



Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

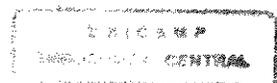
Códigos de Bloco de Modulação M-DPSK com uso de CSI para um Canal de Desvanecimento Rayleigh

Por: Eng Cristhof Johann Roosen Runge

Orientador: Prof Dr. Jaime Portugheis

Dissertação submetida à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, para preenchimento dos pré-requisitos parciais para obtenção do Título de Mestre em Engenharia Elétrica.

Este exemplar corresponde a redação final da tese defendida por Cristhof J. R. Runge e aprovada pela Comissão Julgada em 17 / 10 / 97.
Jaime Portugheis
Orientador



no 23455

UNIDADE	BC
N.º CHAMADA:	UNICAMP
	R873c
V	Es
TERMO:	80/36311
PROD.	229/99
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	27/04/99
N.º CPD	

CM-00120201-2

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

R873c Runge, Cristhof Johann Roosen
Códigos de bloco de modulação M-DPSK com uso de
CSI para um canal de desvanecimento Rayleigh. /
Cristhof Johann Roosen Runge.--Campinas, SP: [s.n.],
1997.

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

1. Códigos de controle de erros (Teoria da
informação). 2. Rádio - Transmissores e Transmissão -
Desvanecimento. 3. Modulação digital. 4. Sistemas de
comunicação móvel. I. Portugheis, Jaime. II.
Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título.

A meus pais e irmãos, que embora muitas vezes o destino nos tenha feito distantes, nunca saíram de meu coração.

“A ausência diminui
as pequenas paixões
e aumenta as grandes,
da mesma forma como
o vento apaga as velas
e atíça as fogueiras.”

Francois de La Rochefoucault

MEUS SINCEROS AGRADECIMENTOS:

- Ao Jaime Portugheis pela amizade, compreensão, dedicação e preciosa orientação durante a elaboração desta tese.
- Aos professores da banca examinadora pela atenção e pelas valiosas sugestões para a melhoria deste trabalho.
- Ao amigo Eder, que nos momentos difíceis me aconselhou de maneira sábia e cuja ajuda foi de grande importância para a realização deste trabalho.
- Aos amigos que estiveram presentes durante esta jornada que se encerra, e a todos aqueles que encontrei ao longo de minha vida e que deixaram a indelével marca de sua existência.

**Este trabalho contou com o apoio financeiro
da entidade CAPES e do grupo PET.**

Sumário

Resumo	VII
---------------	------------

Abstract	VIII
-----------------	-------------

1- INTRODUÇÃO	1
----------------------	----------

1.1 - Estrutura do Trabalho	3
-----------------------------	---

2 - CONCEITOS BÁSICOS	4
------------------------------	----------

2.1 - Caracterização do Canal com Desvanecimento	4
--	---

2.1.1 - Representação Complexa em Banda Básica	5
--	---

2.1.2 - Sinal Recebido e Resposta do Canal	6
--	---

2.1.3 - Densidades Espectrais e Funções de Correlação	8
---	---

2.1.4 - Efeito Doppler	10
------------------------	----

2.2 - Caracterização Matemática de Sistemas	13
---	----

2.2.1 - Caracterização do Sinal Enviado	14
---	----

2.3 - O Receptor de Correlação	16
--------------------------------	----

3 - CARACTERIZAÇÃO DO SISTEMA	19
--------------------------------------	-----------

3.1 - Modelagem Matemática do Sistema	22
---------------------------------------	----

3.1.1 - Modelagem do Canal	22
----------------------------	----

3.2 - Métrica de Máxima Verossimilhança	24
---	----

3.3 - Codificação Diferencial	27
-------------------------------	----

3.4 - Métrica de Decisão de Sequência	29
---------------------------------------	----

3.5 - Embaralhamento	30
----------------------	----

4 - LIMITANTE DE ERRO	32
4.1 - Determinação da Matriz de Covariância e do Vetor Esperança \underline{W}	35
4.2 - Obtenção dos Resultados Numéricos	37
4.2.1 - Minimização de $\Delta\Phi(\lambda)$	40
4.2.2 - Aproximação do Fator de Chernoff para uma sequência	43
5 - CONSTRUÇÃO DE CÓDIGOS	46
5.1 - Análise do Limitante de Comportamento (LC	46
5.2 - Estudo do Comportamento de	49
5.3 - Algoritmo para Construção de Códigos	53
5.4 - Códigos Gerados	55
6 - RESULTADOS DE SIMULAÇÃO	57
6.1 - Métrica de Máxima Verossimillança	58
6.2 - Influência de $D\delta$	60
6.3 - Influência de D_H	61
6.4 - Influência de N	62
6.5 - Influência do uso de CSI	63
6.6 - Melhores Códigos 1	64
6.7 - Melhores Códigos 2	65
7 - CONCLUSÃO	67
ANEXO A	69
ANEXO B	82
BIBLIOGRAFIA	93

Resumo

Este trabalho estuda a construção de códigos de bloco de modulação M-DPSK, com detecção multi-simbólica e presença de informação de estado de canal (CSI), para canais com desvanecimento Rayleigh. Através de uma nova interpretação do fator de Chernoff, critérios para construção de códigos são sugeridos. Estes critérios coincidem com aqueles obtidos para o modelo de canal sem CSI. Um algoritmo de procura de códigos curtos é implementado. Resultados de simulação para os códigos encontrados comprovam a importância da otimização dos parâmetros de projeto sugeridos.

Abstract

This work studies the construction of block codes for M-DPSK modulation with multi-symbol detection and the use of channel state information (CSI) in a Rayleigh fading channel. Based on a new interpretation of the Chernoff bound, criteria for codes construction are suggested. This criteria coincides with those obtained for the channel model without the use of CSI. A search algorithm for short block codes generation is implemented. Simulation results for the obtained codes confirm the importance of the suggested design parameters.

Capítulo 1

Introdução

O rápido desenvolvimento das telecomunicações neste século gerou condições para o aparecimento de um mercado que se apresenta em contínua expansão. A necessidade de atender a este mercado com sistemas cada vez mais eficientes e capazes de suprir esta demanda, tem motivado centros de pesquisa e empresas do setor em todo o mundo na busca de soluções para os sistemas atuais e no desenvolvimento de novas tecnologias.

Particularmente nesta última década as comunicações móveis pessoais assumiram um papel de destaque neste cenário, e um grande esforço tem sido realizado para a obtenção de sistemas cada vez mais confiáveis, robustos e com maior capacidade, sendo que atenção especial tem sido dada aos sistemas digitais.

Neste trabalho será estudado um dos estágios que constituem o processo de comunicação em sistemas digitais, a codificação de canal. Para melhor definirmos o objetivo do nosso estudo, vamos inicialmente considerar o diagrama de blocos de um sistema de comunicação digital na Figura 1.1.

A codificação de fonte, ou seja a forma com que a informação pode ser representada digitalmente de maneira eficiente, e a codificação de canal, que aborda a questão da transmissão da informação através de um canal ruidoso, são assuntos tratados pela teoria de informação, que define seus limites fundamentais.

Este trabalho está concentrado na codificação de canal. Nosso estudo é dedicado à busca e construção de códigos de forma a melhorar a eficiência dos atuais sistemas. A fim de obtermos esquemas de codificação eficientes, um estudo matemático do sistema em questão é realizado. Nