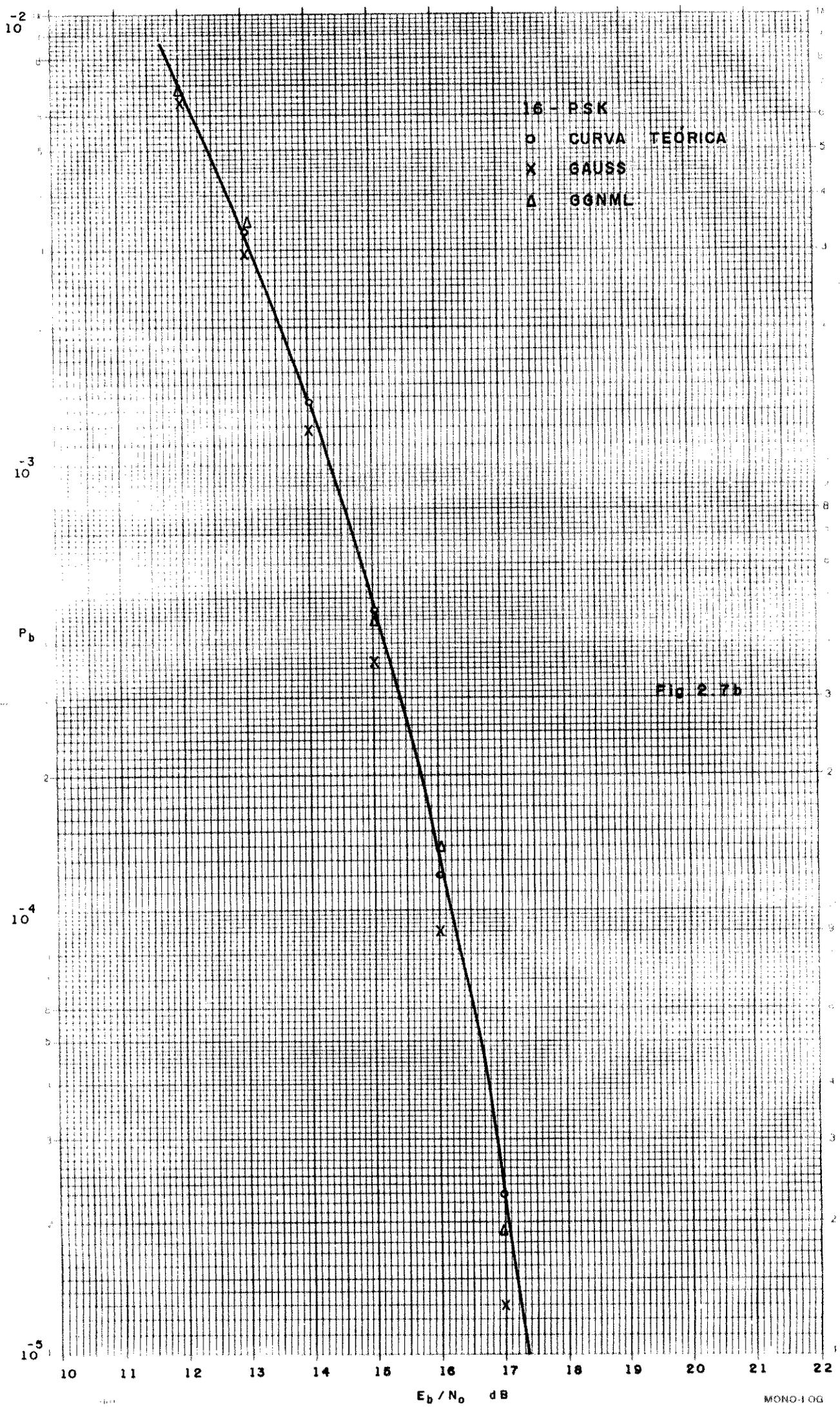
### 2.6.3 Pré-processamento.

É o estágio que transforma as coordenadas retangulares do vetor recebido  $\underline{r}$  em coordenadas polares  $\underline{r}^\theta = (R, \theta)$ , e determina o conjunto  $\{\underline{s}^p\}$  que contém os quatro sinais mais próximos (seção 2.2).

### 2.6.4 Quantizador e associação das métricas.

O quantizador define a região de decisão do vetor recebido  $\underline{r}^\theta$ . O número de regiões de decisão dependerá de  $Q_a$  e  $Q_f$ . Se a simulação for com recepção não quantizada ( $Q_a=Q_f=\infty$ ),  $\underline{r}^a=\underline{r}^\theta$ . A associação das métricas  $\{M(\underline{s}^p)\}$  é feita através de uma





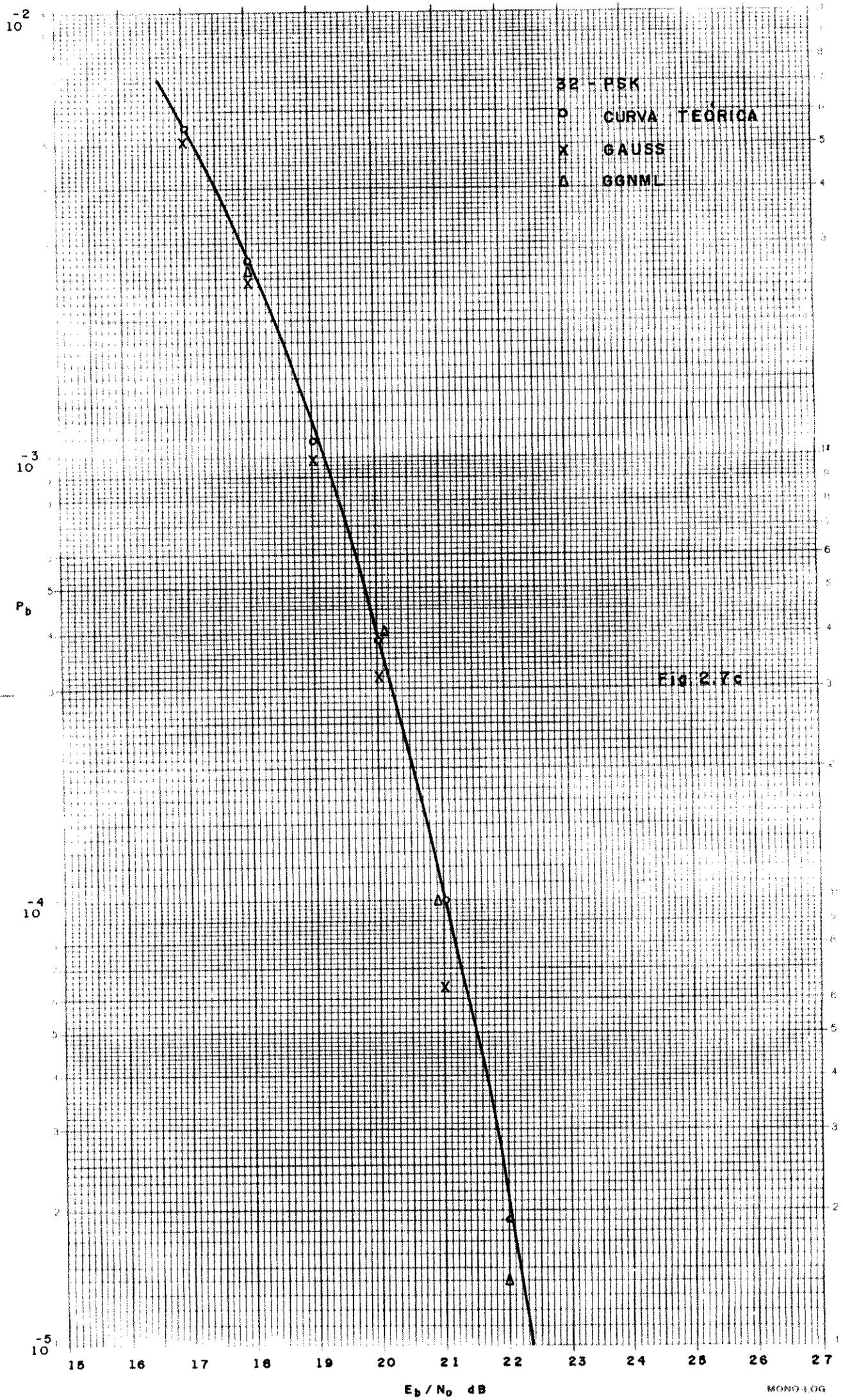


tabela de quantização como descrito na seção 2.5.

### 2.6.5 Algoritmo de Viterbi.

#### 2.6.5.1 Estado inicial do decodificador.

O algoritmo de Viterbi pode ser iniciado sem conhecimento prévio do estado inicial do codificador. Podemos, então, associar a todos os estados uma métrica de valor nulo. No início, durante vários ramos, o desempenho do algoritmo não será satisfatório. Depois de alguns ramos (um múltiplo pequeno do tamanho  $m$  da memória), o caminho correto irá passar a ter a menor métrica [14]. Para efeito de simulação, preferimos fazer com que o estado inicial fosse conhecido. Isto é obtido se fizermos a métrica inicial do estado conhecido igual a zero e as métricas dos outros estados um valor "infinito".

#### 2.6.5.2 Comprimento de Decodificação (CD).

Um caminho na treliça que divergiu do caminho correto, irá acumular uma métrica em relação ao mesmo, que aumenta com o aumento do número de ramos divergentes. Ao convergir novamente com o caminho correto, um caminho que divergiu durante muitos ramos, terá uma probabilidade pequena de ter uma métrica menor que a do caminho correto. Então, existe uma grande probabilidade de que todos os sobreviventes no instante  $t$ , passem pelos mesmos nós até um instante  $t-JT$ , se  $J$  for escolhido de maneira suficientemente grande. Simulações têm mostrado [22,17], que um valor de  $J$ , chamado de comprimento de decodificação (CD), de 5-6 vezes o tamanho da memória total do codificador, apresenta um

efeito desprezível no desempenho do código.

Sendo assim, a cada novo conjunto de  $k$  bits que entram no codificador, nós fazemos uma decisão sobre os  $k$  bits que entraram CD ramos anteriormente. A decisão sobre estes  $k$  bits pode ser feita usando-se vários critérios [13]. Podemos decidir pelo conjunto de bits que são maioria entre todos os estados, ou escolher arbitrariamente qualquer estado, ou ainda escolher os bits pertencentes ao sobrevivente de menor métrica.

O programa de simulação desenvolvido tem capacidade para podermos variar o comprimento de decodificação e observar o seu efeito na degradação do desempenho. O critério de decisão utilizado foi o de escolher os bits iniciais do caminho sobrevivente de menor métrica. Se houver igualdade entre as métricas, descartamos um dos sobreviventes de maneira a sempre conservar o obtido na comparação anterior.

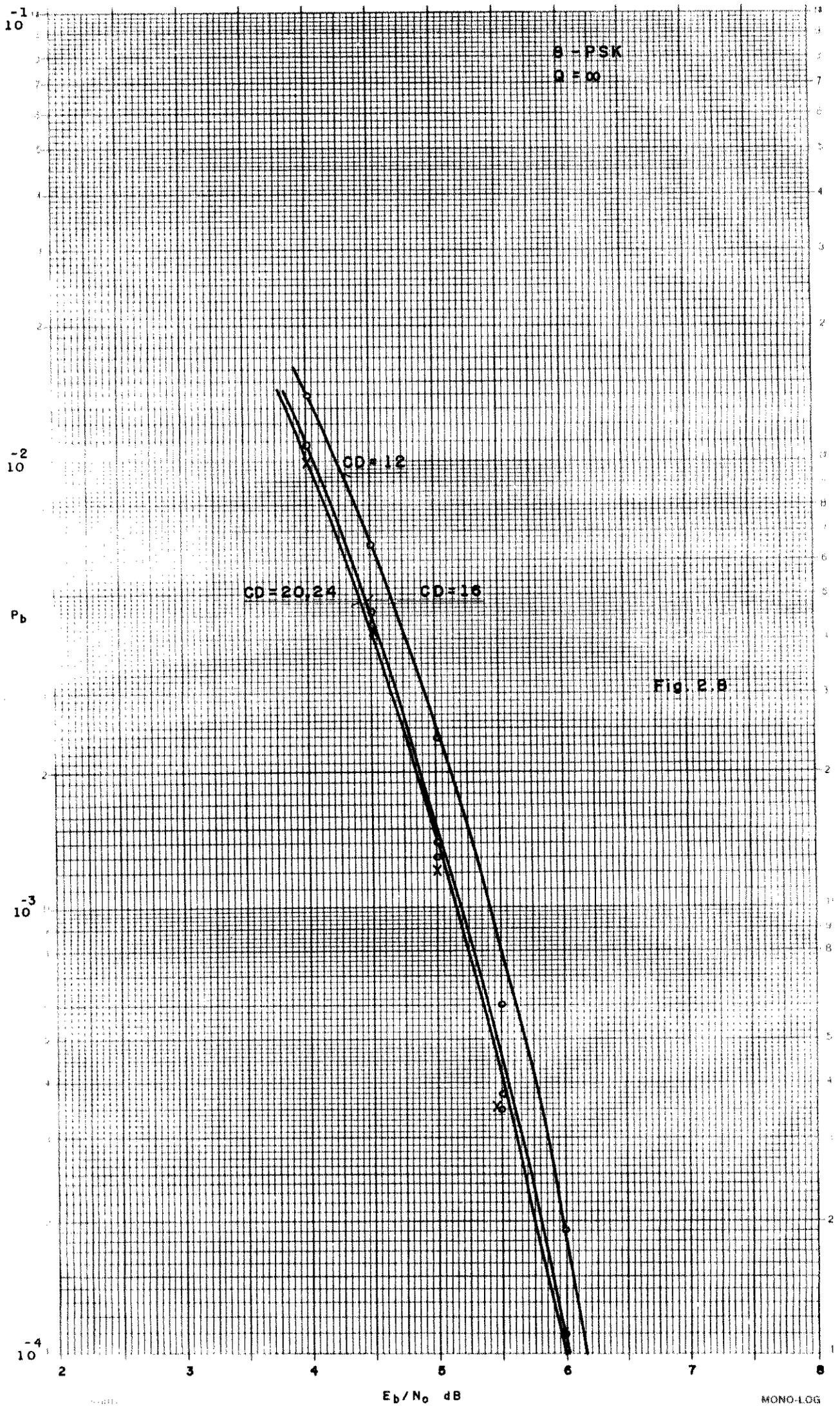
## 2.7 Resultados da simulação.

Todas as simulações foram feitas para códigos com memória  $m=2$ .

Para o caso da constelação 8-PSK codificada, o ganho assintótico de codificação é de aproximadamente 3 dB. Para constelações com  $M > 8$ , o ganho é de aproximadamente 3,5 dB (tabela 1, capítulo um).

### 2.7.1 A constelação 8-PSK.

A figura 2.8 mostra a variação do desempenho como



função do comprimento de decodificação para um esquema de recepção não quantizada (denotaremos este esquema por  $Q = \infty$ ). O que podemos observar na figura é uma "lei de diminuição do retorno" [18]. Isto é, a cada aumento de 4 ramos no comprimento de decodificação, o ganho adicional em dB é cada vez menor (de 12 para 16,  $\cong 0,2$  dB; de 16 para 20,  $\cong 0,05$  dB). Nós fixamos, então, o comprimento de decodificação em 20 para todas as simulações posteriores. O resultado obtido para  $CD=12$  está em concordância com o resultado da referência [2].

#### 2.7.1.1 Quantização da componente de fase do vetor recebido.

A figura 2.9 mostra a variação do desempenho como função da quantização da componente de fase ( $Q_a=1, Q_f=2,4,8,16$ ). O esquema com  $Q_a=1$  e  $Q_f=2$  (que define  $J=2M$  regiões de decisão) tem uma degradação menor que 0,9 dB em relação à  $Q=\infty$  ( $P_b=10^{-3}$ ). Novamente aqui observamos uma lei de 'diminuição do retorno. Para  $P_b = 10^{-3}$ , o esquema com  $Q_a=1$  e  $Q_f=4$  tem uma degradação menor que 0,5 dB em relação à  $Q=\infty$ ; e o esquema com  $Q_a=1$  e  $Q_f=8$ , menor que 0,4 dB. Um esquema com  $Q_f=16$  nos traz um ganho adicional menor que 0,1 dB.

#### 2.7.1.2 Quantização das componentes de fase e amplitude do vetor recebido.

Ao propor um esquema que quantize uniformemente a amplitude do vetor recebido, uma questão a ser respondida é sobre o espaçamento entre os limiares de decisão de amplitude,  $\Delta r$ . Para um dado esquema, qual será o espaçamento ótimo, no sentido

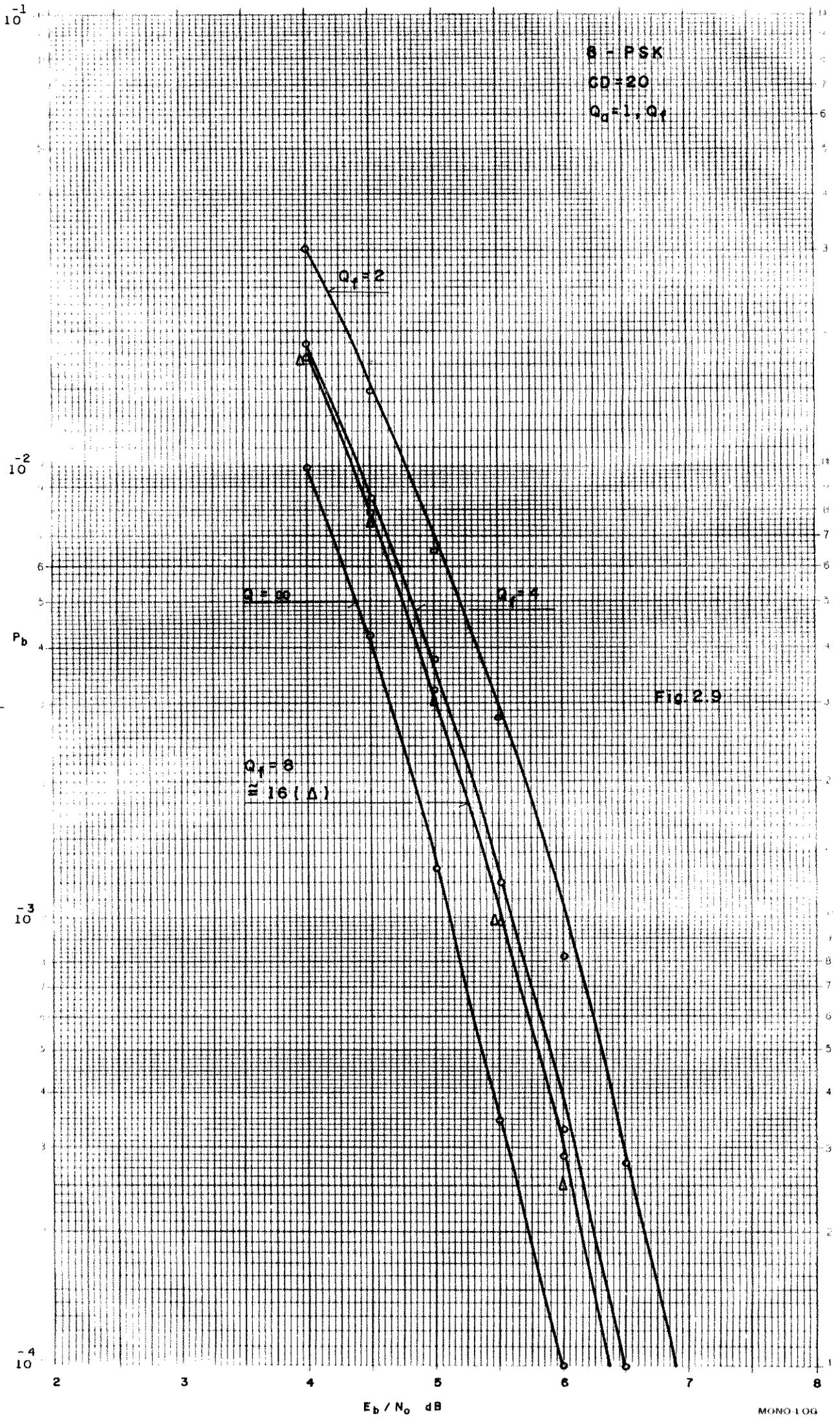


Fig. 2.9

de minimizar a probabilidade de erro ?

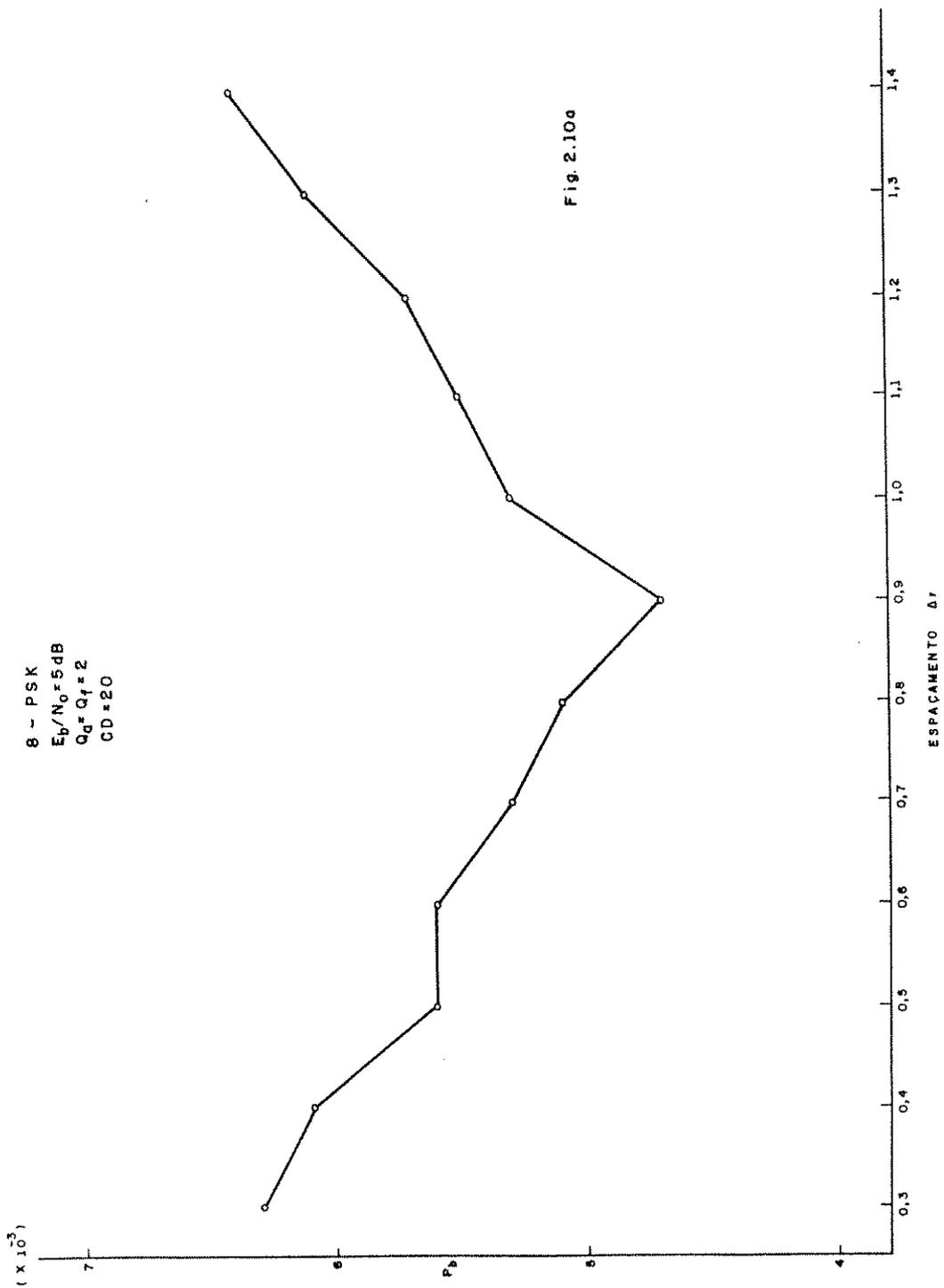
As curvas das figuras 2.10a (para um esquema com  $Q_a=Q_f=2$ ), 2.10b ( $Q_a=Q_f=4$ ), 2.10c ( $Q_a=2, Q_f=8$ ) e 2.10.d ( $Q_a=4, Q_f=8$ ), respondem a esta questão. Todas foram obtidas para um valor de  $E_b/N_0 = 5$  dB, o que corresponde a uma região de probabilidades em torno de  $10^{-3}$ .

Como tínhamos mencionado na seção 2.4, o CAG do receptor controla a precisão dos limiares de decisão da componente de amplitude. O efeito de uma imprecisão do sinal de CAG é o equivalente a uma variação no valor de  $\Delta r$ . Se  $r_i$  e  $r_{i+1}$  são dois limiares consecutivos, então,  $\Delta r = r_{i+1} - r_i$ . Se  $r_i = K_i A$  e  $r_{i+1} = K_{i+1} A$ , uma variação percentual no valor de  $A$ , corresponde a uma mesma variação percentual no valor de  $\Delta r$ .

O que podemos inferir por observação das figuras 2.10, é que os esquemas são pouco sensíveis ao CAG do receptor. O valor da probabilidade de erro mínima é, praticamente inalterado para uma variação de 25% do CAG no esquema  $Q_a=4, Q_f=8$ . O mesmo acontece para um esquema  $Q_a=Q_f=4$ . Para um esquema  $Q_a=2, Q_f=8$  a variação é de 23% (estamos considerando  $\Delta r = 0,65$  como sendo o ponto de operação). E finalmente, o esquema  $Q_a=Q_f=2$  suporta 11% de variação no CAG.

A robustez do desempenho dos esquemas de quantização polar, em relação à uma variação no CAG, concorda com resultados obtidos em [22] para uma dimensão, e em [7] para duas dimensões, quando foi utilizado um esquema de quantização das componentes de fase e quadratura (coordenadas retangulares) do vetor recebido.

A figura 2.11 mostra os resultados das simulações para



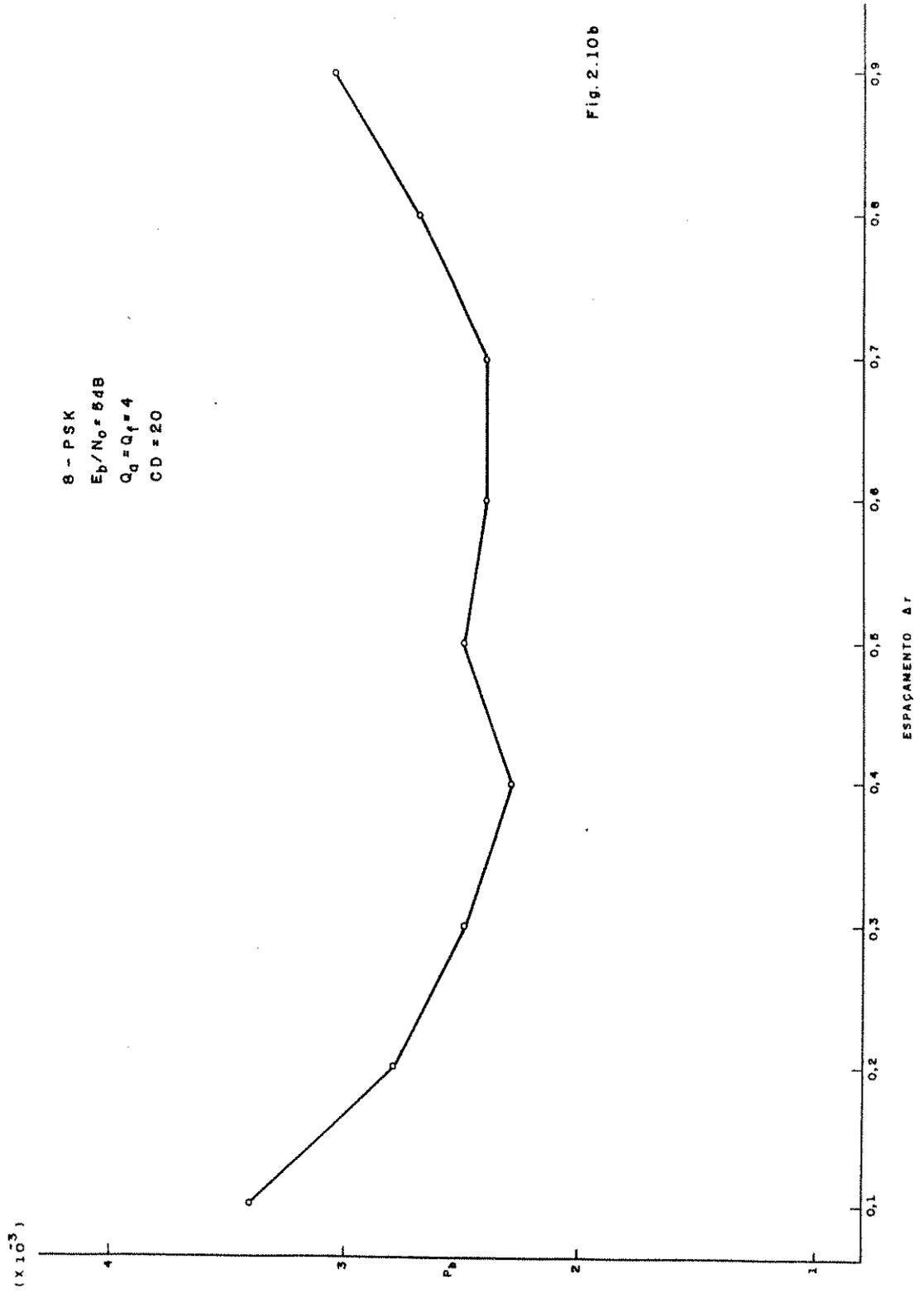


Fig. 2.10b

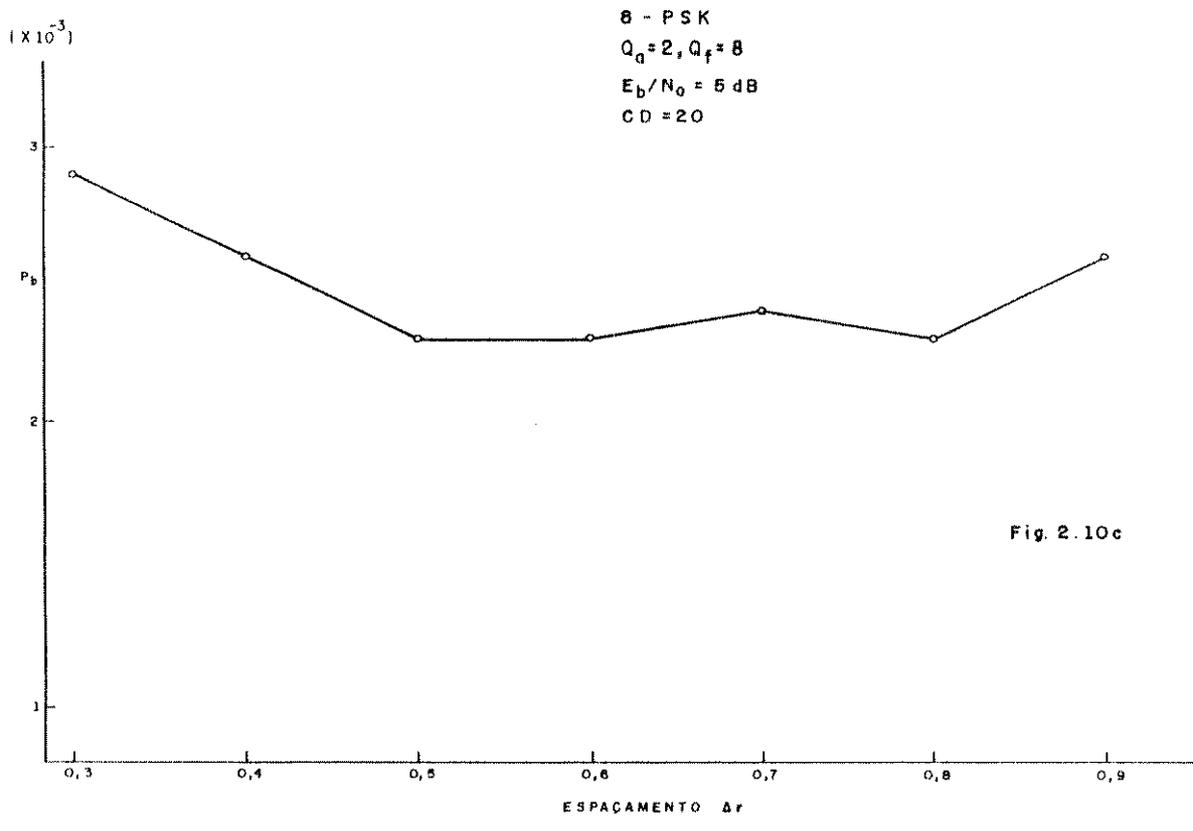


Fig. 2.10c

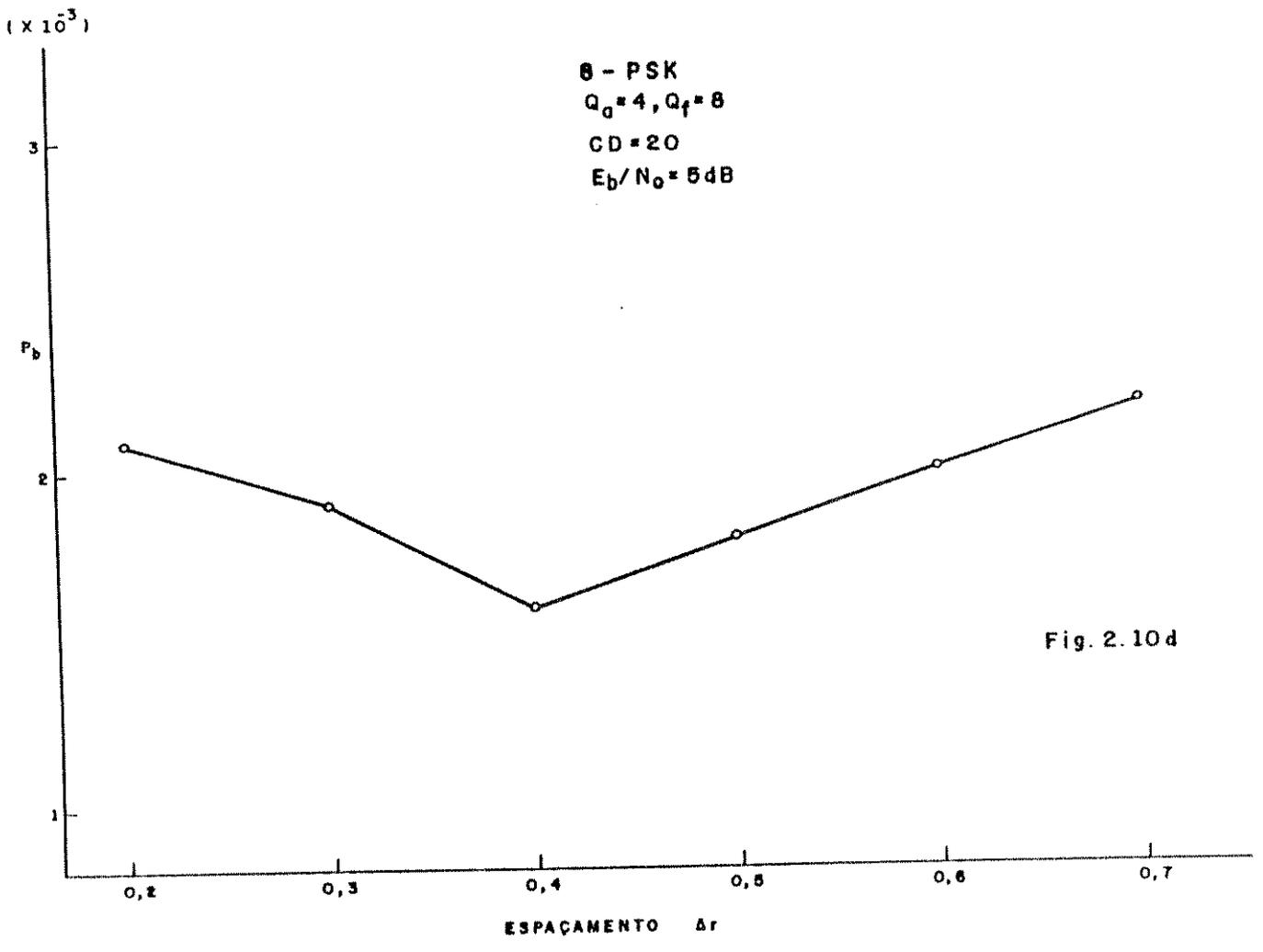


Fig. 2.10d

$10^{-1}$

$10^{-2}$

$10^{-3}$

$10^{-4}$

$P_b$

B = PSK

GD = 20

$\Delta_f$  OTIMOS

$Q_d = Q_f = 2$

$Q = \infty$

$Q_d = 4, Q_f = 8$

$Q_d = Q_f = 4$  (□)

$Q_d = 2, Q_f = 8$  (Δ)

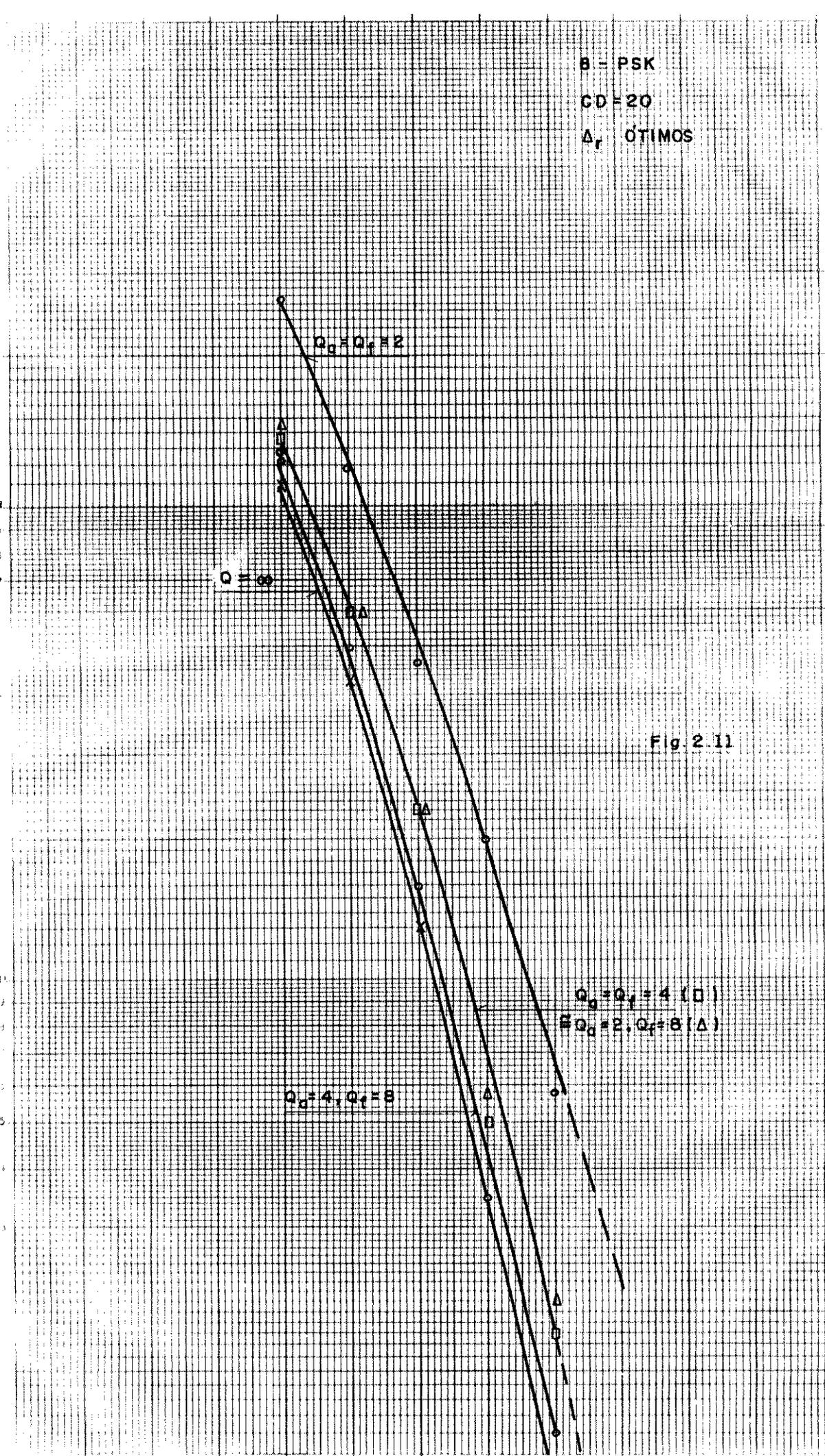
BA

CA

DA

Fig. 2.11

$E_b / N_0$  dB



os valores de  $\Delta r$  ótimos. Aqui podemos observar que o esquema com  $Q_a=4$  e  $Q_f=8$  tem um desempenho muito próximo da quantização infinita (degradação menor que 0,05 dB).

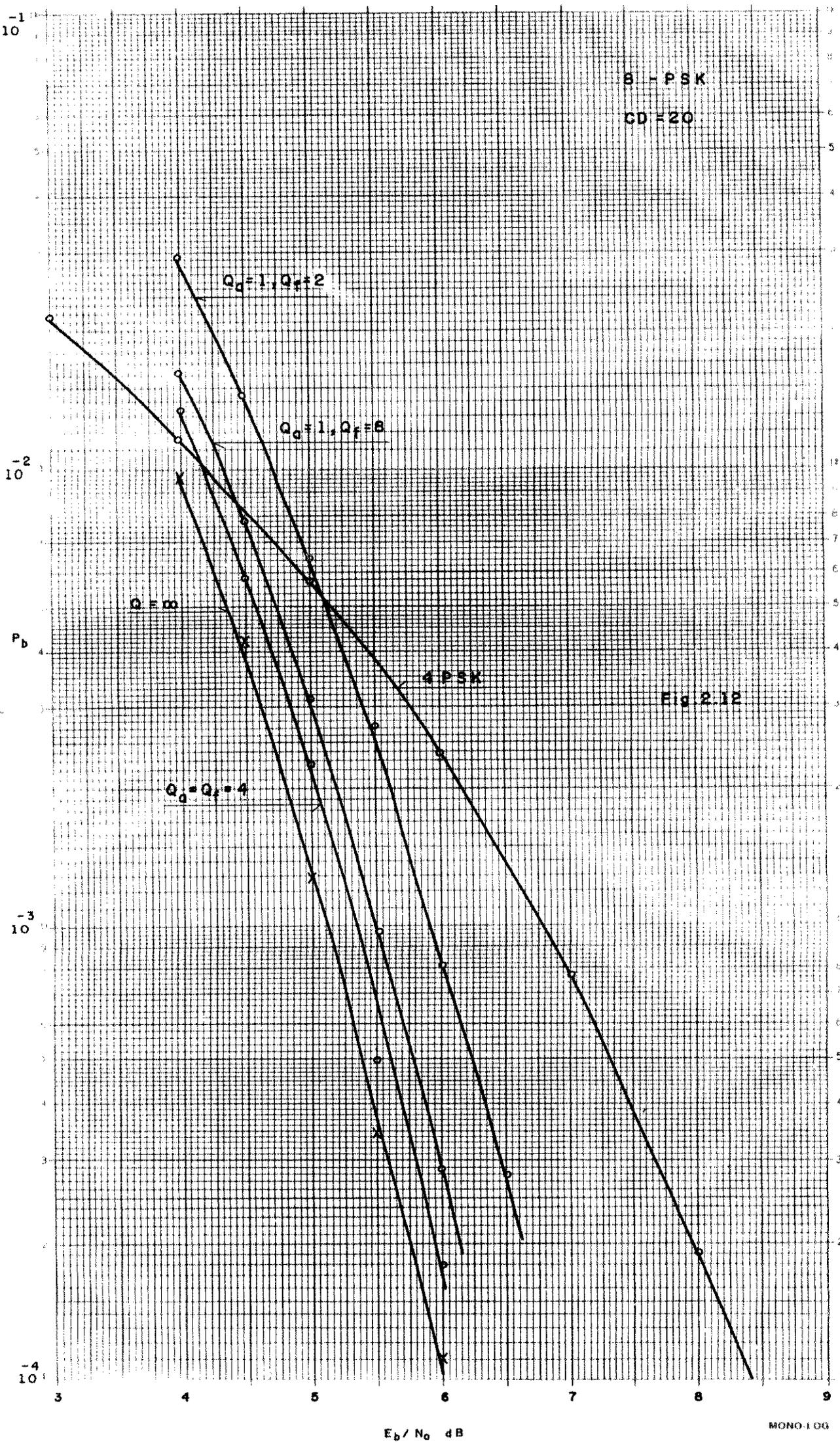
A figura 2.12 faz uma comparação de alguns dos esquemas descritos anteriormente. A curva teórica do sistema 4-PSK não codificado [26] também é mostrada na figura. Podemos agora ter uma idéia dos ganhos não assintóticos de codificação.

Há aspectos importantes a salientar: a) O esquema com  $Q_a=1$  e  $Q_f=8$ , que define 8 regiões de decisão por setor, tem uma degradação sempre menor que 0,4 dB em relação à  $Q=\infty$ . b) O esquema  $Q_a=Q_f=4$ , que define 16 regiões por setor, tem degradação sempre menor que 0,25 dB. Ou seja, o ganho adicional é de apenas 0,15 dB. c) O esquema com 32 regiões de regiões de decisão por setor ( $Q_a=4, Q_f=8$  - figura 2.11) tem um ganho adicional próximo de 0,1 dB.

Numa situação prática, uma decisão sobre qual o esquema a ser utilizado, deverá levar em conta fatores como: ganho de codificação almejado, complexidade de implementação, erros de implementação dos circuitos (principalmente erros na detecção da fase da portadora) do receptor.

#### 2.7.1.3 Um esquema de decisão abrupta.

A figura 2.13 mostra as regiões de decisão abrupta ótimas (no sentido de maximizar o valor do parâmetro exponencial  $R_0$ ).  $R_j$  representa a região de decisão associada ao sinal  $j$ .  $DR_j$  representa o setor definido pelo limiar de decisão inferior ("à direita") do  $R_j$  e a bissetriz do ângulo que define  $R_j$ . O



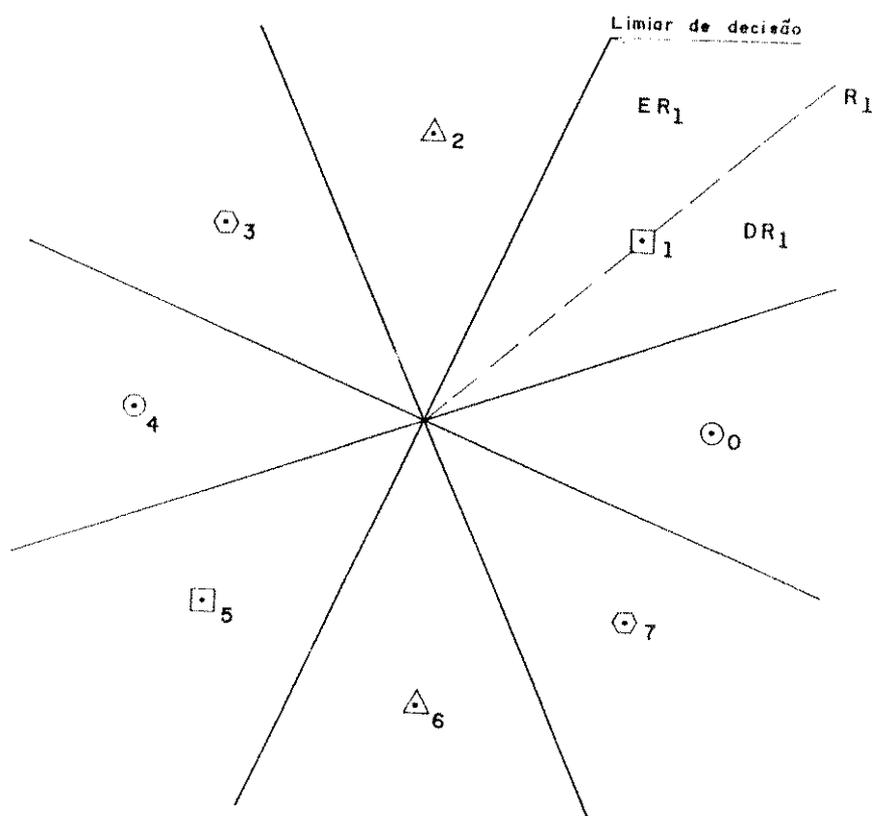


fig.2.13 Regiões ótimas de decisão abrupta.

mesmo é válido para  $ER_j$  em relação ao limiar superior ("à esquerda").

Neste esquema, seriam necessários apenas 3 bits para definição das 8 regiões (uma constelação  $2^m$ -PSK necessita de  $n$  bits). Como poderíamos descobrir os quatro sinais mais próximos? Por exemplo, se o vetor  $\underline{r} \in R_1$ , os quatro sinais serão: 7, 0, 1 e 2, se  $\underline{r} \in DR_1$ ; ou, 0, 1, 2 e 3, se  $\underline{r} \in ER_1$ .

A probabilidade de  $\underline{r} \in DR_j$  é dada por:

$$P(\underline{r} \in DR_j) = \sum_{i=0}^{M-1} P(\underline{r} \in DR_j, \underline{g}_i) = (1/M) \sum_{i=0}^{M-1} P(\underline{r} \in DR_j / \underline{g}_i).$$

Mas, pela simetria da constelação e da densidade do ruído, esta probabilidade independe de  $j$ . Pelas mesmas razões, concluímos que:  $P(\underline{r} \in ER_j)$  é igual a  $P(\underline{r} \in DR_j)$  e também independe de  $j$ .

Portanto,  $P(\underline{r} \in R_j) = P(\underline{r} \in DR_j) + P(\underline{r} \in ER_j) = 2 P(\underline{r} \in DR_j)$ , ou seja:  $P(\underline{r} \in DR_j) = P(\underline{r} \in ER_j) = (1/2) P(\underline{r} \in R_j)$ .

Agora podemos responder a questão: Qual a probabilidade de que  $\underline{r} \in ER_j$  ( $DR_j$ ), se conhecemos o fato de que  $\underline{r} \in R_j$ ?

Utilizando o resultado anterior:

$$\begin{aligned} P(\underline{r} \in ER_j / \underline{r} \in R_j) &= P(\underline{r} \in R_j, \underline{r} \in ER_j) / P(\underline{r} \in R_j) = \\ &= P(\underline{r} \in ER_j) / P(\underline{r} \in R_j) = 1/2 = P(\underline{r} \in DR_j / \underline{r} \in R_j). \end{aligned}$$

Então, uma estratégia para decidir quais os quatro sinais mais próximos seria: supor sempre que  $\underline{r} \in ER_j$ . Em média, estaríamos acertando 50% das vezes.

A conclusão do parágrafo anterior sugere apenas uma estratégia para decidir sobre os quatro sinais mais próximos. O

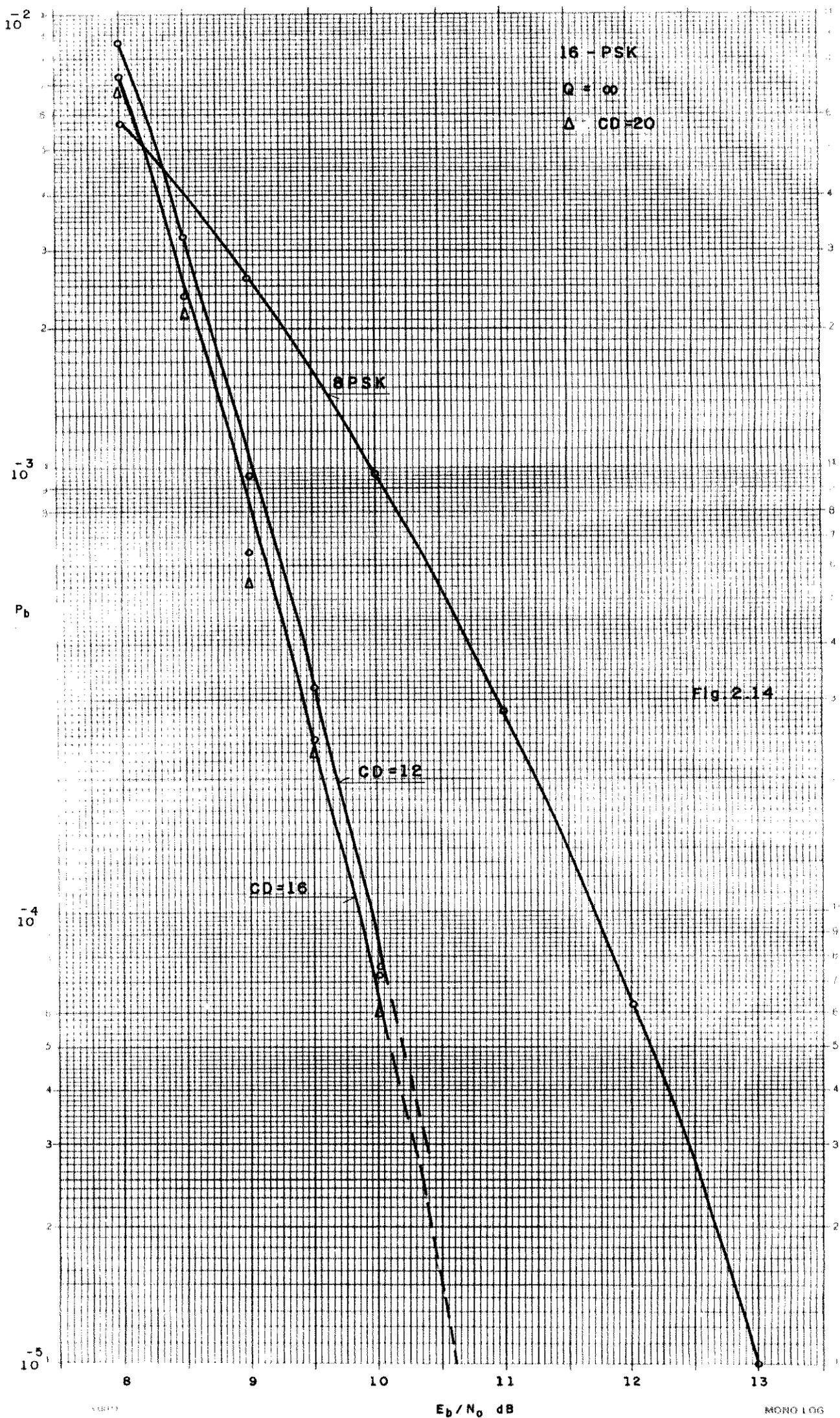
desempenho de um esquema de decisão abrupta utilizando esta estratégia, poderá ser obtido através de simulação. É de se esperar que este desempenho seja inferior ao do esquema de decisão suave com  $J = 2M$  regiões de decisão ( $Q_a=1, Q_f=2$  - seção 2.7.1.1).

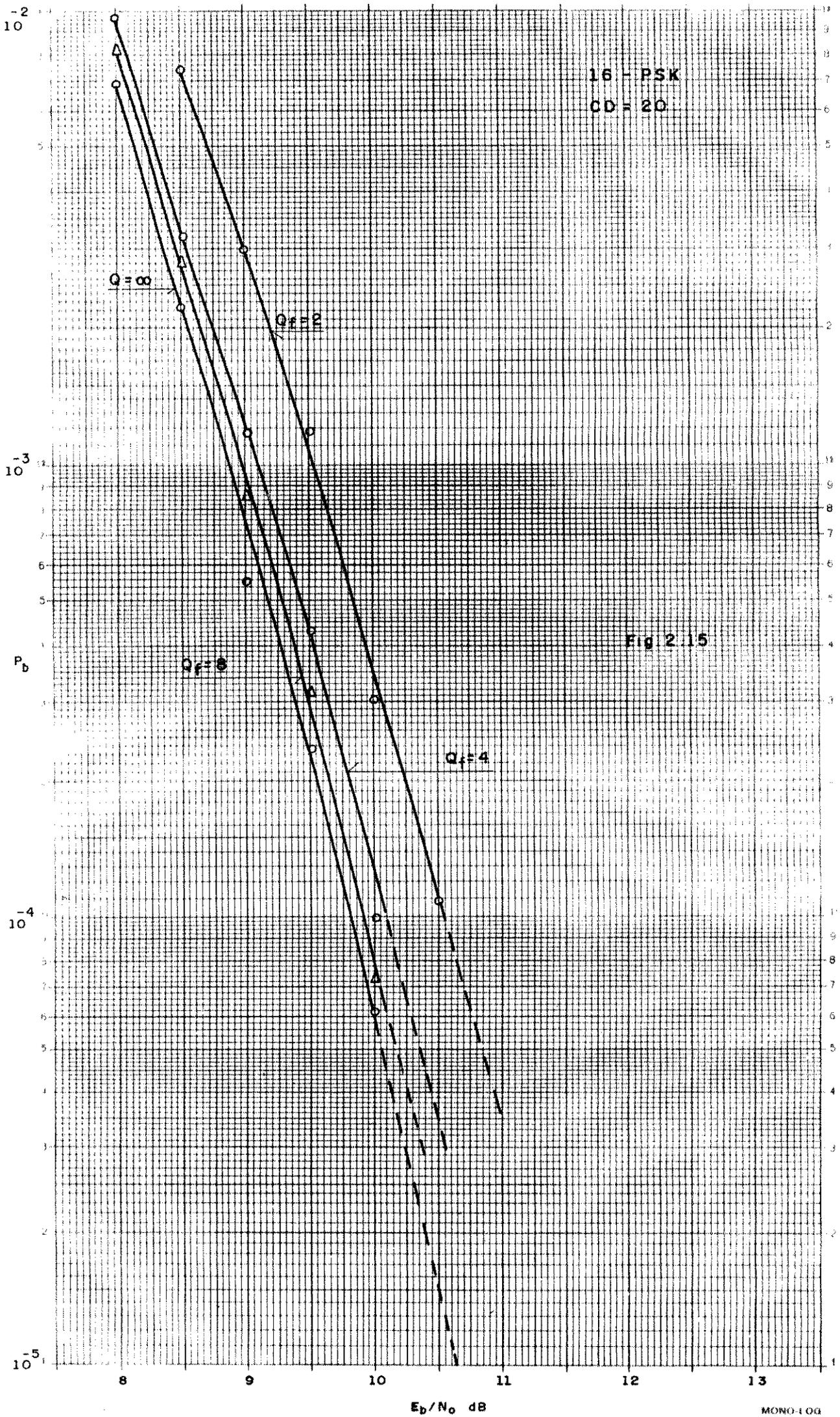
### 2.7.2 A constelação 16-PSK.

A figura 2.14 mostra a variação do desempenho como função do comprimento de decodificação para um esquema de recepção não quantizada. A degradação do desempenho de um esquema com  $CD = 16$  é desprezível em relação ao esquema com  $CD = 20$ . Mesmo assim, nós optamos por fixar o comprimento de decodificação em 20 para todas as simulações realizadas posteriormente (quantização da componente de fase). Da figura, pode-se inferir um ganho de codificação de aproximadamente 1,9 dB para  $P_b = 10^{-4}$ . Para  $P_b = 10^{-5}$ , uma extrapolação da curva de simulação demonstra um ganho de 2,5 dB.

#### 2.7.2.1 Quantização da componente de fase do vetor recebido.

A figura 2.15 mostra a variação do desempenho como função da componente de fase ( $Q_a = 1, Q_f = 2, 4, 8$ ). Para  $P_b = 10^{-3}$ , o esquema com  $Q_a = 1$  e  $Q_f = 2$  tem uma degradação menor que 0,7 dB em relação à  $Q = \infty$ , enquanto que o esquema com  $Q_a = 1$  e  $Q_f = 4$  tem uma degradação de 0,2 dB. Para o mesmo valor de  $P_b$ , o esquema com  $Q_a = 1$  e  $Q_f = 8$  tem uma degradação de 0,1 dB em relação à  $Q = \infty$ .





É interessante fazer uma comparação com o esquema  $Q_a = 1, Q_f = 16$  para a constelação 8-PSK. Como vimos na seção 2.7.1.1 para o código 8-PSK, este esquema apresentou uma degradação menor que 0,4 dB em relação à  $Q = \infty$ . Ambos os esquemas ( $Q_f=16$  para 8-PSK e  $Q_f=8$  para 16-PSK), definem 128 regiões de decisão. Isto mostra que a capacidade de correção de erros é dependente da estrutura do código utilizado. Nós conjecturáramos que um aumento no comprimento do decodificação utilizado para o código 8-PSK, tornaria o desempenho um pouco mais próximo do desempenho com quantização infinita.

## 2.8 Conclusão.

Na referência [7], vários códigos para a constelação 8-PSK foram simulados (inclusive o código com memória  $m = 2$  descrito neste trabalho), utilizando-se uma quantização das coordenadas retangulares do vetor recebido: componentes de fase e quadratura do sinal. Com 5 bits para cada componente de fase/quadratura, o desempenho se mostrou praticamente o mesmo da recepção não quantizada. Podemos fazer uma comparação com o esquema  $Q_a = 4$  e  $Q_f = 8$  (seção 2.7.1.2), que também demonstrou uma degradação de desempenho desprezível em relação à recepção não quantizada. Se considerarmos 3 bits adicionais necessários para determinação do setor do vetor recebido, um total de 8 ( $= 2 + 3 + 3$ ) bits foram utilizados. Isto nos dá uma economia de 20% (8 em 10) no número total de bits. Em termos do número total de regiões de decisão (que está diretamente relacionado com a complexidade de armazenamento das

métricas) a redução é significativa: de 1024 ( $=2^{10}$ ) regiões para 256 ( $=2^8$ ) regiões.

Para a classe de códigos considerada, estamos sempre determinando quatro sinais mais próximos. A obtenção da componente de fase do vetor recebido é utilizada na determinação destes sinais. As simulações demonstraram a viabilidade da quantização da componente de fase. Portanto, um esquema de quantização polar parece ser o mais apropriado para sistemas de modulação M-PSK codificada.

## APENDICE 1:

### LIMITANTES SUPERIORES E A PROCURA DO $d_{\min}$ .

A referência [45] descreve um método de obtenção de limitantes superiores para o desempenho do algoritmo de Viterbi num contexto geral de detecção/estimação. A referência [9] utiliza este método para os códigos AR. O método é uma generalização dos limitantes obtidos através da definição da função de transferência do código, como foi descrito em [12] para códigos convolucionais binários. Por isso mesmo, o método é denominado função de transferência generalizada.

A propriedade de fechamento (primeira propriedade de um grupo) dos códigos convolucionais binários que utilizam métrica de Hamming, simplifica a obtenção do limitante. Podemos fixar uma determinada sequência transmitida, e através dela calcular o limitante. O diagrama de estados possuirá  $2^m$  estados, onde  $m$  é a memória do código. Para códigos AR, nem sempre é possível se fixar uma sequência transmitida. Quando todas as possíveis sequências transmitidas são consideradas, há a necessidade de definir um diagrama de "super-estados" [45]. Este diagrama passa a ter  $(2^m)^2$  super-estados.

Considere a figura 1.2 do capítulo um. Seja  $d(\underline{e}_t, \tilde{\underline{e}}_t)$  uma medida de distorção entre os símbolos de informação  $\underline{e}_t$  e  $\tilde{\underline{e}}_t$ . Se quisermos que  $E(d(\underline{e}_t, \tilde{\underline{e}}_t))$  represente a probabilidade de erro de bit média no instante  $t$ , e, sendo  $\underline{e}$  um símbolo que contém dois bits, devemos utilizar a seguinte tabela para o valor de  $d(\underline{e}_t, \tilde{\underline{e}}_t)$ :

$\begin{matrix} \underline{e}_t \\ \tilde{e}_t \end{matrix}$	00	01	10	11
00	0	1/2	1/2	1
01	1/2	0	1	1/2
10	1/2	1	0	1/2
11	1	1/2	1/2	0

Em [45] mostra-se que  $E( d(\underline{e}_t, \hat{\underline{e}}_t) / \underline{S}^L )$ , a distorção média, no instante  $t$ , condicionada ao envio de uma sequência de símbolos de canal  $\underline{S}^L$ , é independente de  $t$  e pode ser limitada por:

$$E( d(\underline{e}_t, \hat{\underline{e}}_t) / \underline{S}^L ) \leq \sum_{j=1}^{\infty} \sum_{S(0,j/\underline{S}^L[0,j])} \left[ \sum_{l=0}^{j-1} d(\underline{e}_l, \hat{\underline{e}}_l) \right] \prod_{n=0}^{j-1} D(\underline{e}_n, \hat{\underline{e}}_n) \quad (\Lambda.1)$$

onde:

$$- D(\underline{e}_n, \hat{\underline{e}}_n) = \exp(-|\underline{e}_n - \hat{\underline{e}}_n|^2 / 8 \sigma^2).$$

-  $\sigma^2$  é a variância do ruído em cada dimensão.

-  $\underline{S}^L[0,j]$  é uma subsequência da sequência  $\underline{S}^L$  enviada, desde o instante 0 ao instante  $j$  ( $(j-1)$  símbolos transmitidos).

-  $S(0,j/\underline{S}^L[0,j])$  é o conjunto de todos os possíveis eventos de erro que começam no instante 0 (o instante inicial) e terminam no instante  $j$ :

$$S(0,j/\underline{S}^L[0,j]) = \{ \tilde{\underline{S}}[0,j] : \tilde{\underline{G}}_0 = \underline{G}_0, \tilde{\underline{G}}_j = \underline{G}_j, \tilde{\underline{G}}_t \neq \underline{G}_t, 0 < t < j \}$$

( $\underline{G}$  denotando o estado do codificador).

Note que a expressão acima inclui todas as transições paralelas. O conjunto  $S(0,1/\underline{S}^L[0,1])$  inclui todos os eventos de erro de comprimento unitário.

Nós nos perguntamos quais seriam as condições no código AR considerado, para que o limitante da equação  $\Lambda.1$  independesse

da sequência enviada  $\underline{s}^L$ . Se o código é linear (seção 1.5.3), o conjunto de distorções de uma sequência  $\underline{s}^L$ ,  $(\sum_{l=0}^{j-1} d(\underline{s}_l, \underline{\hat{s}}_l))$ , independe da sequência enviada. Se associamos aos símbolos do alfabeto de modulação, um grupo que contém a soma "  $\oplus$  ", como fizemos na seção 1.5.4, o código AR terá a mesma propriedade dos códigos convolucionais binários: o conjunto de distâncias de Hamming em termos de bits de informação (o conjunto de distorções), independe da sequência enviada [46].

Resta saber se o conjunto dos termos  $(\prod_{n=0}^{j-1} D(\underline{\hat{s}}_n, \underline{s}_n^L))$ , também irá independe da sequência enviada. Para códigos convolucionais com métrica de Hamming, isto ocorre. Para códigos AR, a condição suficiente é a superlinearidade (seção 1.5.12).

Voltando à equação A.1, a distorção média não condicionada será então dada por:

$$E(d(\underline{e}_t, \underline{\hat{e}}_t)) = \sum_i q(i) E(d(\underline{e}_t, \underline{\hat{e}}_t) / \underline{s}^L) \quad (A.2)$$

onde  $q(\cdot)$  é a distribuição de probabilidades das possíveis sequências transmitidas.

Vamos separar alguns termos da equação A.1. Considere os seguintes conjuntos:

1.  $S(0, 1 / \underline{s}^L[0, 1])$ , o conjunto de eventos de erro de comprimento unitário.
2.  $S'(0, j / \underline{s}^L[0, j])$ ,  $j > 1$ , o conjunto de eventos de erro cuja distância euclidiana à subsequência  $\underline{s}^L[0, j]$ , é a menor possível dentre cada conjunto de eventos de erro definido por uma subse-

quência de estados, ou seja:

$$S'(0, j/\underline{S}^L[0, j]) = ( \underline{S}[0, j]: \underline{\tilde{G}}_0 = \underline{G}_0, \underline{\tilde{G}}_j = \underline{G}_j, \underline{\tilde{G}}_t \neq \underline{G}_t, 0 < t < j / \underline{\tilde{S}}[0, j] \text{ é o evento de erro de menor distância } ).$$

$$3. \overline{S}(0, j/\underline{S}^L[0, j]) = S(0, j/\underline{S}^L[0, j]) - S(0, 1/\underline{S}^L[0, 1]) - S'(0, j/\underline{S}^L[0, j]),$$

o conjunto de eventos de erro restantes.

Considere que o conjunto  $S(0, 1/\underline{S}^L[0, 1])$  possui o mesmo conjunto de termos  $( \prod_{n=0}^{i-1} D(\underline{\tilde{g}}_n, \underline{g}_n) )$  para qualquer sequência  $\underline{S}^L$ . Isto acontece para as constelações consideradas no capítulo um.

Ora, para uma relação sinal ruído alta, os termos da equação A.1 correspondentes a  $\overline{S}(0, j/\underline{S}^L[0, j])$  serão desprezíveis. Isto pode ser visto da seguinte maneira: Considere um conjunto de eventos de erro definido por uma subsequência de estados. Considere

$\exp[ -(d_{menor}^2 + \Delta d)/8 \sigma^2 ]$ , como sendo um dos termos

$[ \prod_{n=0}^{i-1} D(\underline{\tilde{g}}_n, \underline{g}_n) ]$  pertencente ao conjunto  $\overline{S}(0, j/\underline{S}^L[0, j])$ , onde

$d_{menor}$  é a distância euclidiana do termo que pertence ao conjunto  $S'(0, j/\underline{S}^L[0, j])$ . Quando  $\sigma^2$  diminui, o fator  $\exp(-\Delta d/8 \sigma^2)$

diminui exponencialmente. O número de erros de bits de informação, em cada evento de erro possível, permanece sempre inalterado. Portanto, é de se esperar que para relações sinal ruído altas, os termos pertencentes a  $\overline{S}(0, j/\underline{S}^L[0, j])$  sejam desprezíveis.

Podemos então, retirar estes eventos de erro da treliça. Se a treliça for superlinear, a equação A.1 irá independor do  $\underline{S}^L$ . Sendo assim, não precisaremos recorrer a super-estados para calcular a equação A.2. Os eventos de

comprimento unitário são obtidos diretamente da constelação de sinais considerada.

O diagrama de estados correspondente à treliça reduzida para o código 8-PSK de memória  $m = 2$ , simulado na seção 2.7.1, é mostrado na figura 2.16. Utilizando a regra de Mason [47], obtemos a função de transferência deste diagrama, e dela, o limitante aproximado. O resultado é mostrado na figura 2.17 em comparação com o resultado de simulação. Para uma probabilidade de erro igual a  $10^{-4}$ , a diferença em relação ao desempenho real é menor que 0,5 dB. Para probabilidades mais baixas, será cada vez menor.

É interessante notar que este limitante aproximado, será sempre mais próximo da curva real de desempenho, do que um limitante obtido pela equação A.2, pois estamos desprezando termos. Para relações sinal ruído altas, os dois limitantes se confundem.

A figura 2.17 também mostra o limitante inferior obtido da equação 1.4 (seção 1.5.8). Como já foi visto, o  $d_{min}$  deste código é a distância entre os dois símbolos que compõem as transições paralelas. O bit de informação não codificado é que define qual dos dois símbolos será transmitido. Portanto, um evento de erro corresponde a apenas um erro nos bits de informação. Conclusão:  $P_b \approx (1/2) N(d_{min}) Q(d_{min}/2\sigma) = (1/2) Q(d_{min}/2\sigma)$ .

Como vimos na seção 1.5.12, a condição de superlinearidade facilita enormemente a procura do  $d_{min}$  de um

código: podemos fixar uma sequência de referência. Se o código não for superlinear, será necessário considerar todos os pares possíveis de sequências.

Há uma maneira de se evitar a comparação entre todos os pares de sequências, sob a restrição de se obter um limitante inferior para o  $d_{min}$  verdadeiro do código. Construa a matriz  $H$  e a matriz  $D$  do código (seção 1.5.12). Construa uma matriz  $D'$  como se segue:

1. Considere um conjunto de distâncias em  $D$  que possui entradas idênticas em  $H$ .
2. Substitua todos os elementos do conjunto em  $D$ , pela menor das distâncias que compõem o conjunto.
3. Repita o procedimento para todos os conjuntos possíveis de entradas idênticas em  $H$ .

Se utilizarmos  $D'$  como sendo a matriz de distâncias do código, podemos fixar uma sequência de referência para encontrarmos o  $d_{min}$  do código.

As figuras 2.18 mostram um exemplo de construção de  $D'$  para a constelação 8-PSK (todas as matrizes são simétricas em relação à diagonal). Para esta constelação, só tem sentido obter-se a matriz  $D'$  para códigos que não contenham transições paralelas; códigos com transições paralelas serão superlineares (exemplo 3 - seção 1.5.13).

É interessante notar que o limitante inferior no valor de  $d_{min}$ , obtido da maneira descrita acima, é sempre maior ou igual ao limitante inferior obtido em [2], onde são utilizadas as distâncias intra-subconjuntos de cada nível da partição da constelação.

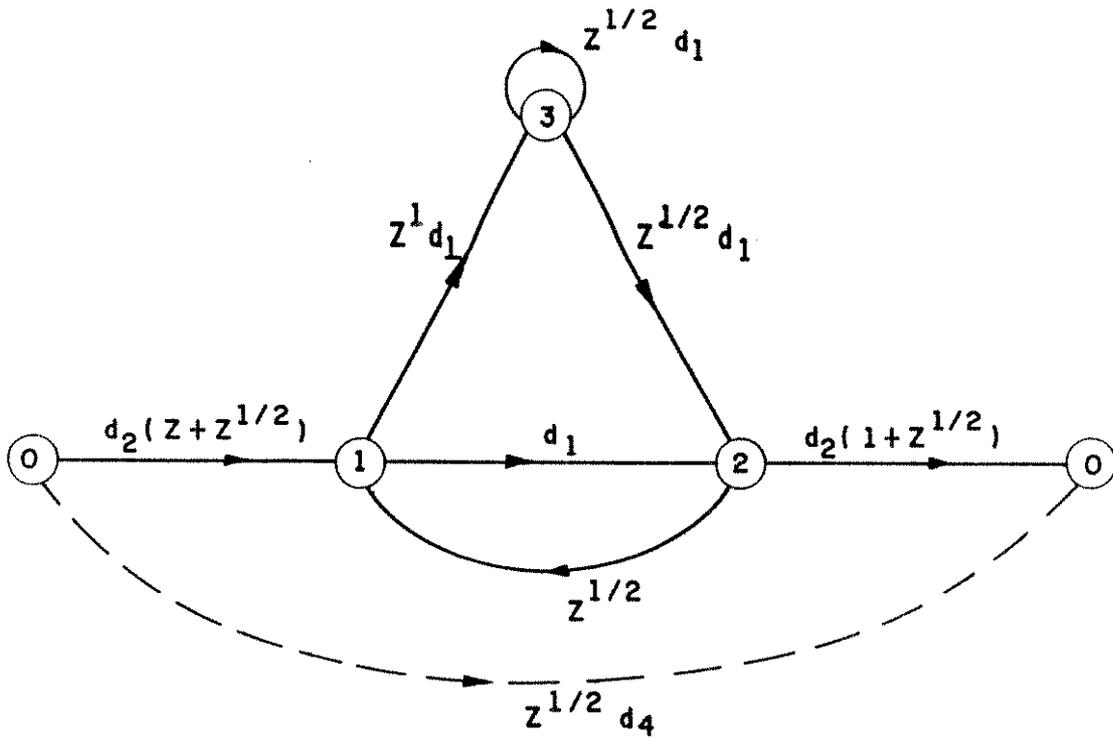


fig. 2.16. Diagrama de estados.

$$d_i = \exp(-|\underline{e}_i - \underline{e}_0|^2 / 8 \sigma^2),$$

onde  $\underline{e}_0$  é o símbolo do canal 0,

$\underline{e}_i$  é o símbolo do canal i, o,

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

Veja figura 2.1.

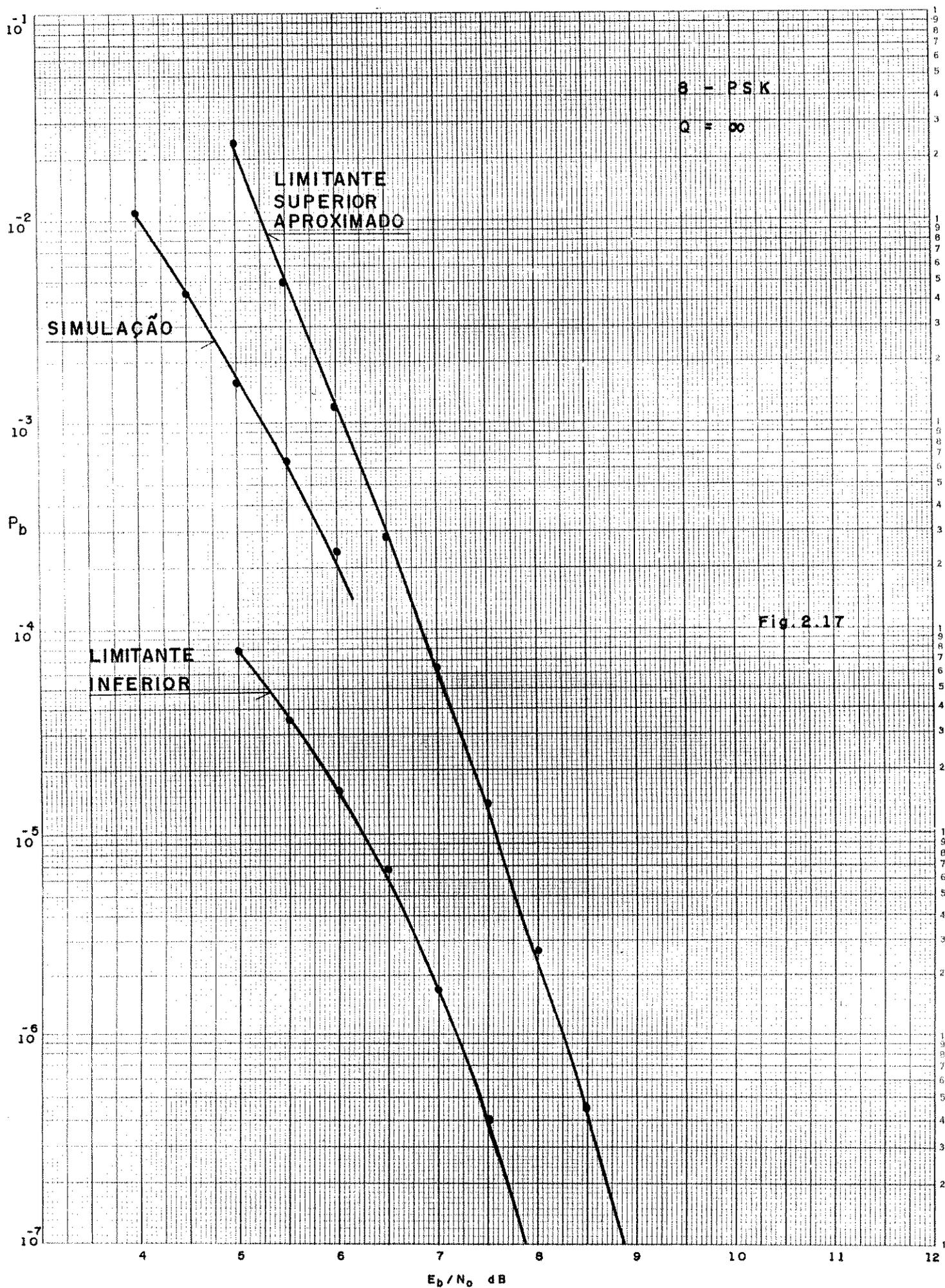


Fig. 2.17

	0	1	2	3	4	5	6	7
0	0							
1	1							
2	2	3						
3	3	2	1					
4	4	5	6	7				
5	5	4	7	6	1			
6	6	7	4	5	2	3		
7	7	6	5	4	3	2	1	0

fig 2.18a Matriz soma II.

	0	1	2	3	4	5	6	7
0	0							
1	$d_1$							
2	$d_2$	$d_1$						
3	$d_3$	$d_2$	$d_1$					
4	$d_4$	$d_3$	$d_2$	$d_1$				
5	$d_3$	$d_4$	$d_3$	$d_2$	$d_1$			
6	$d_2$	$d_3$	$d_4$	$d_3$	$d_2$	$d_1$		
7	$d_1$	$d_2$	$d_3$	$d_4$	$d_3$	$d_2$	$d_1$	0

fig. 2.18b Matriz de distâncias D.

$d_l$  representa a distância entre o símbolo de canal 0 e o símbolo de canal 1. Veja fig.2.1.

	0	1	2	3	4	5	6	7
0	0							
1	$d_1$							
2	$d_2$	$d_1$						
3	$d_1$	$d_2$	$d_1$					
4	$d_4$	$d_3$	$d_2$	$d_1$				
5	$d_3$	$d_4$	$d_1$	$d_2$	$d_1$			
6	$d_2$	$d_1$	$d_4$	$d_3$	$d_2$	$d_1$		
7	$d_1$	$d_2$	$d_3$	$d_4$	$d_1$	$d_2$	$d_1$	0

fig. 2.18c Matriz de distâncias  $D'$   
(Matriz  $D$  "superlinearizada").

## APÊNDICE 2:

### ALGUNS RESULTADOS COM METRICA QUANTIZADA.

A figura 2.20 mostra os resultados de simulação do código 8-PSK com memória  $m=2$ , utilizando-se as métricas quantizadas definidas na seção 2.5.2. Todas as simulações foram realizadas com um comprimento de decodificação  $CD = 12$ . Como trabalhamos com números inteiros no algoritmo de Viterbi, as métricas foram escalonadas e arredondadas. Não nos preocupamos em minimizar o número de bits necessários à representação das métricas. Os seguintes esquemas foram considerados:

a) Um esquema com  $Q_a = 1$ ,  $Q_f = 2$ . Como apenas a componente de fase é quantizada, nós definimos o raio médio como sendo

$r_M^{\lambda} = |s_{\lambda}| = A$ . A degradação de desempenho, em relação à  $Q = \infty$ , é sempre menor que 0,8 dB.

b) Um esquema com  $Q_a = 2$ ,  $Q_f = 2$ . O espaçamento  $\Delta r$  ótimo obtido, foi de 0,6 A. O desempenho se mostra praticamente o mesmo do esquema com  $Q_a = 1$ ,  $Q_f = 2$ .

c) Um esquema com  $Q_a = Q_f = 4$ . A otimização do espaçamento é mostrada na figura 2.19a. A degradação de desempenho, em relação à  $Q = \infty$ , é sempre menor que 0,2 dB.

d) Um esquema com  $Q_a = Q_f = 8$ . A otimização do espaçamento é mostrada na figura 2.19.b. A degradação de desempenho se mostrou desprezível em relação à  $Q = \infty$ .

$10^{-1}$

$10^{-2}$

$10^{-3}$

$10^{-4}$

$P_b$

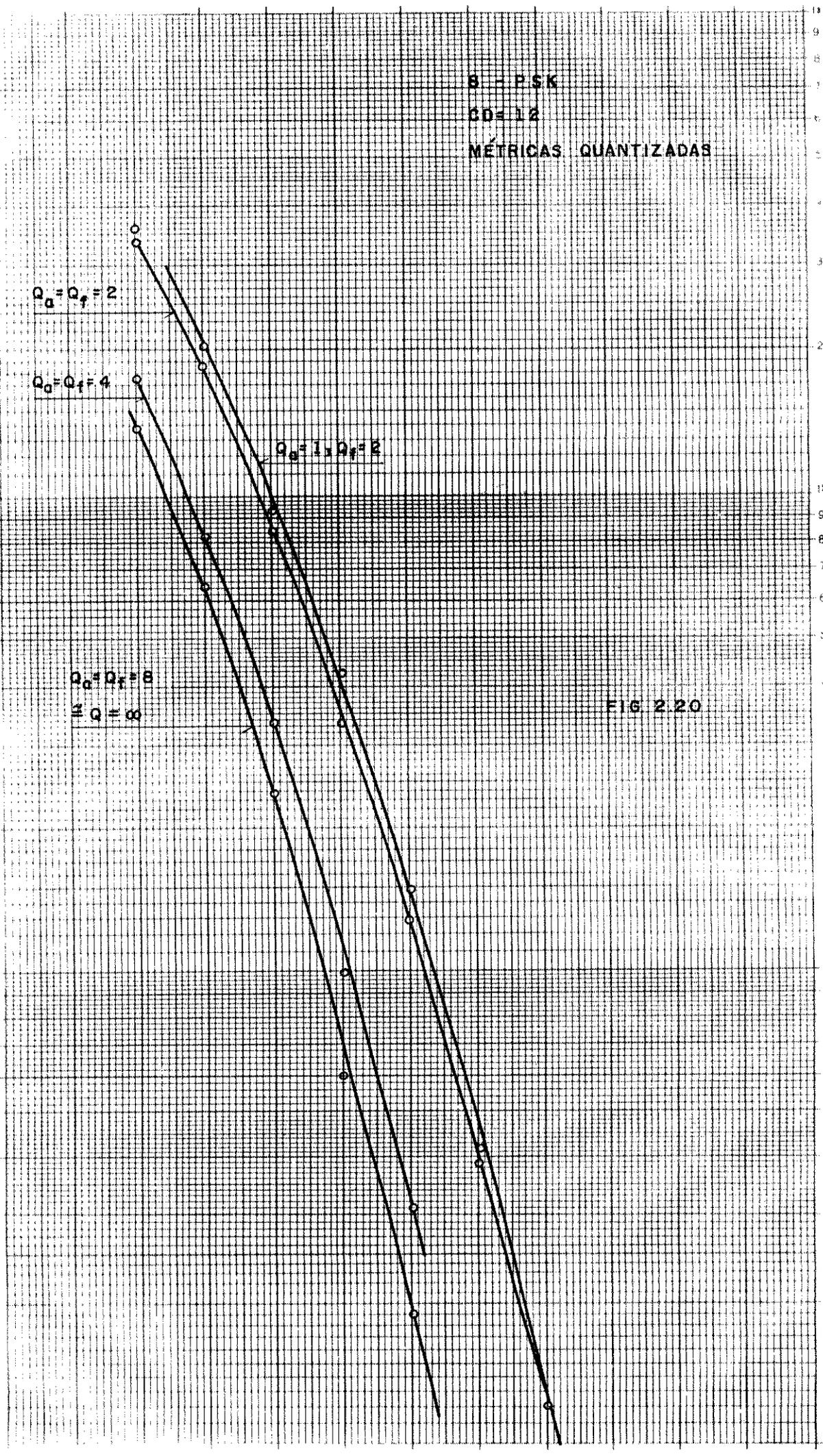
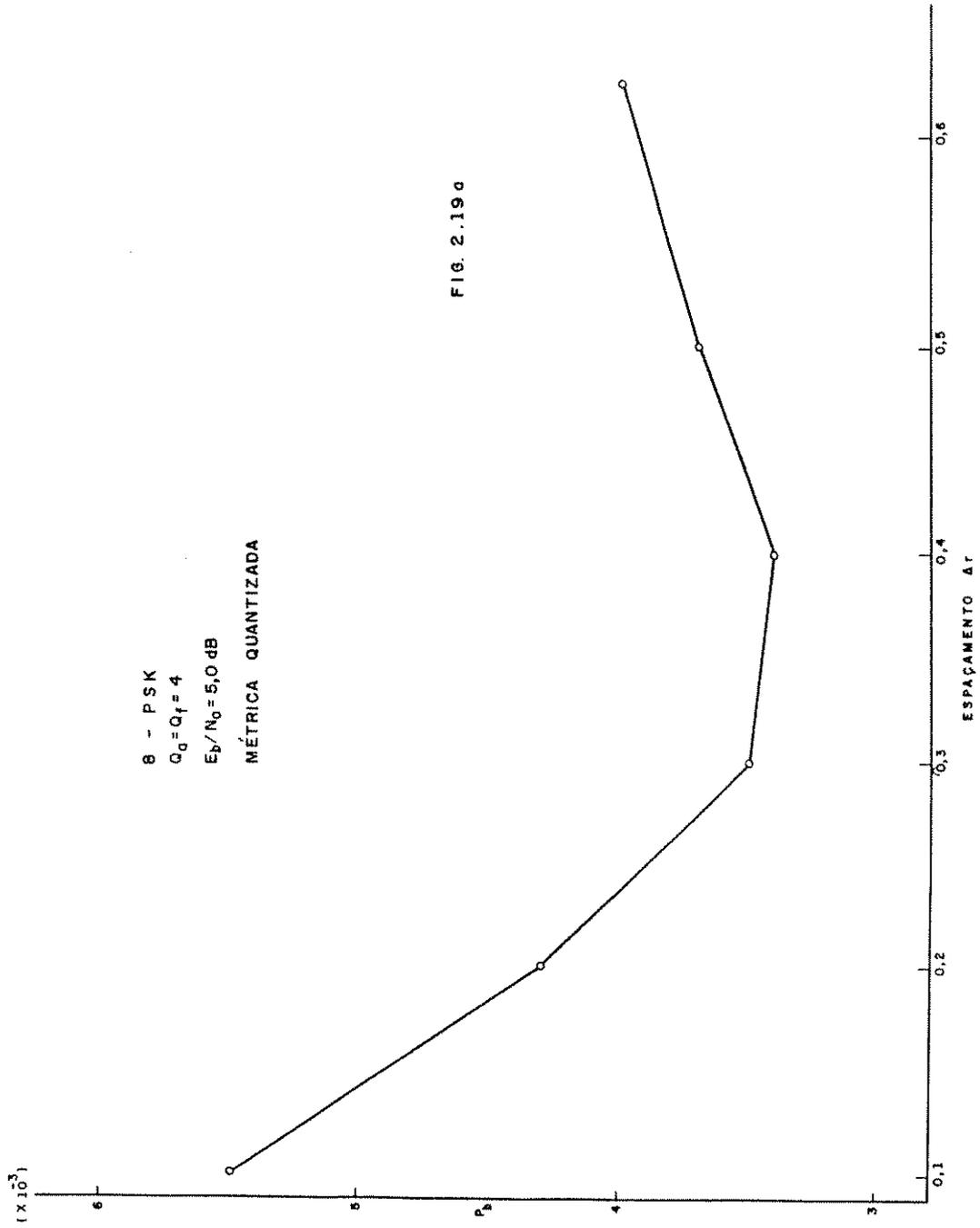
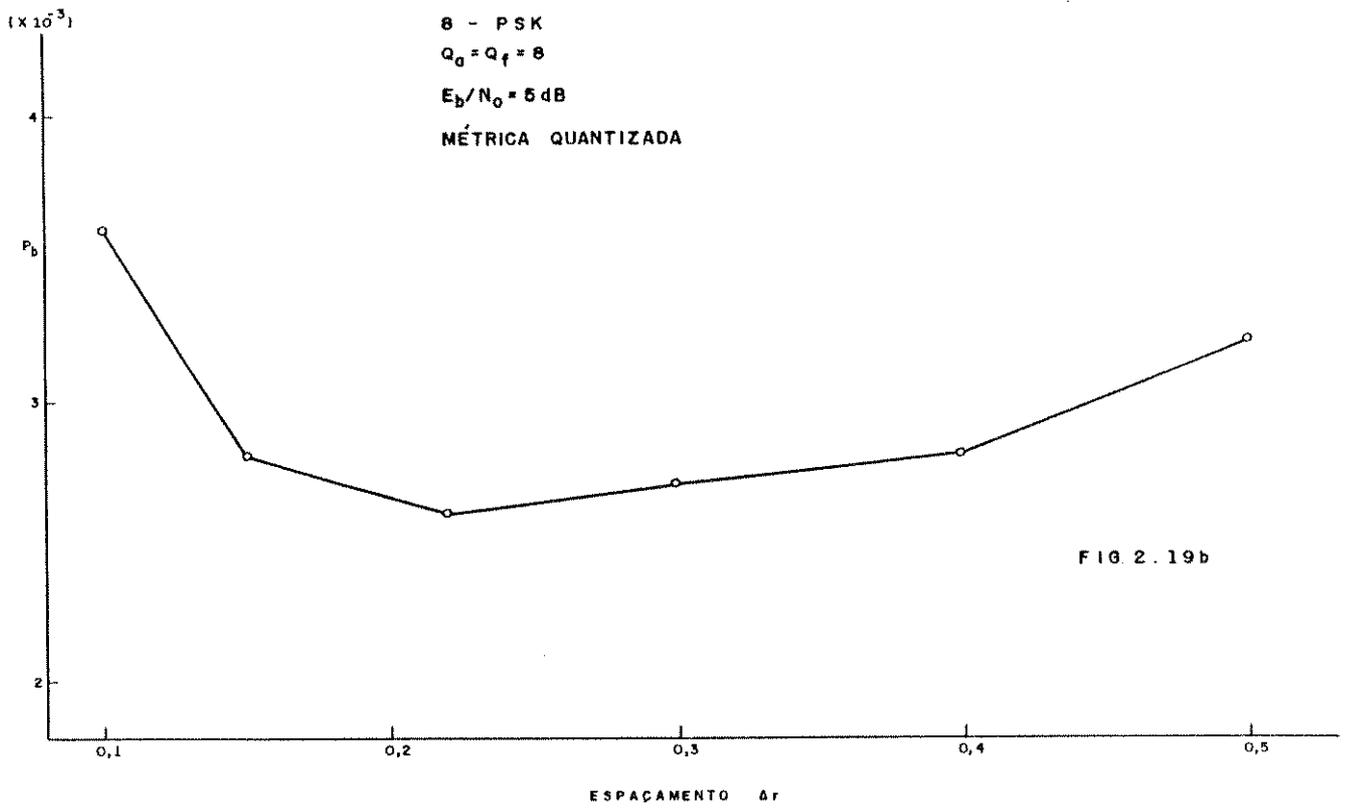


FIG 2.20

$E_b/N_0$  dB

MONO-LOG





C A P Í T U L O

T R E S:

DESEMPENHO EM CANAIS NÃO LINEARES DE FAIXA LIMITADA:

CANAIS DE SATELITE.

### 3.1 Introdução.

Um modelo não detalhado de um sistema de comunicação por satélite é mostrado na figura 3.1 [27].

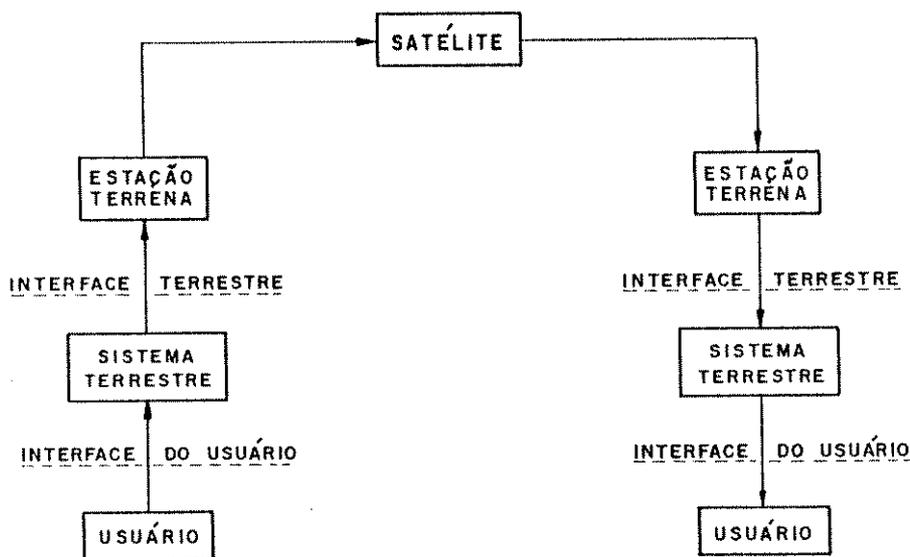


fig.3.1 Sistema de comunicação por satélite.

Nós estamos interessados no enlace de comunicação formado pelas estações terrenas e o satélite.

"Para explorar as propriedades geométricas únicas de ampla área de visibilidade e múltipla conectividade disponíveis pelos satélites, os diversos enlaces de comunicação que utilizam o satélite devem estar separados uns dos outros" [28].

Uma maneira de se conseguir isto, é através do sistema AMDT (acesso múltiplo por divisão no tempo). "Neste sistema, à cada estação terrena é designado um intervalo de tempo para transmissão, e todas as estações usam a mesma frequência de

portadora num particular "transponder" do satélite" [28].

Desta maneira, os produtos de intermodulação são eliminados e o amplificador de potência do satélite pode operar na região de saturação onde se obtém a máxima potência transmitida [29]. Comunicação digital através da técnica AMDT é utilizada em vários sistemas já existentes [29].

A modulação M-PSK se torna apropriada para utilização em canais que contenham um amplificador trabalhando numa região não linear, pelo fato de que todos os sinais da constelação possuem a mesma amplitude.

O uso de modulação 8-PSK [7,31] e 16-PSK [9,8] codificada, demonstraram manter os ganhos relativos de codificação, mesmo sob o efeito de não linearidades e interferência intersimbólica.

Neste capítulo, nós descrevemos o desenvolvimento de um programa de simulação que nos permitiu obter o desempenho dos esquemas de recepção quantizada, propostos no capítulo anterior, sob a ação de um canal de satélite. Vamos inicialmente considerar o modelo adotado.

### 3.2 O modelo adotado.

Um modelo detalhado do enlace de subida, do "transponder" do satélite, e do enlace de descida, pode ser encontrado em [29]. Como modelo do canal de satélite a ser simulado, nós vamos considerar o que é mostrado na figura 3.2 [30,31].

Comparando com a figura 1.1, o canal elétrico de

comunicação é a combinação dos filtros de transmissão e recepção e do amplificador do "transponder". Os blocos restantes representam os mesmos elementos que seriam utilizados se o canal fosse o AWGN.

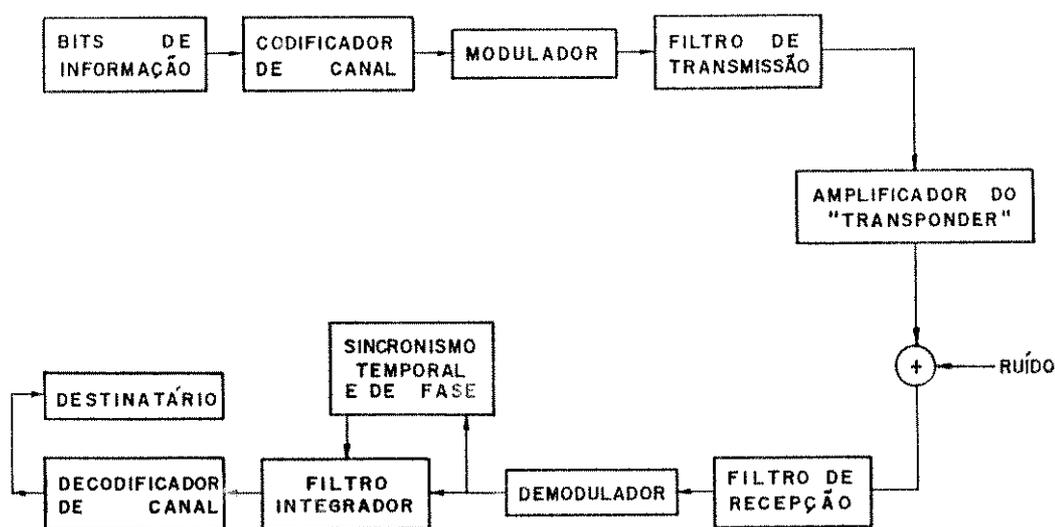


fig. 3.2 Modelo do canal de satélite.

O modelo considerado exclui o ruído do enlace de subida. Ele se aplica em situações em que as estações transmissoras possuem alta potência e grandes antenas, o que ocorre na maioria das vezes [31]. O fato de considerarmos apenas o ruído do enlace de descida traz simplificações ao programa de simulação que serão ressaltadas posteriormente.

Os efeitos do processamento de sinais de frequências altas através dos diversos elementos do canal, podem ser obtidos de maneira mais simples se utilizarmos uma representação complexa em banda base, como descreve a próxima seção.

### 3.3 Representação complexa.

#### 3.3.1 Sinais M-PSK.

Como vimos na seção 1.3, os sinais M-PSK podem ser considerados como sendo um processo estocástico de faixa estreita, expressos por:

$$s_i(t) = \sqrt{2E_s/T} \cos(\omega_0 t + 2\pi i/M) g(t) = \\ = \sqrt{2E_s/T} [\cos(2\pi i/M) \cos(\omega_0 t) - \text{sen}(2\pi i/M) \text{sen}(\omega_0 t)] g(t),$$

$$1 \leq i \leq M.$$

A pré-envoltória [5] de  $s(t)$  é dada por:

$$s_i^P(t) = s_i(t) + j s_i^H(t),$$

onde  $s_i^H(t)$  é a transformada de Hilbert de  $s_i(t)$ .

Mas  $s_i^P(t) = \tilde{s}_i(t) \exp(j\Omega t)$ , onde  $\tilde{s}_i(t)$  é a envoltória complexa de  $s_i(t)$ .

$$\text{Assim: } s_i^P(t) = \sqrt{2E_s/T} [\cos(2\pi i/M) + j \text{sen}(2\pi i/M)] e^{j\omega_0 t} g(t).$$

Fazendo  $\Omega = \omega_0$ , vem:

$$\tilde{s}_i(t) = \sqrt{2E_s/T} [\cos(2\pi i/M) + j \text{sen}(2\pi i/M)] g(t) = \\ = \sqrt{2E_s/T} e^{j2\pi i/M} g(t), \quad 1 \leq i \leq M.$$

#### 3.3.2 Filtros.

Se  $h(t)$  é a resposta impulsiva de um filtro de faixa estreita, nós podemos representá-la por  $\tilde{h}(t)$ , sendo que [5],

$$h(t) = \text{Re} [2 \tilde{h}(t) \exp(j\Omega t)].$$

Desprezando as harmônicas de segunda ordem, podemos escrever [5]:

$$\tilde{y}(t) \cong \tilde{h}(t) * \tilde{x}(t)$$

onde,  $\tilde{x}(t)$  é a envoltória complexa do sinal de entrada do filtro,  $\tilde{y}(t)$  é a envoltória complexa do sinal de saída do filtro, e "\*" denota convolução.

Para obtermos a resposta em frequência de  $\tilde{h}(t)$  basta deslocarmos a parte de frequências positivas de  $H(j\omega)$  (a transformada de Fourier de  $h(t)$ ) [5], como mostra a figura 3.3.

Como a representação complexa desloca o espectro dos sinais para a origem do eixo de frequências ( $\omega=0$ ), a simulação do demodulador não precisa ser realizada já que iremos supor uma recuperação perfeita da frequência de portadora  $\omega_0$ .

### 3.4 Teoremas.

Os próximos teoremas nos garantem a possibilidade de fazermos a simulação digital (através de amostras) do sistema de comunicação considerado.

#### 3.4.1 Teorema da amostragem para processos estocásticos [32,33].

Se  $s(t)$  é um processo estocástico cuja densidade espectral de potência é limitada em faixa:

$$S_s(\omega) = 0, \quad |\omega| > \lambda, \quad e,$$

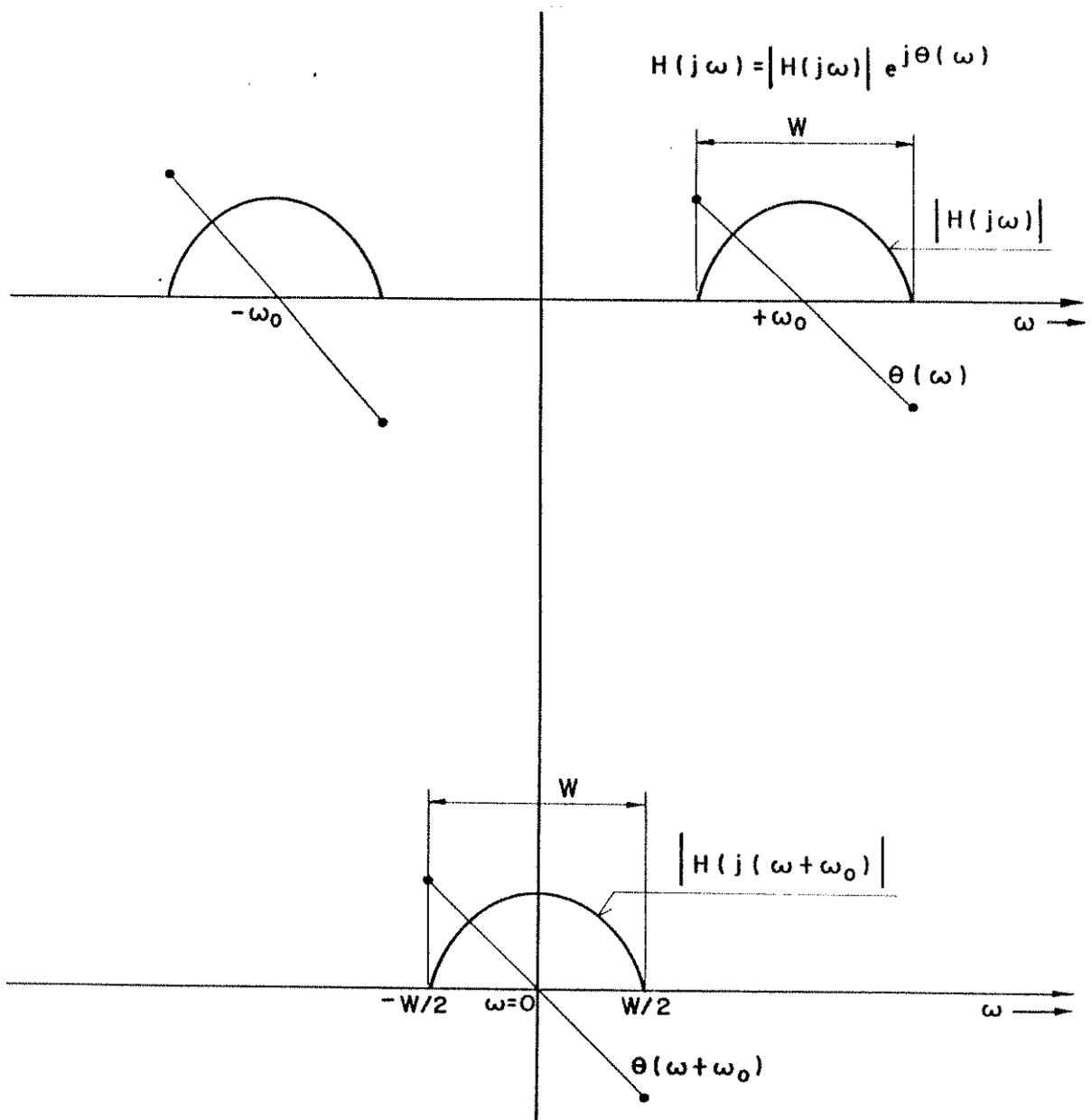


fig. 3.3 Obtenção da resposta em frequência de  $\kappa(t)$ .

$$\hat{s}(t) = \sum_{n=-\infty}^{\infty} s(nT_a) S_{\alpha}[\lambda(t-nT_a)], \quad T_a = \pi/\lambda, \quad (1.1)$$

então:

$\hat{s}(t)$  é igual a  $s(t)$  no sentido de erro quadrático médio,  $E(|s(t) - \hat{s}(t)|^2) = 0$ .

Sendo assim, podemos utilizar as amostras dos sinais M-PSK na saída do modulador, desde que  $T_a \leq \pi/\lambda$ ,  $\lambda$  sendo a faixa dos mesmos.

### 3.4.2 Teorema da simulação para processos estocásticos.

Este teorema é equivalente ao teorema da simulação para sinais determinísticos [33].

Considere a figura 3.4:  $H_a(\omega)$  é o sistema analógico a ser simulado;  $f(t)$  é o processo estocástico na entrada do sistema e  $g(t)$  é o processo na saída.  $f[n]$  e  $g[n]$  são as seqüências de entrada e saída do sistema discreto, respectivamente.

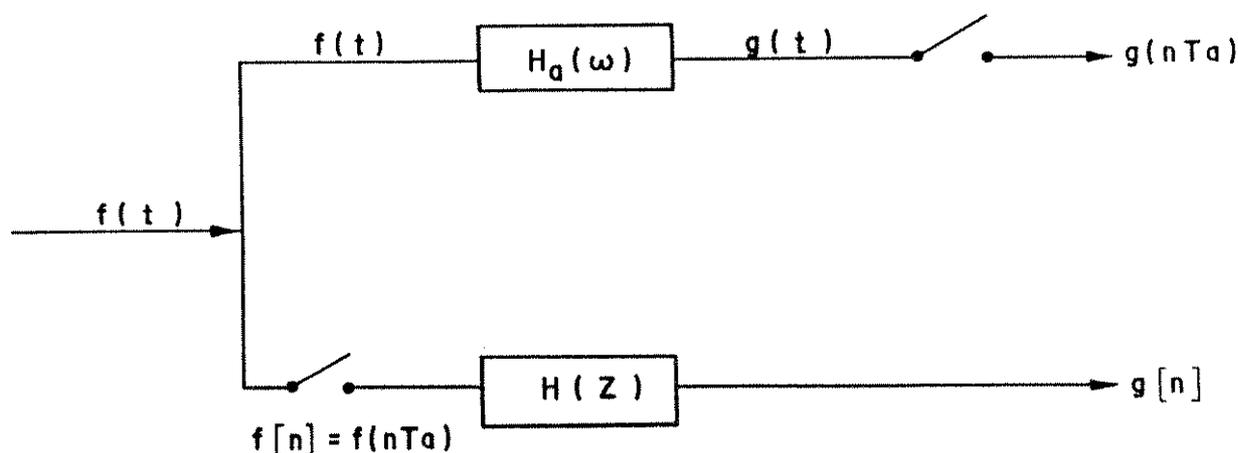


fig. 3.4 Teorema da simulação para processos estocásticos.

Nossa intenção é saber quais as condições em  $H(z)$ , a

função de transferência do sistema discreto, para que  $g[n]$  seja igual a  $g(nT_a)$  no sentido de erro quadrático médio.

#  
Teorema.

Se a densidade espectral de potência de  $f(t)$  é limitada em faixa, isto é,

$$S_f(\omega) = 0 \quad , \quad |\omega| > \lambda \quad , \quad \lambda = \pi/T_a,$$

e,

$$H(e^{j\omega T_a}) = H_a(\omega) \quad \text{para} \quad |\omega| < \lambda, \quad (3.2)$$

então,

$$g[n] = g(nT_a) \quad \text{e} \quad \hat{g}(t) = g(t),$$

ambas as igualdades no sentido de erro quadrático médio.

$\hat{g}(t)$  é reconstruído a partir de  $g[n]$  como na equação (3.1).

#  
Demonstração.

Da equação 3.2, temos:  $|H(e^{j\omega T_a})|^2 = |H_a(\omega)|^2, |\omega| < \lambda$ .

De [33, pg.320]:

$$S_{ff}(\omega) = (1/T_a) \sum_{m=-\infty}^{\infty} S_f(\omega + 2\lambda m)$$

onde  $S_{ff}(\omega)$  é a densidade espectral de potência de  $f[n]$ .

Então,  $S_{gg}(\omega)$ , a densidade espectral de  $g[n]$  é expressa por:

$$S_{gg}(\omega) = S_{ff}(\omega) |H(e^{j\omega T_a})|^2 = (1/T_a) \sum_{m=-\infty}^{\infty} S_f(\omega + 2\lambda m) |H(e^{j\omega T_a})|^2.$$

Como,

$$S_f(w) = 0, |w| > \lambda, \text{ e, } |H(e^{jwT_a})|^2 = |H_a(w)|^2, |w| < \lambda,$$

então,

$$S_{gg}(w) = (1/T_a) S_f(w) |H_a(w)|^2 = S_g(w)/T_a, |w| < \lambda,$$

onde  $S_g(w)$  é a densidade espectral de  $g(t)$ .

Concluindo,

$$\begin{aligned} E( |g[n]|^2 + |g(nT_a)|^2 ) &= \\ &= (1/2\lambda) \int_{-\lambda}^{\lambda} S_{gg}(w) dw + (1/2\pi) \int_{-\infty}^{\infty} S_g(w) dw = \\ &= (1/\pi) \int_{-\lambda}^{\lambda} S_g(w) dw, \end{aligned} \quad (3.3)$$

Já que  $S_g(w) = 0, |w| > \lambda$ .

Da equação 3.2 podemos escrever que:

$$h[n] = (1/2\lambda) \int_{-\lambda}^{\lambda} H(e^{jwT_a}) e^{jn\omega T_a} dw = (1/2\lambda) \int_{-\lambda}^{\lambda} H_a(w) e^{jn\omega T_a} dw \quad (3.4)$$

onde  $h[n]$  é a resposta impulsiva de  $H(z)$ .

$$\text{Considerando, } h_{\lambda}(t) = (1/2\pi) \int_{-\lambda}^{\lambda} H_a(w) e^{jn\omega T_a} dw,$$

$$\text{a resposta impulsiva do sistema } H_{\lambda}(w) = \begin{cases} H_a(w), & |w| < \lambda; \\ 0, & \text{fora,} \end{cases}$$

concluimos que:

$$h[n] = T_a h_{\lambda}(nT_a). \quad (3.5)$$

Sabemos que:

$$g[m] = \sum_{k=-\infty}^{\infty} h[k] f[m-k] = \sum_{k=-\infty}^{\infty} T_a h_{\lambda}(kT_a) f(nT_a - kT_a)$$

e,

$$g(nT_a) = \int_{-\infty}^{\infty} h_a(\gamma) f(nT_a - \gamma) d\gamma.$$

Então:

$$\begin{aligned} E\{g(nT_a)g^*[m]\} &= \sum_k T_a h_{\lambda}^*(kT_a) \int_{-\infty}^{\infty} h_a(\gamma) R_f(kT_a - \gamma) d\gamma = \\ &= \sum_k T_a h_{\lambda}^*(kT_a) (1/2\pi) \int_{-\infty}^{\infty} H_a(\omega) S_f(\omega) e^{j\omega kT_a} d\omega = \\ &= (1/2\pi) \int_{-\infty}^{\infty} H_a(\omega) S_f(\omega) \left[ \sum_k T_a h_{\lambda}^*(kT_a) e^{j\omega kT_a} \right] d\omega = \\ &= (1/2\pi) \int_{-\lambda}^{\lambda} H_a(\omega) S_f(\omega) H_a^*(\omega) d\omega = (1/2\pi) \int_{-\lambda}^{\lambda} S_g(\omega) d\omega, \quad (3.6) \end{aligned}$$

onde  $R_f(t)$  é a função de autocorrelação de  $f(t)$ .

Também,

$$\begin{aligned} E\{g[n]g^*(nT_a)\} &= \sum_k T_a h_{\lambda}(kT_a) \int_{-\infty}^{\infty} h_a^*(\gamma) R_f(\gamma - kT_a) d\gamma = \\ &= \sum_k T_a h_{\lambda}(kT_a) \int_{-\infty}^{\infty} h_a^*(\gamma) R_f^*(kT_a - \gamma) d\gamma = \\ &= (1/2\pi) \int_{-\lambda}^{\lambda} H_a^*(\omega) H_a(\omega) S_f(\omega) d\omega = (1/2\pi) \int_{-\lambda}^{\lambda} S_g(\omega) d\omega. \quad (3.7) \end{aligned}$$

Usamos o fato de que  $R_f(-t) = R_f^*(t)$  e portanto  $S_f(\omega)$

é real.

Usando as equações 3.3, 3.6 e 3.7, concluímos que:

$$E( |g(nT_a) - g[n]|^2 ) = 0 .$$

CQD.

Para demonstrarmos a segunda parte do teorema vamos utilizar a desigualdade de Cauchy [33,pg.134]:

$$| \sum_n z_n w_n |^2 \leq \sum_n |z_n|^2 \sum_n |w_n|^2 , \quad z_n \text{ e } w_n \text{ complexos.}$$

Sabemos que:

$$g(t) = \sum_{n=-\infty}^{\infty} g(nT_a) S_a[\lambda(t-nT_a)] ,$$

$$\hat{g}(t) = \sum_{n=-\infty}^{\infty} g[n] S_a[\lambda(t-nT_a)] .$$

Mas,

$$\begin{aligned} |g(t) - \hat{g}(t)|^2 &= \left| \sum_{n=-\infty}^{\infty} [g(nT_a) - g[n]] S_a[\lambda(t-nT_a)] \right|^2 \leq \\ &\leq \sum_{n=-\infty}^{\infty} |g(nT_a) - g[n]|^2 \sum_{n=-\infty}^{\infty} |S_a[\lambda(t-nT_a)]|^2 . \end{aligned}$$

Então,

$$E \{ |g(t) - \hat{g}(t)|^2 \} \leq \sum_{n=-\infty}^{\infty} E \{ |g(nT_a) - g[n]|^2 \} \sum_{n=-\infty}^{\infty} |S_a[\lambda(t-nT_a)]|^2 .$$

Mas, se considerarmos  $f_1(t) = \exp(jwt)$ ,  $|w| \leq \lambda$ , uma função periódica, vem:

$$f_1(t) = \sum_{m=-\infty}^{\infty} a_m e^{jmT_a t}, \quad T_a = \pi/\lambda, \quad e$$

$$a_m = S_a[\lambda(t - mT_a)].$$

Da fórmula de Parseval [33, pg.72],

$$\sum_{m=-\infty}^{\infty} |S_a[\lambda(t - mT_a)]|^2 = (1/2\lambda) \int_{-\lambda}^{\lambda} |f_1(t)|^2 dt = 1.$$

Logo,  $E(|g(t) - \hat{g}(t)|^2) \leq 0$ , o que implica que

$$E(|g(t) - \hat{g}(t)|^2) = 0.$$

CQD.

### 3.5 Simulação dos filtros de transmissão e recepção.

De acordo com a seção 3.2, a representação complexa permite que os filtros de transmissão e recepção possam ser simulados como filtros passa-baixa.

Da equação 3.4 podemos observar que se fizermos  $\lambda = \pi/T_a$ , suficientemente grande,  $h[n]$  (equação 3.5) pode ser aproximada por,

$$h[n] = T_a h_\lambda(nT_a) \cong T_a h_a(nT_a) \quad (3.8),$$

onde  $h_a(t)$  é a resposta impulsiva de  $H_a(w)$ .

Filtros passa-baixa ideais podem ser aproximados pela função de transferência [34]:

$$H_a(s) = \frac{Gb_0}{s^m + b_{m-1}s^{m-1} + \dots + b_1s + b_0}$$

onde  $G$ ,  $b_0$ ,  $b_1$ ,  $\dots$ ,  $b_m$  são constantes apropriadas.

Expandindo  $H_a(s)$  em frações parciais e obtendo a transformada de Laplace inversa, vem :

$$h_a(t) = \sum_{k=1}^N A_k \exp(s_k t) U(t)$$

onde  $U(t)$  é a função degrau unitário, isto é:

$$U(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 ; \end{cases}$$

$s_k$  são os pólos da função de transferência e  $A_k$  são coeficientes adequados.

Utilizando a equação 3.8 obtemos:

$$h[n] = \sum_{k=1}^N A_k T_a \exp(s_k n T_a) U[n] \quad (3.9)$$

onde  $U[n]$  é a sequência degrau,  $U[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0. \end{cases}$

Aplicando a transformada Z a ambos os lados da equação 3.9, vem ,

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=1}^N \frac{T_a A_k}{1 - \exp(s_k T_a) z^{-1}}$$

ou,

$$Y(z) = \sum_{k=1}^N Y_k(z) ,$$

onde  $Y_k(z) = T_a A_k X(z) + \exp(s_k T_a) Y_k(z) z^{-1}$ .

Portanto,

$$y[m] = \sum_{k=1}^N y_k[m] ,$$

onde  $y_k[n]$  é a transformada inversa de  $Y_k(z)$ .

As células básicas  $Y_k(z)$  são usadas para simularmos os filtros passa-baixa como mostra a figura 3.5.

Basta então, obtermos os valores dos coeficientes  $A_k$  e  $s_k$  para o filtro passa-baixa considerado.

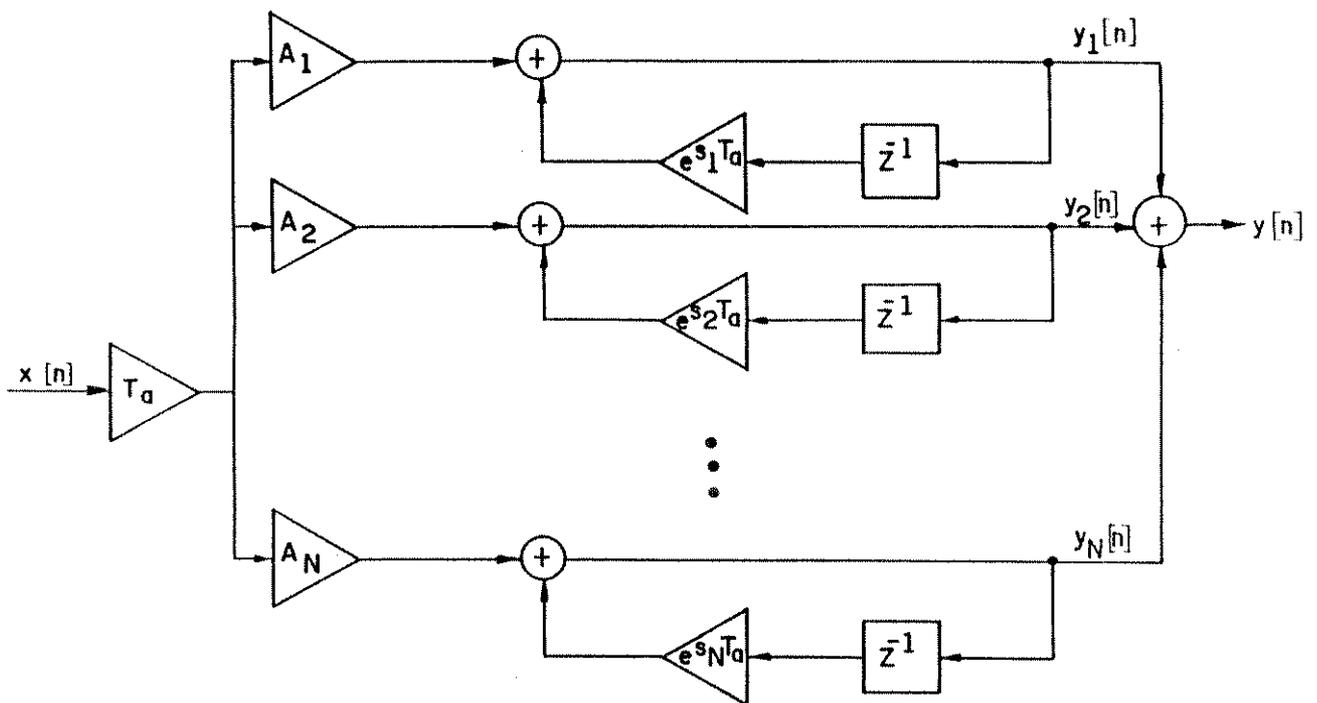


fig. 3.5 Diagrama dos filtros digitais passa-baixa.

### 3.5.1 Obtenção dos coeficientes e pólos para os filtros Chebyshev.

O módulo da função de transferência para os filtros Chebyshev passa-baixa, com comportamento "equiripple" na faixa de passagem, é dado por [35]:

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \varepsilon^2 V_N^2(\Omega/\Omega_c)}$$

onde  $V_N(x) = \cos(N \cos^{-1} x)$  é um polinômio de Chebyshev de ordem  $N$ , que pode ser obtido por uma equação recursiva [35] e  $\varepsilon$  e  $\Omega_c$  são constantes.

$$\text{Para } \Omega = 0 \text{ temos: } |H_a(j\Omega)| = \begin{cases} 1, & N \text{ ímpar} \\ 1/\sqrt{1+\varepsilon^2}, & N \text{ par.} \end{cases}$$

$$\text{Para } \Omega = \Omega_c, |H_a(j\Omega)| = 1/\sqrt{1+\varepsilon^2}.$$

O comprimento de "ripple" é definido em decibéis por [34]:  $R_{dB} = 20 \log_{10} \sqrt{1 + \varepsilon^2} = 10 \log_{10} (1 + \varepsilon^2)$ ; e é o que caracteriza o filtro. Note que  $\Omega_c$  é a frequência de 3dB somente para um filtro com "ripple" de 3dB. Estamos considerando como frequência de 3dB o valor de frequência no qual  $|H_a(j\Omega)|^2$  é igual a 1/2 vezes o seu valor máximo [34]. No nosso caso o valor máximo de  $|H_a(j\Omega)|^2$  é a unidade.

A localização dos pólos,  $s_i = \sigma_i + j \omega_i$ , é dada por [35]:

$$\sigma_i = -\Omega_c \sinh(a) \sin\left[\frac{(2i-1)}{N} \left(\frac{\pi}{2}\right)\right]$$

$$\omega_i = +\Omega_c \cosh(a) \cos\left[\frac{(2i-1)}{N} \left(\frac{\pi}{2}\right)\right], \quad i=1, 2, \dots, N,$$

onde,

$$a = (1/N) \operatorname{senh}^{-1}(1/\epsilon) = \ln(\epsilon^{-1} + \sqrt{1 + \epsilon^{-2}}).$$

Assim, obtemos uma expressão modificada para  $H_a(s)$ :

$$H_a(s) = \frac{GN}{\prod_{i=1}^N (s - s_i)} \quad .$$

Daf derivamos os valores de GN e  $A_k$ :

$$GN = \left[ \prod_{i=1}^N (s - s_i) \right] H_a(s) = \left| \prod_{i=1}^N (s - s_i) \right|_{s=j\Omega_c} (1/\sqrt{1 + \epsilon^2}),$$

$$A_k = H_a(s) (s - s_k) \Big|_{s=s_k} \quad .$$

Na simulação, só foram utilizados filtros Chebyshev.

Considere novamente  $u(t) = \sum_{n=0}^{\infty} I_n g(t - nT)$ , como sendo uma sequência de símbolos enviados através de um sistema cuja resposta impulsiva é  $h(t)$ . Na saída do sistema teremos:

$$\begin{aligned} v(t) &= \int_{-\infty}^{\infty} u(\alpha) h_a(t - \alpha) d\alpha = \\ &= \sum_{n=0}^{\infty} I_n \int_{-\infty}^{\infty} g(\alpha - nT) h_a(t - \alpha) d\alpha = \sum_{n=0}^{\infty} I_n q(t - nT), \end{aligned}$$

onde,

$$q(t) = \int_{-\infty}^{\infty} g(\alpha) h_a(t - \alpha) d\alpha.$$

Pela expressão acima para  $q(t)$  podemos concluir que a duração da resposta ao impulso dos filtros que compõem o canal é que nos dá a extensão da interferência intersimbólica existente no mesmo. Se  $h(t) = K \delta(t)$ , onde  $\delta(t)$  é um impulso unitário,

então,  $q(t) = K g(t)$  e não há distorção introduzida.

A figura 3.6 mostra a resposta ao impulso do canal de maior interferência, usado nas simulações. Os filtros de transmissão e recepção são idênticos: Chebyshev, 0,5 dB "ripple", com  $N=4$  pólos. O produto  $2BT$  considerado é igual 1,2. O valor de  $B$  é a frequência de 3dB do filtro de recepção (passa-baixa). Como mostra a figura, a extensão da interferência intersimbólica é de aproximadamente 5 símbolos (estamos desprezando a influência da resposta ao impulso para valores abaixo de 10% do seu valor máximo). Esta duração será importante na definição das sequências pseudo-aleatórias utilizadas para gerar os bits de informação como veremos na próxima seção.

### 3.6 Geração dos dígitos de informação.

Nós utilizamos uma sequência binária pseudo-aleatória para gerar os bits de informação.

A geração da sequência pseudo-aleatória é feita através de um registrador de deslocamento linear; isto é, o dígito binário utilizado na realimentação do registrador é o resultado de somas módulo dois de alguns ou todos os  $n$  dígitos binários do registrador [36].

A sequência é de comprimento máximo: o período é igual a  $(2^n - 1)$ , onde,  $n$  é o tamanho do registrador. A referência [36] traz uma descrição da lógica de realimentação para vários valores de  $n$ .

A pergunta que se coloca é por que não utilizarmos o mesmo gerador de números pseudo-aleatórios usado no capítulo

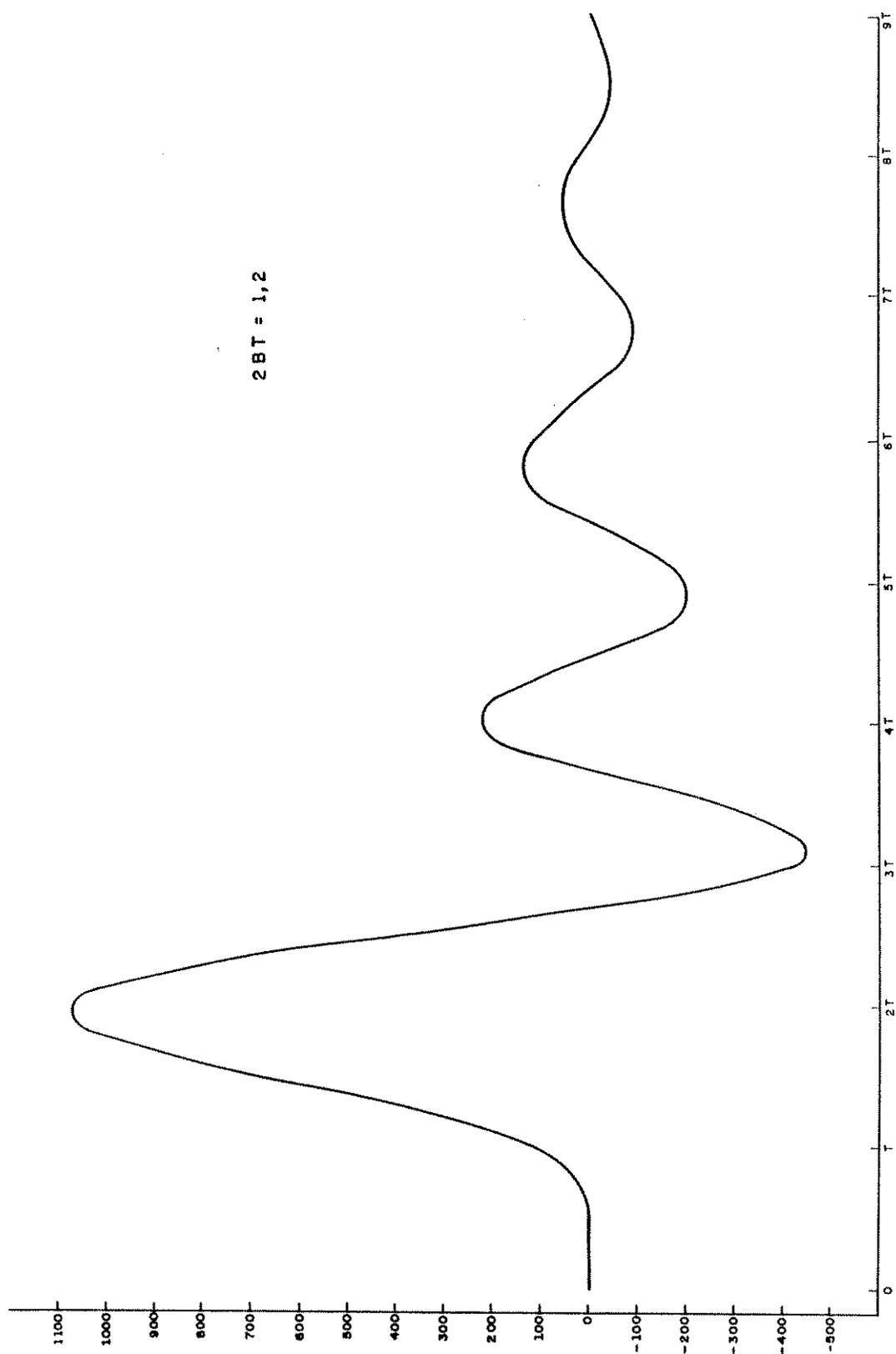


fig.3.6 Resposta ao impulso do canal ( $2BT = 1,2$ ).

dois ?

Por duas razões:

1. O registrador de deslocamento nos dá controle sobre os padrões de interferência intersimbólica possíveis no canal. Por exemplo: se utilizarmos um registrador de tamanho  $n$ , todos os possíveis padrões de  $n$  bits (a menos do padrão em que todos os bits são nulos e de alguns no começo e no final da sequência) são gerados após completar-se um período de tamanho  $(2^M - 1)$ . Isto irá garantir uma mesma frequência de ocorrência para os padrões de interferência.

2. Como a sequência gerada é periódica, todo o processamento feito para obtenção das amostras dos sinais antes da adição de ruído, não precisará ser repetido, após termos gerado um período da mesma.

Se considerarmos que a duração da interferência intersimbólica é de 5 símbolos de canal e que o sistema codificado é um 8-PSK (dois bits de informação por símbolo), uma sequência de período igual a  $[2^{(2 \times 5 + 1)} - 1] = (2^{11} - 1)$  será suficiente para a geração de quase a totalidade dos padrões de interferência.

### 3.7 A simulação do amplificador não linear.

Para simular o amplificador do "transponder", nós escolhemos um amplificador de potência TWT (do inglês: Traveling Wave Tube). O comportamento do TWT tem a tendência de dominar o comportamento do "transponder" [30]. A figura 3.7

mostra curvas de transferência típicas de amplitude e fase para um TWT [29]. Como vemos, o TWT é um amplificador não linear. As curvas são independentes da frequência da portadora dentro da faixa de frequências considerada. Sendo assim, o TWT simulado pode ser considerado uma não linearidade, passa-faixa e sem memória.

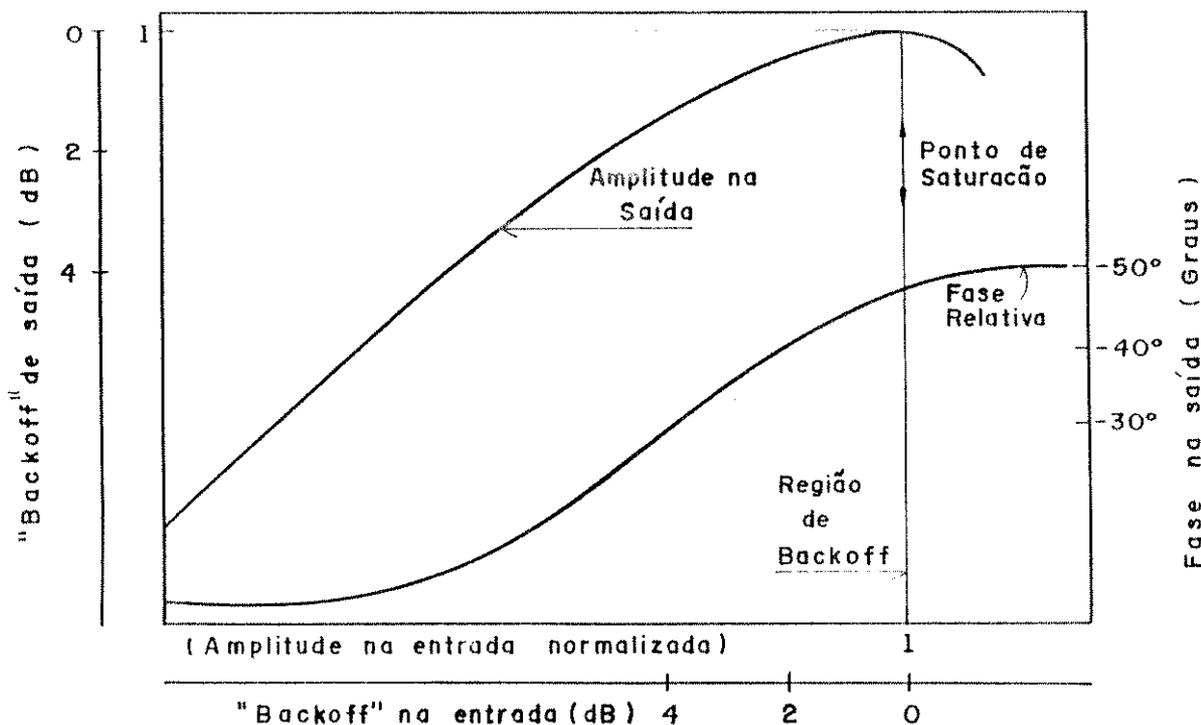


fig. 3.7 Curvas típicas de um TWT.

As curvas de transferência relacionam a amplitude (indiretamente a potência) e fase instantâneas do sinal de saída com a amplitude instantânea do sinal de entrada.

Em geral, define-se como "backoff" de entrada a diferença (em decibéis) entre a potência média do sinal de entrada modulado e a potência média de um sinal não modulado que

leva o TWT à trabalhar no ponto de saturação (figura 3.7). Consideramos um "backoff" de entrada positivo, quando os valores de entrada forem menores que o valor no ponto de saturação. O "backoff" de saída é definido de maneira análoga em relação ao valor de amplitude na saída correspondente ao ponto de saturação.

No programa de simulação nós definimos o "backoff" de entrada não em relação à potência média do sinal modulado, mas em relação à potência média do sinal não modulado. Para levar o TWT à trabalhar no ponto de operação desejado, o programa calcula um ganho  $K_a$  sobre o sinal não modulado. Esse mesmo ganho é aplicado ao sinal modulado. A potência de saída é normalizada para 1 (um) watt.

Para simular o amplificador TWT nós utilizamos o modelo em quadratura obtido da referência [37]. A figura 3.8 mostra o modelo.

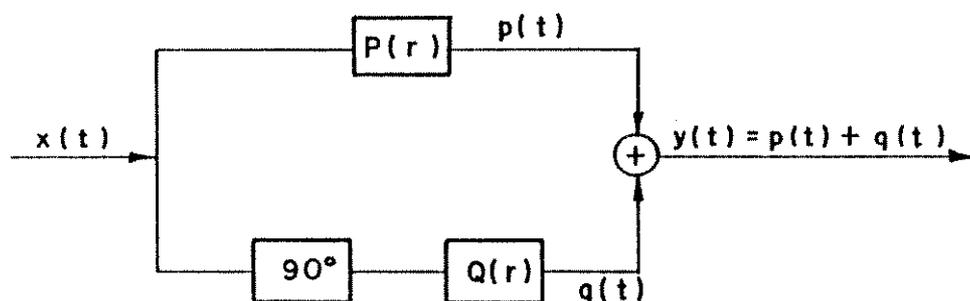


fig. 3.8 Modelo em quadratura do TWT.

Se a entrada é dada por

$$x(t) = r(t) \cos(\omega_0 t + \psi(t)),$$

a saída é a soma das componentes em fase (  $p(t)$  ) e quadratura (  $q(t)$  ),

$$\begin{aligned} p(t) &= P[r(t)] \cos(\omega t + \psi(t)) \\ q(t) &= -Q[r(t)] \sin(\omega t + \psi(t)) = \\ &= Q[r(t)] \cos(\omega t + \psi(t) + \pi/2). \end{aligned}$$

As funções  $P(r)$  e  $Q(r)$  são expressas por:

$$P(r) = \alpha_p r / (1 + \beta_p r^2) \quad (3.10a)$$

$$Q(r) = \alpha_q r / (1 + \beta_q r^2)^2 \quad (3.10b)$$

Como a saída pode ser escrita na forma:

$$y(t) = A[r(t)] \cos(\omega t + \psi(t) + \Phi[r(t)]),$$

deduzimos as seguintes relações:

$$P[r(t)] = A[r(t)] \cos(\Phi[r(t)])$$

$$Q[r(t)] = A[r(t)] \sin(\Phi[r(t)]).$$

Assim,  $P$  e  $Q$  podem ser obtidas das curvas experimentais de  $A$  e  $\Phi$ .

Os parâmetros  $\alpha_p$ ,  $\alpha_q$ ,  $\beta_p$  e  $\beta_q$  foram obtidos em [37]. Eles estão otimizados, no sentido de minimizar o erro médio quadrático, ao ajustar as equações 3.10a e 3.10b às curvas experimentais.

A representação complexa do sinal  $y(t)$  na saída do TWT, é dada por:

$$\begin{aligned} \tilde{y}(t) &= \tilde{p}(t) + \tilde{q}(t) = \\ &= P[r(t)] \cos[\psi(t)] + j P[r(t)] \sin[\psi(t)] - \\ &\quad - Q[r(t)] \sin[\psi(t)] + j Q[r(t)] \cos[\psi(t)] = \\ &= ( P[r(t)] \cos[\psi(t)] - Q[r(t)] \sin[\psi(t)] ) + \\ &\quad + j ( P[r(t)] \sin[\psi(t)] + Q[r(t)] \cos[\psi(t)] ). \end{aligned}$$

Como  $x(t) = r(t) \cos[\omega t + \psi(t)]$ , a envoltória

complexa da entrada é:

$$\tilde{x}(t) = r(t) \cos[\psi(t)] + j r(t) \text{sen}[\psi(t)] = \text{Re}\tilde{x}(t) + j\text{Im}\tilde{x}(t).$$

Então,  $r(t)$ ,  $\cos[\psi(t)]$  e  $\text{sen}[\psi(t)]$ , são facilmente obtidos:

$$r(t) = \sqrt{\text{Re}^2[\tilde{x}(t)] + \text{Im}^2[\tilde{x}(t)]},$$

$$\cos[\psi(t)] = \text{Re}\tilde{x}(t)/r(t) \text{ e } \text{sen}[\psi(t)] = \text{Im}\tilde{x}(t)/r(t).$$

Como é do nosso conhecimento o defasamento  $\psi_d$  introduzido pelo TWT no ponto de operação desejado, nós diminuímos de cada amostra do sinal na saída do TWT, um valor de fase (indiretamente  $P_d$  e  $Q_d$ ) igual ao valor  $\psi_d$ . Isto é o correspondente a rotacionar os sinais M-PSK de referência do receptor do mesmo valor  $\psi_d$ .

### 3.8 Os filtros casados.

Considere a seguinte propriedade:

#

Propriedade. Para  $k$  um inteiro qualquer, a integral

$$I = \int_{kT+t_0}^{(k+1)T+t_0} r(t) \psi(t) dt,$$

onde,

$$\psi(t) = \psi_0(t) g(t - (kT + t_0))$$

e  $g(t)$  é o pulso unitário definido na seção 1.3,

é equivalente à:

$$I = y(t) \Big|_{t=(k+1)T+t_0} = \int_0^t r(t-\alpha) \psi((k+1)T+t_0-\alpha) d\alpha \Big|_{t=(k+1)T+t_0}$$

Ou seja,  $I$  é a saída  $y(t)$ , no instante  $t = (k+1)T + t_0$ ,

de um filtro cuja resposta ao impulso é  $\varphi((k+1)T + t_0 - t)$ .  
Esse filtro é conhecido como filtro casado [10].

#  
Demonstração.

$$\int_0^t r(t-\tau) \varphi((k+1)T + t_0 - \tau) d\tau =$$

$$= \int_{(k+1)T + t_0 - t}^{(k+1)T + t_0} r(t + \alpha - (k+1)T - t_0) \varphi(\alpha) d\alpha = y(t).$$

Então,

$$y(t) \Big|_{t=(k+1)T + t_0} = \int_0^{(k+1)T + t_0} r(\alpha) \varphi(\alpha) d\alpha.$$

Como

$$\varphi(\alpha) = \varphi_0(\alpha) g(\alpha - (kT + t_0)) = \begin{cases} \varphi_0(\alpha), & kT + t_0 \leq \alpha < (k+1)T + t_0, \\ 0, & \text{fora;} \end{cases}$$

então:

$$I = y(t) \Big|_{t=(k+1)T + t_0}.$$

CQD.

#

Assim, as coordenadas do k-ésimo símbolo de canal na entrada do decodificador serão obtidas das saídas dos filtros casados.

A maneira como foi obtida a propriedade acima, nos

permite distinguir dois problemas na recepção:

1. Como escolher o instante  $t_0$  ?
2. O que é preciso conhecer em  $\varphi_0(t)$  ?

A escolha do instante  $t_0$  é conhecida como sincronismo temporal. Ela será tratada no item 3.9.

Para os sinais M-PSK,

$$\varphi_0(t) = \sqrt{2/T} \cos(\omega_0 t + \phi_n),$$

ou,

$$\varphi_0(t) = -\sqrt{2/T} \operatorname{sen}(\omega_0 t + \phi_n).$$

O problema mais crítico é a recuperação da fase  $\phi_r$ .

Assumiremos que o receptor conhece a frequência  $\omega_0$  com precisão.

Vamos assumir também que  $\phi_r$  é recuperada pelo receptor. Um circuito de recuperação de fase PLL (do inglês: "Phase Locked Loop") irá neutralizar parcialmente os efeitos da não linearidade do TWT [39,40]. Então, ao assumirmos uma recuperação da fase  $\phi_r$ , estaremos, portanto, obtendo resultados de desempenho com degradação maior ou igual a de um sistema real.

### 3.8.1 Representação Complexa.

Sem perda de generalidade podemos considerar a fase  $\phi_r$  da portadora não modulada, como sendo nula.

$$\text{Para } \varphi_0(t) = \sqrt{2/T} \cos[\omega_0 t],$$

$$\varphi_0((k+1)T + t_0 - t) = \sqrt{2/T} \cos[\omega_0 t - \phi_0],$$

onde  $\phi_0 = \omega_0 t_0$  e  $\omega_0 = 2\pi l/T$ ,  $l$  inteiro.

Mas,

$$\psi((k+1)T + t_0 - t) = \psi_0((k+1)T + t_0 - t) g(T-t) = h_p(t).$$

Então, a representação complexa da resposta ao impulso,  $\tilde{h}_p(t)$ , será:

$$\tilde{h}_p(t) = \sqrt{1/2T} e^{-j\phi_0}, \quad 0 \leq t < T.$$

Para  $\psi_0(t) = -\sqrt{2/T} \text{sen}(\omega_0 t)$ ,

$$\psi_0((k+1)T + t_0 - t) = -\sqrt{2/T} \text{sen}(\phi_0 - \omega_0 t) = \sqrt{2/T} \cos(\omega_0 t - (\phi_0 + \pi/2)).$$

Então:

$$\tilde{h}_p(t) = \sqrt{1/2T} e^{-j(\phi_0 + \pi/2)} = -\sqrt{1/2T} j e^{j\phi_0}, \quad 0 \leq t < T.$$

Considere que  $\tilde{h}_p'(t) = \exp(j\phi_0) \tilde{h}_p(t)$ ;

$\tilde{y}'(t)$  como sendo a resposta complexa para  $\tilde{h}_p'(t)$ ;

e que  $y(t)$  é a resposta para  $h_p(t)$ . Então:

$$y(t) \Big|_{t=(k+1)T+t_0} = \text{Re} [\tilde{y}(t) e^{j\omega_0 t}] \Big|_{t=(k+1)T+t_0} =$$

$$= \text{Re} [\tilde{y}'(t) e^{-j\phi_0} e^{j\omega_0 t}] \Big|_{t=(k+1)T+t_0} =$$

$$= \text{Re} [\tilde{y}'(t)] \cos(\omega_0 t - \phi_0) - \text{Im} [\tilde{y}'(t)] \text{sen}(\omega_0 t - \phi_0) \Big|_{t=(k+1)T+t_0} =$$

$$= \operatorname{Re} [\tilde{y}'(t)] \Big|_{t=(k+1)T+t_0} \cdot$$

Ou seja, se usarmos

$$h_{\varphi}(t) = 1/\sqrt{2T} (-j/\sqrt{2T}), \quad 0 \leq t < T,$$

as partes reais da saída dos filtros complexos em  $t=(k+1)T + t_0$  serão as coordenadas do vetor recebido.

Para efeito de simulação, é necessário apenas projetarmos um filtro com resposta impulsiva  $h_{\varphi}(t) = 1$ ,  $0 \leq t < T$ . A parte real da saída multiplicada por  $[1/\sqrt{2T}]$  será a coordenada em fase do vetor recebido. Como  $\operatorname{Re}[-jX] = \operatorname{Im}[X]$ , a parte imaginária da saída multiplicada por  $[1/\sqrt{2T}]$  será a coordenada em quadratura do vetor recebido.

Para simularmos o filtro de resposta impulsiva finita do parágrafo anterior, nós utilizamos novamente a equação 3.8:

$$h[n] = T_a h_{\varphi}(nT_a) = T_a, \quad 0 \leq t < T,$$

onde  $T_a = T/\lambda$  é suficientemente pequeno para que a aproximação da equação 3.8 seja válida.

Se fizermos  $T_a = T/N_a$ , a saída do filtro casado digital será:

$$\tilde{y}[n] = \sum_{k=0}^{N_a-1} h[k] \tilde{f}[n-k] = T_a \sum_{k=0}^{N_a-1} \tilde{f}[n-k],$$

$\tilde{f}[n]$  representando as amostras do sinal na entrada do filtro casado.

### 3.9 A escolha de $t_0$ .

Como vimos anteriormente, a cada símbolo obtido na saída do filtro casado, está associado um início de integração  $t_0$ .

Um critério de escolha do instante  $t_0$  seria o de utilizar um método de "força bruta". Escolher um ponto de operação (uma determinada relação sinal ruído e um produto BT). Simular o sistema com  $t_0$  igual a  $kT_a$ ,  $0 \leq k < N_a$  e  $T_a = T/N_a$ , onde  $N_a$  é o número de amostras por símbolo. Por exemplo, se  $N_a = 16$ , precisaríamos simular o sistema 16 vezes. Utilizar como  $t_0$  aquele que nos levou a uma probabilidade de erro de bit mínima. Repetir o processo quando variarmos o ponto de operação, ou, o produto BT apenas.

Em [41] e [42], o procedimento de estimação de  $t_0$  é feito de uma maneira "idealizada". As amostras dos sinais na saída do filtro de recepção (sem ruído) são observadas para determinar-se a posição média de cruzamento do zero [o termo cruzamento do zero refere-se a mudança de sinal ou da componente de fase ou da componente em quadratura das amostras complexas observadas]. Esta média é calculada sobre todas as amostras da sequência pseudo-aleatória periódica utilizada. Este valor médio é escolhido como o início de cada símbolo.

Nós optamos pelo critério de maximização da correlação entre o sinal transmitido e o recebido [8].

Seja  $S^1(t)$  o sinal que representa um conjunto de símbolos transmitidos. Seja  $S^2(t)$  o sinal que representa o conjunto de símbolos na saída do filtro de recepção. Seja  $N$  o

número total de símbolos transmitidos.

O nosso objetivo é encontrar um valor de  $\tau$ , para o qual,

$$\phi_{1,2} = (1/NT) \int_{NT} S^1(t) S^2(t+\tau) dt,$$

é máximo.

Considerando-se  $S^1(t)$  e  $S^2(t)$  sinais de faixa estreita, pode-se mostrar que [38]:

$$\phi_{1,2} = \text{Re}[\tilde{\phi}] = \text{Re}[(1/2NT) \int_{NT} \tilde{S}^1(t) S^{2*}(t+\tau) dt],$$

onde " $\sim$ " indica a representação complexa e

"\*" denota o complexo conjugado.

Podemos aproximar  $\tilde{\phi}$  por:

$$\tilde{\phi} \cong (1/2NT) T_a \sum_{k=0}^{NN_a-1} \tilde{S}^1(kT_a) S^{2*}(kT_a+\tau),$$

onde  $\tau$  é agora um múltiplo inteiro de  $T_a$ ,

e  $N_a$  representa o número de amostras por símbolo.

Nas simulações, para obtermos o instante ótimo, nós utilizamos um valor de  $N = 900$ .

### 3.10 A simulação do ruído.

Seja  $n(t)$  o ruído gaussiano aditivo e branco.

As amostras do ruído, nas saídas dos filtros casados, podem ser expressas por:

$$n_x(t) = \int_{-\infty}^{\infty} n(\alpha) h_x(t-\alpha) d\alpha$$

$$n_y(t) = \int_{-\infty}^{\infty} n(\beta) h_y(t-\beta) d\beta,$$

onde  $h_x(t)$  ( $h_y(t)$ ) é a resposta ao impulso da junção do filtro de recepção e o filtro casado da coordenada  $x$  ( $y$ ).

As variáveis aleatórias  $n_x(t)$  e  $n_y(t)$  serão gaussianas de média nula e variância:

$$\sigma_{x/y}^2 = (1/2\pi) \int_{-\infty}^{\infty} (N_0/2) |H_{x/y}(w)|^2 dw. \quad (3.11)$$

onde  $H_x(w)$  ( $H_y(w)$ ) é a transformada de Fourier de  $h_x(t)$  ( $h_y(t)$ ).

Sem perda de generalidade, considere que:

$H_R(w)$  é a função de transferência do filtro de recepção;

$$H_{FC}^x(w) = (K/2) e^{-jwT} [\text{Sa}(w+w_0) + \text{Sa}(w-w_0)], \text{ e}$$

$$H_{FC}^y(w) = j(K/2) e^{-jwT} [\text{Sa}(w+w_0) - \text{Sa}(w-w_0)],$$

sendo  $K$  uma constante e  $w_0$  a frequência da portadora, são as funções de transferência dos filtros casados.

A correlação entre  $n_x(t)$  e  $n_y(t)$  é dada por:

$$E\{n_x(t) n_y(t)\} =$$

$$= (N_0/2) \int_{-\infty}^{\infty} h_x(\alpha) h_y(\alpha) d\alpha =$$

$$= (N_0/2) (1/2\pi) \int_{-\infty}^{\infty} H_x(w) H_y^*(w) dw =$$

$$= (N_0/2) (1/2\pi) \int_{-\infty}^{\infty} |H_R(w)|^2 H_{FC}^x(w) H_{FC}^{y*}(w) dw = 0.$$

Portanto, como as amostras são gaussianas, elas serão também independentes.

Devido à ação do filtro de recepção, amostras

sucessivas de ruído (separadas de um intervalo  $T$ : a duração do símbolo de canal) devem ser ligeiramente correlacionadas. Simulações anteriores [31], têm mostrado que esta correlação é desprezível.

Sendo assim, se somarmos amostras de ruído gaussiano branco, com variância apropriada (equação 3.11), às amostras dos símbolos na saída do filtro casado, da mesma maneira como fizemos para o canal AWGN, a influência nos resultados é desprezível. Isto simplifica extremamente a simulação. Podemos então: a) obter os valores dos símbolos na saída do filtro casado para apenas um período da sequência pseudo-aleatória; b) armazenarmos estes valores; e; c) posteriormente somarmos estes valores a um determinado número de conjuntos de amostras de ruído branco de maneira que possamos estimar a probabilidade de erro de bit com a confiabilidade desejada.

O cálculo da equação 3.11 pode ser feito de maneira mais simplificada, trasladando-se o espectro das funções de transferência para  $w = 0$ .

### 3.11 A relação sinal ruído $E_b/N_0$ .

Na seção 1.4, nós havíamos definido a energia média por bit de informação como sendo:  $E_b = E_s / \log M$ , onde  $E_s$  era a energia média dos sinais na saída do modulador. Como o canal AWGN é modelado pela simples adição de ruído, o mesmo valor de  $E_b$  estará disponível na entrada do receptor (filtros casados).

Para o canal de satélite, será preciso calcular o valor de  $E_b$  na saída do filtro de recepção (ou equivalentemente a

entrada dos filtros casados).

Considere que o sinal na saída do filtro de recepção é da forma:

$$r(t) = A(t) \cos(\omega_0 t + \theta(t)).$$

A energia de  $r(t)$  é dada por:

$$E_r = \int_{-\infty}^{\infty} r^2(t) dt = \int_{-\infty}^{\infty} (1/2) A^2(t) dt + \\ + \int_{-\infty}^{\infty} (1/2) A^2(t) \cos(2\omega_0 t + 2\theta(t)) dt \cong (1/2) \int_{-\infty}^{\infty} A^2(t) dt,$$

Já que o termo de segunda harmônica pode ser desprezado.

Para um sinal de duração limitada  $T_r$ , a potência média é definida por [10]:

$$P_r = (1/T_r) \int_0^{T_r} r^2(t) dt = E_r/T_r.$$

Como estamos trabalhando com amostras do sinal, espaçadas de intervalos iguais a  $T_a$ , podemos escrever:

$$P_r \cong (1/NT) \sum_{k=0}^{NN_a-1} (1/2) A^2(kT_a) = (1/2NN_a) \sum_{k=0}^{NN_a-1} [x^2(kT_a) + y^2(kT_a)]$$

onde  $N$  é o número total de símbolos transmitidos,

$T$  é a duração de um símbolo ( $NT = T_r$ ), e,

$x$  e  $y$  são, respectivamente, a parte real e imaginária da representação complexa do sinal  $r(t)$ .

Sendo assim, podemos obter o valor de  $E_b$  da expressão:

$$E_b = P_r T_b = P_r T / \log_2 M,$$

onde  $T_b$  é o tempo de duração de um bit.

Na simulação, nós utilizamos para definir o valor de  $E_b$ , não a potência média recebida como foi descrito acima, mas a potência de uma portadora não modulada na saída do filtro de recepção. Como vimos na seção 3.7, a potência da portadora não modulada na saída do TWT, foi normalizada para 1 watt. Então, a potência na saída do filtro será:

$$Pr' = (\sqrt{2})^2 / 2(1 + \epsilon^2) = 1 / (1 + \epsilon^2) ,$$

onde  $\epsilon$  é o fator de "ripple" (seção 3.5.1).

O uso desta potência na definição de  $E_b$  é apropriada para a operação do TWT na região não linear onde a potência tem uma limitação num valor de pico [7]. Em sistemas lineares, a potência média  $Pr$  deveria ser utilizada. O valor de  $Pr$  é geralmente menor do que a potência da portadora não modulada de aproximadamente 0,5 dB [42].

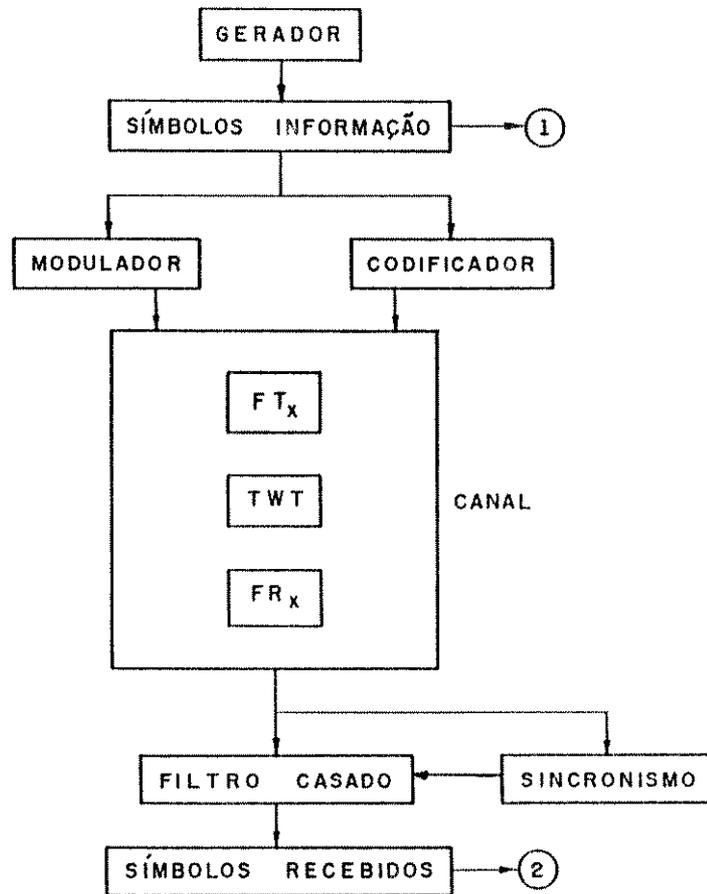
### 3.12 A simulação e os resultados.

A figura 3.9 mostra um diagrama de blocos onde podemos observar a conexão entre as diversas subrotinas que compõem o programa de simulação. A divisão em dois estágios, ressalta o aspecto de simulação híbrida [31,42]: a) o estágio 1 fornece a sequência periódica de símbolos obtidos na saída do filtro casado. b) o estágio 2 adiciona o ruído a esta sequência e estima a probabilidade de erro através da simulação do receptor.

No estágio 1 a simulação segue os seguintes passos:

1. Geração da sequência pseudo-aleatória dos símbolos de informação (GERADOR). Armazenamento num arquivo de dados.

Estágio 1.



Estágio 2.

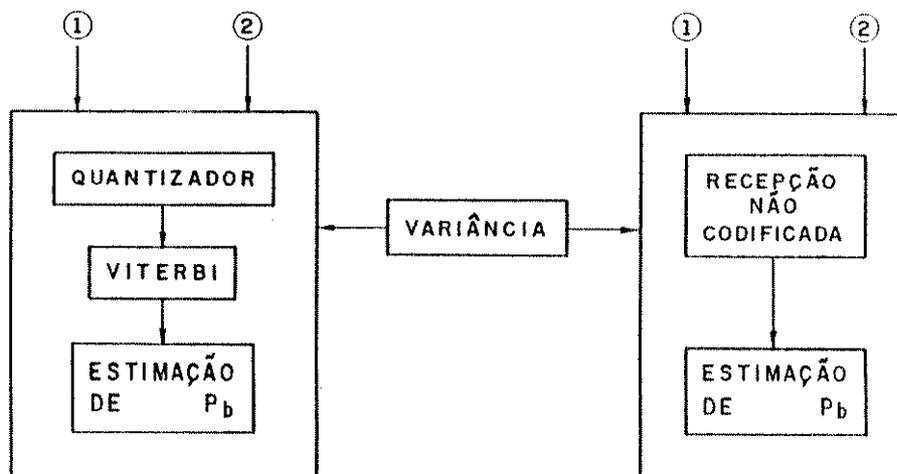


fig. 3.9 Diagrama de blocos do programa de simulação.

2. Obtenção das amostras dos símbolos de canal na saída do filtro de recepção (FRx). Se o sistema é codificado, usa-se a subrotina CODIFICADOR. Caso contrário, a subrotina MODULADOR é utilizada.

3. Obtenção do sincronismo temporal (seção 3.9) através da subrotina SINCRO.

4. E finalmente, a obtenção dos símbolos de canal distorcidos, utilizando-se a subrotina FILTRO CASADO. Armazenamento num arquivo de dados.

O estágio 2 se constitui de três subrotinas principais:

a) a estimação da probabilidade de erro de bit ( $P_b$ ), através do receptor de máxima verossimilhança, para o sistema não codificado.

b) a estimação de  $P_b$  através do algoritmo de Viterbi com recepção não quantizada.

c) a estimação de  $P_b$  através do algoritmo de Viterbi com recepção quantizada.

Todas as subrotinas utilizam como dados de entrada, os símbolos de informação e os símbolos de canal armazenados no estágio 1, além da variância na saída do filtro casado, obtida da equação 3.11 e calculada pela subrotina VARIANCIA.

Em todas as simulações nós utilizamos o código com memória  $m=2$  para a constelação 8-PSK. O valor do comprimento de decodificação utilizado foi de 24 bits. O valor do "backoff" de entrada do TWT foi sempre fixado em 0 dB (ponto de saturação). Os filtros de transmissão e recepção foram sempre idênticos: Chebyshev, 0,5 dB "ripple", com  $N=4$  pólos.

A figura 3.10 mostra os resultados para um produto  $2BT = 2,0$ . Podemos observar que o efeito da interferência intersimbólica é pequeno: a degradação do sistema não codificado 4-PSK em relação ao desempenho no AWGN, é próxima de 1 dB. Este resultado concorda com resultados anteriores obtidos por [44]. Para valores de  $P_b = 10^{-2}$  e  $10^{-3}$ , o ganho de codificação é ligeiramente menor do que no AWGN, mas para um valor de  $P_b = 10^{-4}$  o ganho de codificação se mantém inalterado.

A figura 3.11 mostra os resultados para um produto  $2BT = 1,5$ . O ganho de codificação em  $P_b = 10^{-2}$  se torna negativo. Mas para valores de  $P_b = 10^{-3}$  e  $P_b = 10^{-4}$ , os ganhos de codificação voltam a ser os mesmos obtidos anteriormente para um produto  $2BT = 2,0$ .

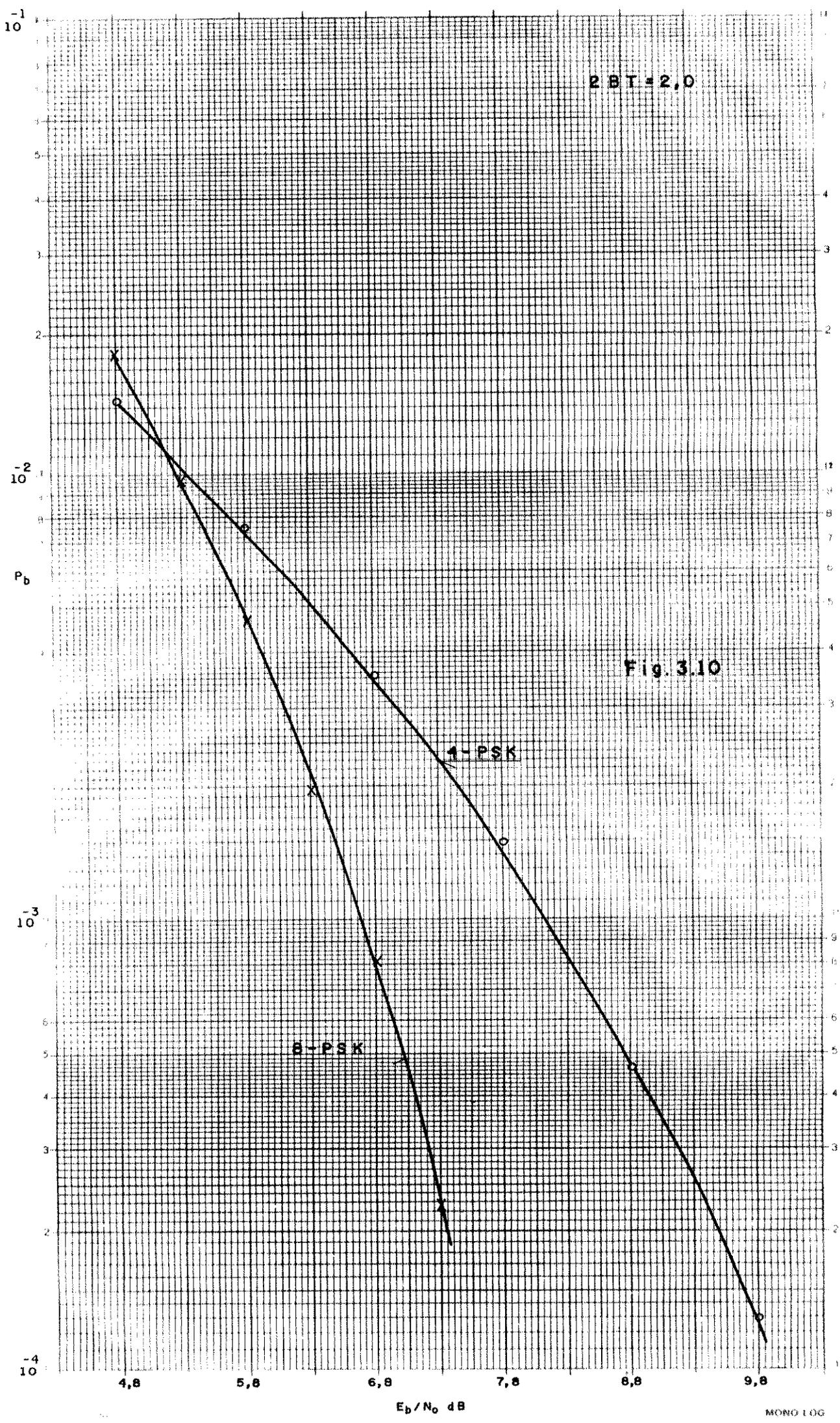
A figura 3.12 mostra os resultados para um produto  $2BT = 1,2$ . Apesar de que agora, o efeito de interferência intersimbólica é acentuado, o ganho de codificação para um valor de  $P_b = 10^{-4}$  se mantém inalterado:  $\cong 2$  dB.

A figura 3.12 também mostra os resultados obtidos para recepção quantizada. Nós simulamos dois esquemas:

a) um esquema com  $Q_a = Q_f = 4$ . A otimização do espaçamento  $\Delta r$  é mostrada na figura 3.13a. O valor de  $\Delta r$  ótimo considerado foi de 0,3.

b) um esquema com  $Q_a = 4$  e  $Q_f = 8$ . A otimização do espaçamento é mostrada na figura 3.13b. O valor de  $\Delta r$  ótimo considerado foi de 0,2.

As métricas utilizadas na recepção quantizada, foram as métricas quantizadas definidas na seção 2.5.2. Nós utilizamos



2 BT = 2,0

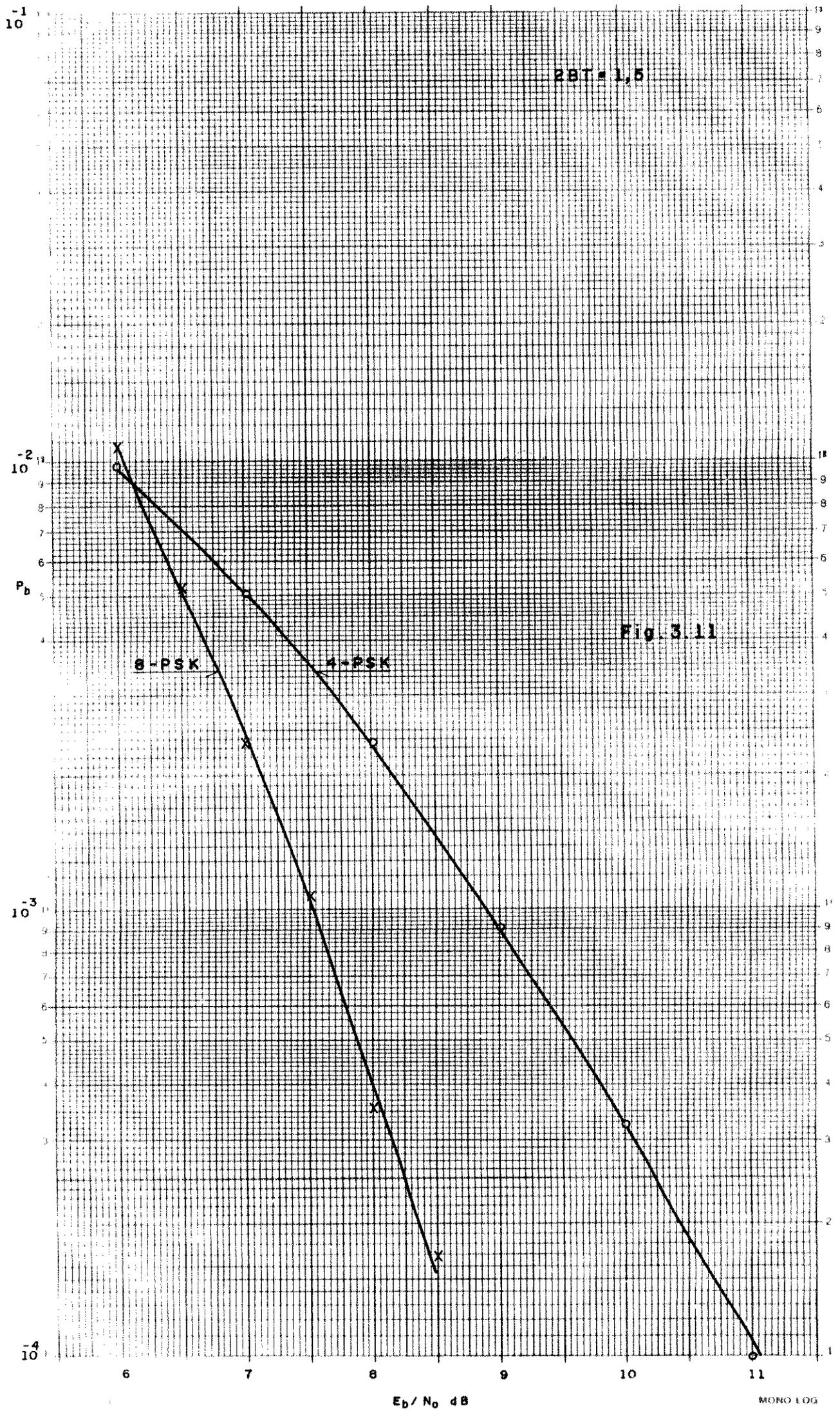
Fig. 3.10

1-PSK

8-PSK

$E_b/N_0$  dB

MONO LOG



$10^{-1}$

2 BT = 1,2

$10^{-2}$

$P_b$

$10^{-3}$

$10^{-4}$

7

8

9

10

11

12

$E_b/N_0$  dB

MONO-LING

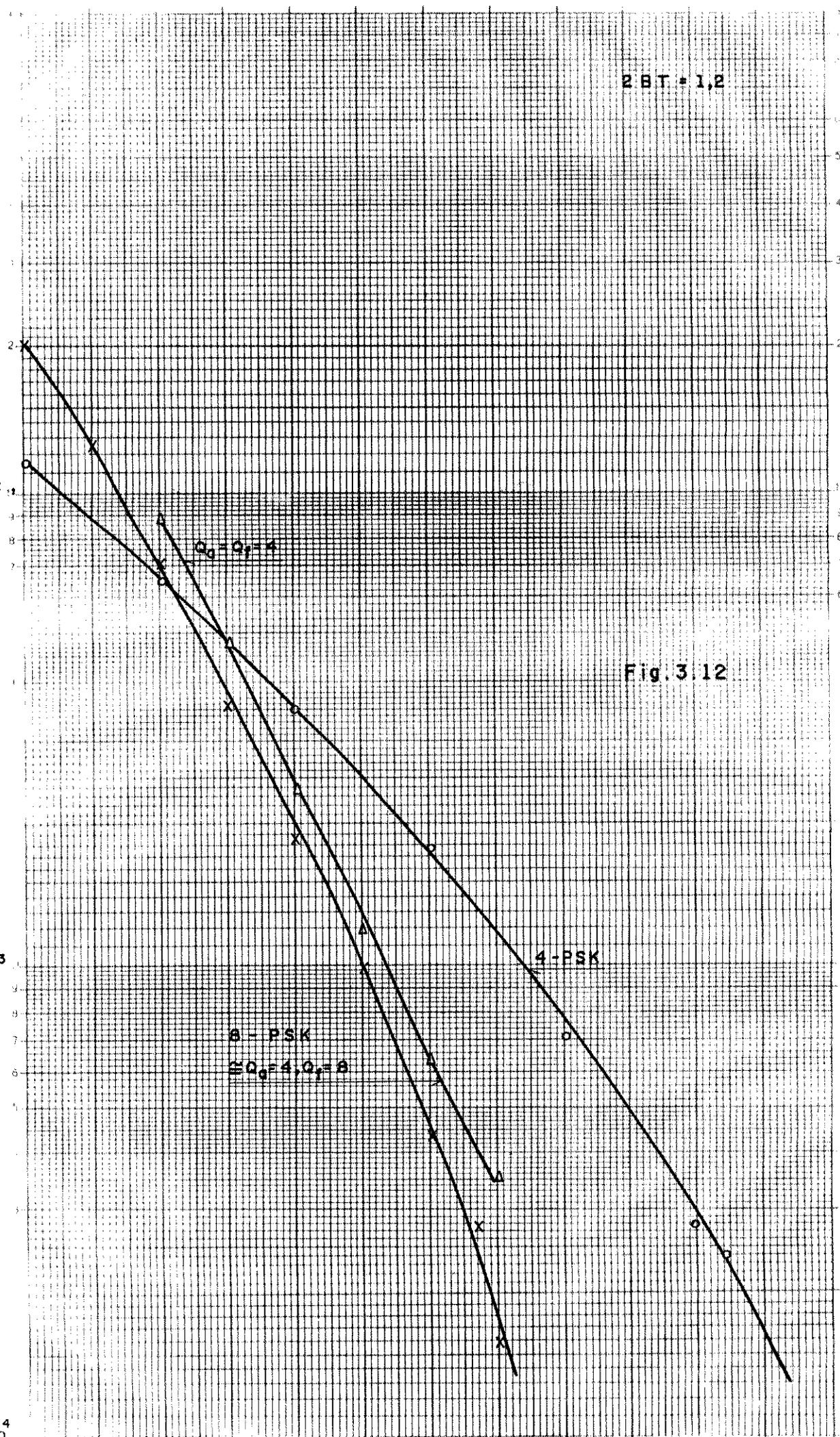
$Q_d = Q_f = 4$

8-PSK

$Q_d = 4, Q_f = 8$

4-PSK

Fig. 3.12



8 - PSK  
 $Q_a = Q_f = 4$   
 $E_b/N_0 = 8,5 \text{ dB}$   
MÉTRICA QUANTIZADA  
 $2BT = 1,2$

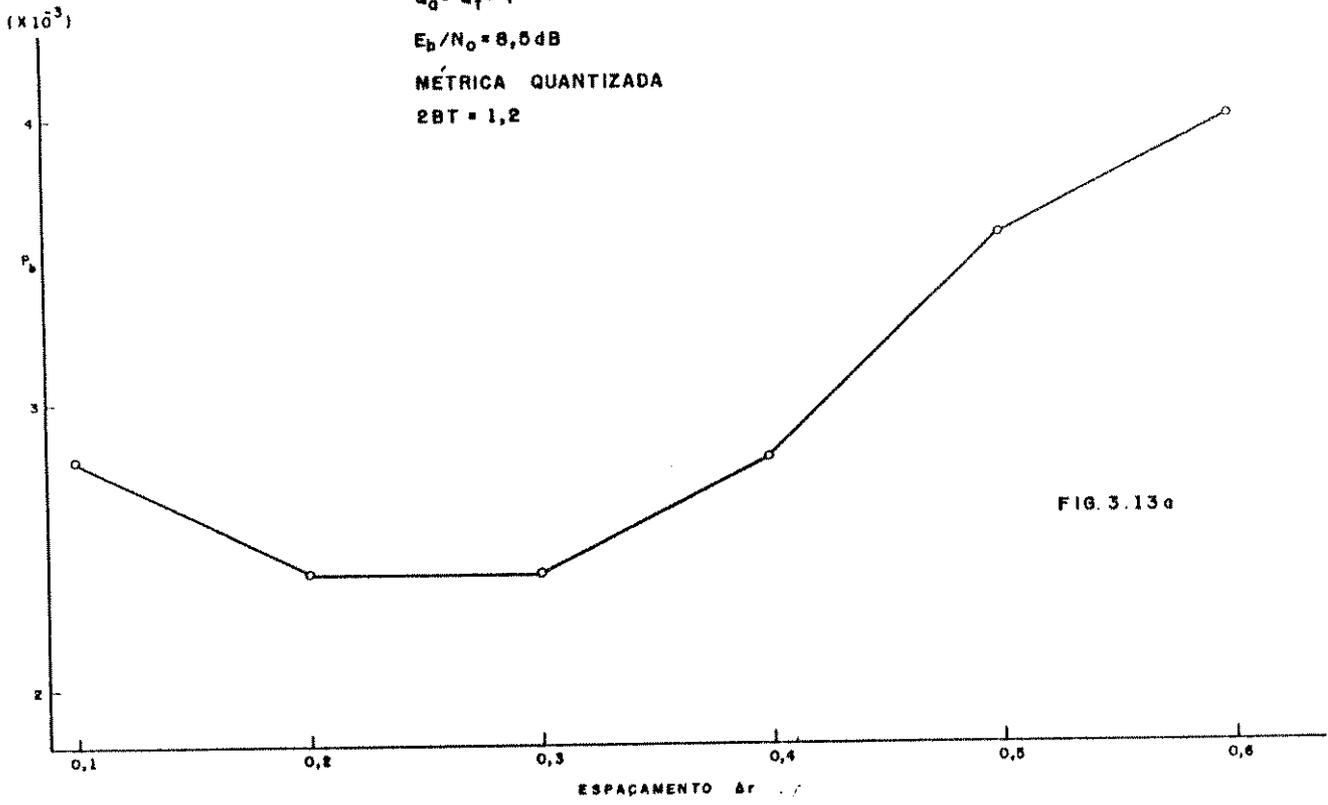


FIG. 3.13a

8 - PSK  
 $Q_d = 4, Q_f = 8$   
 $E_b/N_0 = 8,6 \text{ dB}$   
MÉTRICA QUANTIZADA  
 $2BT = 1,2$

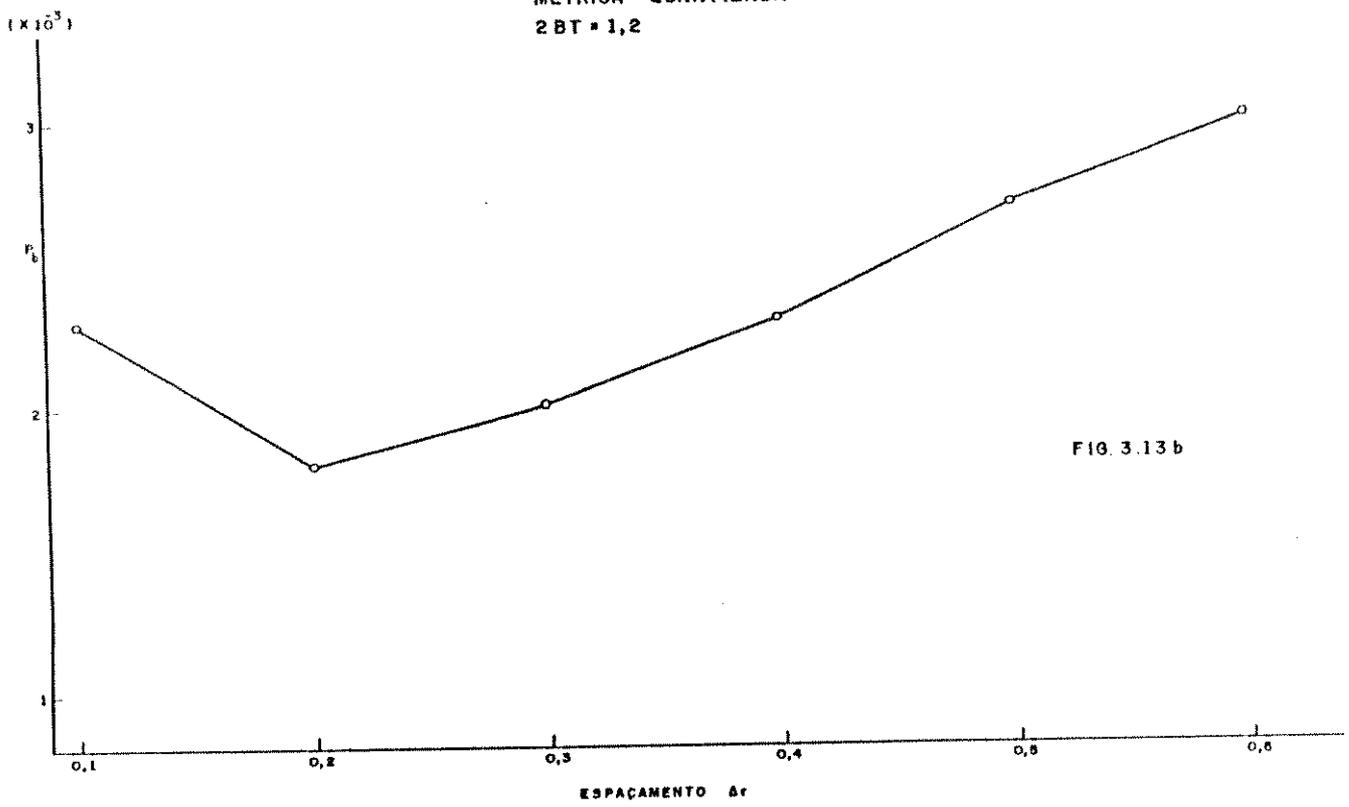


FIG. 3.13 b

as métricas com valores reais. A degradação de desempenho (em relação à recepção não quantizada) do esquema com  $Q_a = Q_f = 4$ , é sempre menor que 0,3 dB. O esquema com  $Q_a = 4$  e  $Q_f = 8$ , mostrou uma degradação desprezível em relação à recepção não quantizada.

### 3.13 Uma nota importante.

Considere a figura 3.14. A variável  $t$  é discreta. Os símbolos de informação  $e_t$  pertencem a um alfabeto  $E$ . Os símbolos na saída do codificador pertencem a um alfabeto  $S$ , e, na saída do canal temos um alfabeto  $X$ . O ruído ( $n_t$ ) é o AWGN.

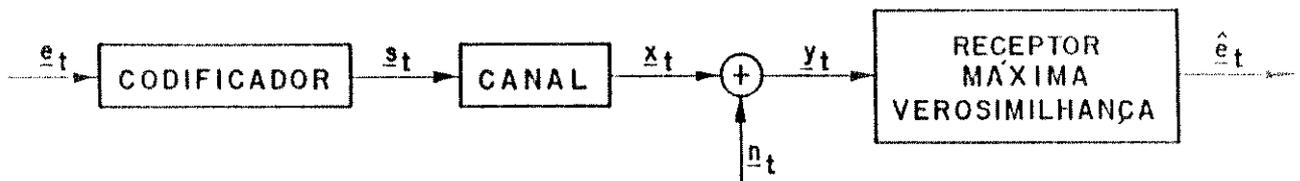


fig. 3.14 Modelo discreto de um sistema de comunicação digital.

A diferença desta figura para a figura 1.2 do capítulo um, é que agora há a presença do canal que transforma o vetor  $s_t$  no vetor  $x_t$ . O receptor de máxima verossimilhança faz o mesmo papel do decodificador da figura 1.2, e ambos podem ser implementados pelo algoritmo de Viterbi (note que o "canal" que existe entre  $x_t$  e  $y_t$  é sem memória).

A simulação do canal de faixa limitada e não linear, é na verdade, a obtenção de amostras dos sinais do alfabeto  $S$ ; o processamento das mesmas para a obtenção do alfabeto  $X$  na saída do filtro casado; e a posterior estimação da probabilidade de erro de bit.

O receptor de máxima verossimilhança deve ser projetado obedecendo a um compromisso entre complexidade e desempenho. Este compromisso está limitado por duas situações extremas [43]:

- 1) O receptor tem completo conhecimento do canal e do codificador de maneira que ele pode estimar os símbolos de informação  $e_t$  baseando-se na saída do canal  $x_t$ .
- 2) O receptor conhece apenas o alfabeto  $S$  do codificador. Não sabe nada sobre o canal. A estimação de  $e_t$  é feita baseada nos símbolos  $s_t$ .

A simulação desenvolvida neste capítulo se insere na situação 2 descrita acima.

Vale ressaltar um ponto importante aqui. A procura de códigos ótimos para o AWGN se insere na situação 2. Nesta situação apenas a memória do codificador é considerada. A procura de códigos ótimos para a situação 1 leva em consideração a memória do codificador além da memória do canal. Como vimos na seção 3.5, filtros introduzem memória no canal. A não linearidade pode ou não introduzir memória.

### 3.14 Conclusão.

Nas simulações nós utilizamos um código com memória  $m=2$  para a constelação 8-PSK: um código de pequena complexidade. Os resultados obtidos confirmaram uma característica importante dos códigos AR: sob o efeito de interferência intersimbólica e não linearidade, os ganhos relativos de codificação são mantidos. Ou seja: o sistema codificado e o não codificado apresentam quase que a mesma degradação de desempenho em relação ao canal AWGN.

As simulações também demonstraram a viabilidade de utilização, num canal de satélite, do esquema de quantização polar para a constelação 8-PSK simulada (o esquema com  $Q_a = 4$  e  $Q_f = 8$ , teve uma degradação desprezível em relação à quantização infinita). Apesar de termos feito a simulação para um determinado código, acreditamos que o resultado obtido seja representativo para a classe de códigos M-PSK focalizada neste trabalho.

## CONSIDERAÇÕES

### FINAIS

---

Como mencionamos na introdução do capítulo um, para um sistema de comunicação digital, estamos interessados em obter maiores taxas de transmissão de informação e melhor confiabilidade. As restrições são: consumo de potência, largura de faixa e complexidade de implementação.

A figura C.1 mostra a curva de probabilidade de erro de bit ( $P_b$ ), num canal AWGN, para um código 32-PSK (4 bits de informação / T) com memória  $m = 2$ , pertencente à classe de códigos descrita no capítulo um. Para um valor de  $P_b = 10^{-4}$ , o ganho de codificação é de 1,9 dB. O ganho assintótico é  $\cong 3,5$ dB. É interessante notar que os códigos pertencentes à mesma classe, para 8-PSK (ganho assintótico  $\cong 3,0$  dB) e para 16-PSK (ganho assintótico  $\cong 3,5$  dB), também apresentaram ganhos de codificação próximos de 2,0 dB, num valor de  $P_b = 10^{-4}$ .

Considere que estamos trabalhando sob restrição de faixa e consumo de potência. Se temos como objetivo, uma alta taxa de transmissão de informação, e, ao mesmo tempo, uma pequena taxa de erros ( $< 10^{-5}$ ), o código da figura C.1 poderá ser um candidato apropriado. É preciso levar em conta, ainda, a complexidade de implementação.

Para as constelações M-PSK consideradas neste trabalho, um problema de implementação importante é o da recuperação da fase da portadora. Em [49], é descrito sumariamente o desenvolvimento de um Modem que utiliza o código 8-PSK analisado

32-PSK  
CD = 24  
Q = 8

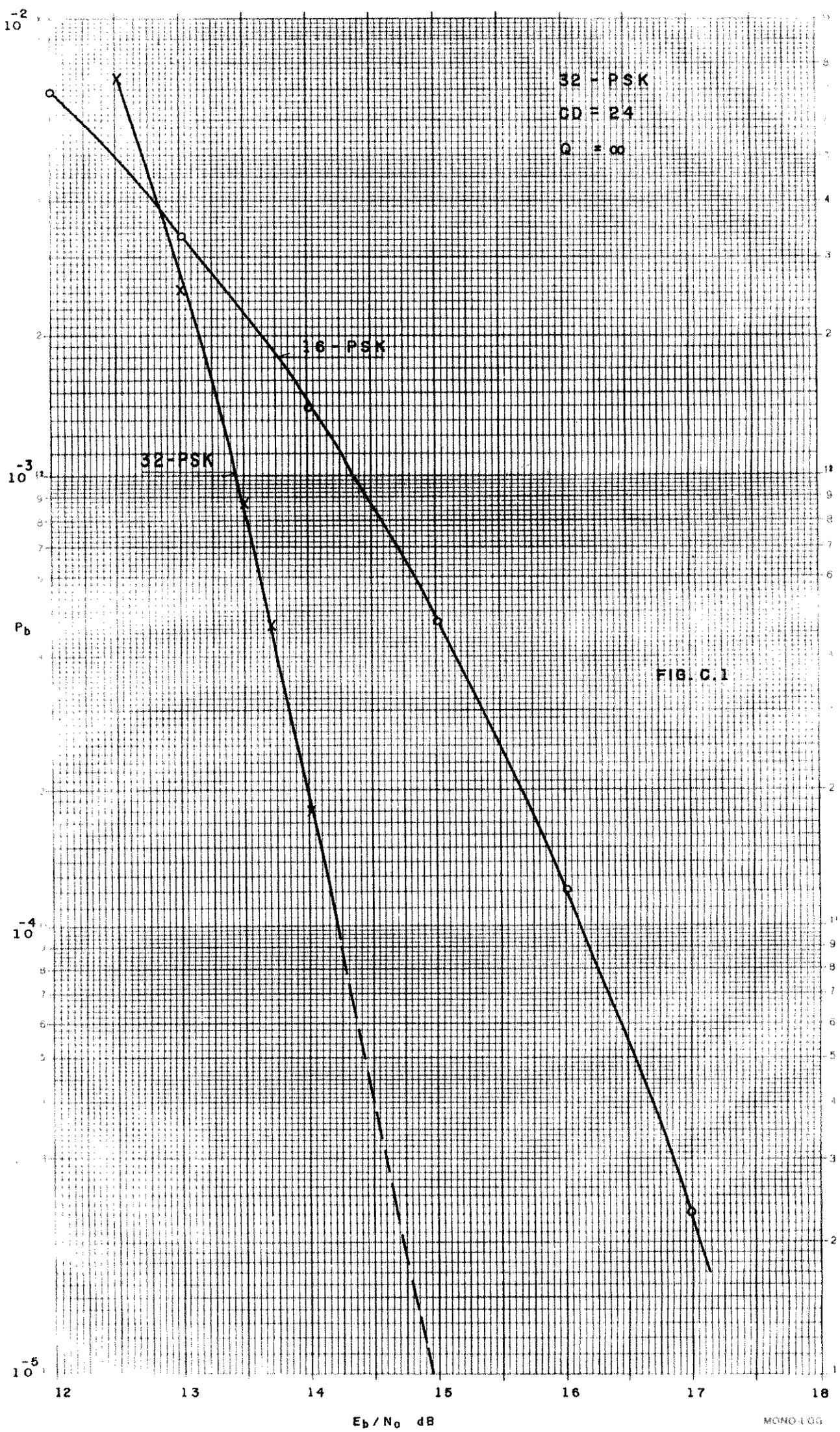


FIG. C.1

neste trabalho. Todo processamento do receptor, inclusive um algoritmo de recuperação de fase, foi implementado em software por um processador de sinais dedicado. O desempenho experimental obtido, concorda com o resultado da simulação por nós realizada (seção 2.7), onde consideramos uma recuperação perfeita da fase da portadora. Fica aqui a sugestão de um futuro trabalho: a análise dos problemas de recuperação de fase para as constelações com  $M = 16$  e  $32$  sinais.

## REFERENCIAS

- [1] Shanmugam, K.S. "Digital and Analog Communication Systems", Wiley, 1979.
- [2] Ungerboeck, G. "Channel coding with multilevel/phase Signals" IEEE Trans. on Information Theory, vol. IT-28, janeiro 1982.
- [3] Massey, J.L. "Coding and Modulation in Digital Communications" Proc. 1974 Int. Zurich Seminar on Digital Comm., Zurich, Switzerland, pp. E2(1)-E2(4), março 1974.
- [4] Arthurs, E. e Dym, H. "On the Optimum Detection of Digital Signals in The Presence of White Gaussian Noise - A Geometric Interpretation and a Study of Three Basic Data Transmission Systems" - IRE Trans. on Communications Systems, dezembro de 1962.
- [5] Whalen, A.D., "Detection of Signals in Noise", Academic Press, 1971.
- [6] Proakis, J.G., "Digital Communications", McGraw Hill, 1983.
- [7] Rhodes, S.A., Fang, R.J., e Chang, P.Y. "Coded octal phase shift keying in TDMA satellite communications", COMSAT Technical Review, vol. 13, No 2, Fall 1983.
- [8] Wilson, S.G. et al "Rate 3/4 convolutional coding of 16-PSK: code design and performance study" IEEE Trans. on Communications, vol. Com-32, N.12, dezembro 1984.
- [9] Biglieri, E. "High-Level Modulation and Coding for Nonlinear Satellite Channels" IEEE Trans. on Communications, vol. Com-32, N.5, maio 1984.
- [10] Wozencraft, J.M. e Jacobs, I.M. "Principles of Communication Engineering." New York, Wiley, 1965.
- [11] Marsan, M.A. et al, "Theoretical aspects and performance evaluation of channel encoding with redundant modulator alphabets" IEEE Global Telecommunications Conference, pp. 20.5.1 a 20.5.5, dezembro 1984, U.S.A.
- [12] Viterbi, A.J. "Convolutional Codes and Their Performance in Communication Systems" IEEE Trans. on Communications Technology, vol. Com-19, N.5, outubro 1971.
- [13] Forney Jr., G.D. "The Viterbi Algorithm" Proceedings of the IEEE, vol. 61, N.3, março 1973.
- [14] Viterbi, A.J. e Omura, J.K. "Principles of Digital Communication and Coding" McGraw Hill-Kogakusha, LTD. 1979.

- [15] Omura, J.K. "On the Viterbi decoding algorithm" IEEE Trans. Inform. Theory (Corresp.), vol. IT-15, janeiro 1969, pp.177-179.
- [16] Thapar, H.K. "Real time application of trellis coding to high speed voice band data transmission" IEEE Journal on Selected Areas in Communications, vol. SAC-2, N.5, setembro 1984.
- [17] Borelli, W.C. "Convolutional codes for multilevel data transmission" Ph.d. Thesis, University of Kent, England, junho 1983.
- [18] Tait, D.J. "Report to Plessey Telecommunications on EDC study for 16 Kbits/s Modem" University of Kent, England, julho 1979.
- [19] Lee, L.N. "On optimal soft-decision demodulation" IEEE Trans. on Information Theory, vol. IT-22, No 4, julho 1976.
- [20] Viterbi, A.J. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm" IEEE Trans. Inform. Theory, vol IT-13, abril 1967, pp.260-269.
- [21] Wozencraft, J.M. e Kennedy, R.S. "Modulation and Demodulation for Probabilistic Coding" IEEE Trans. on Information Theory, vol. IT-12, No 3, julho 1966.
- [22] Heller, J.A. e Jacobs, I.M. "Viterbi decoding for satellite and space communication" IEEE Trans. on Communication Technology, vol. Com-19, No 5, outubro 1971.
- [23] Clark Jr., G.C. e Cain, J.B. "Error-Correction Coding for Digital Communications", Plenum Press, New York, USA, 1981.
- [24] Conan, J. e Oliver, R. "Hardware and Software Implementation of the Viterbi Decoding Algorithm for Convolutional Codes", Proceedings of the 1976 International Symposium on Mini & Micro Computers, Toronto, Canada, Novembro 1976.
- [25] Bowker, A.H. e Lieberman, G.J. "Engineering Statistics", Prentice-Hall, Englewood Cliffs, New Jersey, 1972.
- [26] Lindsey, W.C. e Simon, M.K. "Telecommunication Systems Engineering", Prentice-Hall, INC., Englewood Cliffs, New Jersey, 1972.
- [27] Van Trees, H.L. (Editor) "Satellite Communications", IEEE Press, New York, USA, 1979.
- [28] Pritchard, W.L. "Satellite Communication - An Overview of the Problems and Programs", Proc. IEEE, vol.65, pp-294-307, Março 1977, obtido da referência [27].
- [29] Feher, K. "Digital Communications: Satellite/Earth Station Engineering", Prentice Hall, Englewood Cliffs, N.J., USA, 1983.

- [30] Chan, H.C., Taylor, D.P. e Haykin, S.S. "A comparative study of digital modulation techniques for single-carrier satellite communications", Proc. 3rd. Int. Conf. Digital Satellite Commun., Kyoto, Japão, Novembro, 1975, pp. 56-64.
- [31] Taylor, D.P. e Chan, H.C., "A simulation study of two bandwidth efficient modulation techniques" IEEE Trans. Commun., vol. COM-29, março, 1981.
- [32] Stiffler, J.J. "Theory of Synchronous Communications", Prentice Hall, Englewood Cliffs, N.J., USA, 1971.
- [33] Papoulis, A. "Signal Analysis", McGraw-Hill, Inc., USA, 1977.
- [34] Johnson, D.E. e Hilburn, J.L. "Rapid practical designs of active filters", John Wiley, 1975.
- [35] Van Valkenburg, M.E. "Introduction to Modern Network Synthesis", John Wiley & Sons, Inc., 1960, USA.
- [36] Golomb, S.W. et al "Digital Communications with Space Applications", Prentice-Hall, Inc., Englewood Cliffs, N.J., USA, capítulo 2.
- [37] Saleh, A.A.M. "Frequency independent and frequency dependent nonlinear models of TWT amplifiers" IEEE Trans. Commun., vol. COM-29, Novembro, 1981.
- [38] Van Trees, H.L. "Detection, Estimation and Modulation Theory", Part III, Appendix, John Wiley & Sons, Inc., 1971.
- [39] Weinberg, A. "The impact of pulsed RFI on the coded BER performance of the nonlinear satellite communication channel" IEEE Trans. Commun., vol. COM-29, No 5, Maio, 1981.
- [40] McKenzie, T.M. e Braun, W.R. "Carrier phase recovery performance for PN spread TDRSS links with radio frequency interference", NTC '79 Conf. Rec., Novembro, 1979 (Referência 7 da referência [39]).
- [41] Hamid, A., Baker, S.R. e Cook, W. "A computer program for communications channel modeling and simulation", COMSAT Technical Review, vol. 13, No 2, Fall 1983.
- [42] Palmer, L.C. e Lebowitz, S. "Including synchronization in time domain channel simulations", COMSAT Technical Review, vol. 7, No 2, Fall 1977.
- [43] Marsan, M.A. et al "On the Performance Analysis of Combined Coding and Modulation Schemes over Real Channels", IEEE Global Telecommunications Conference, pp. 21.1.1 a 21.1.5, 1985.
- [44] Jones, J.J. "Filter Distortion and Intersymbol Interference Effects on PSK Signals", IEEE Trans. Comm. Technology, vol. COM-19, No 2, Abril 1971.

[45] Omura, J.K. e Simon, M.K. "Generalized Transfer Function Bounds", Appendix A, Jet Propulsion Laboratory, Technical Report, julho 1981. Veja também: Omura, J.K. "Performance Bounds for Viterbi Algorithms", IEEE Int. Conf. on Comm., pp.2.2.1 a 2.2.5, Denver, Colorado, Junho 1981.

[46] Markman, Ivonete - Comunicação Privada.

[47] Dorf, R.C. "Modern Control Systems", Addison - Wesley Publishing Company, 1974.

[48] Wilson, S.G. "Bandwidth-Efficient Modulation and Coding: A Survey of Recent Results" Int.Conf.Communications, julho, 1986, pp. 31.1.1 a 31.1.5., USA.

[49] "Coded Phase Modem at 64 kbit/s", Contract INTEL-242, German Aerospace Research Establishment (DFVLR) / IBM Zurich Research Laboratory, Completion Report, Outubro, 1983.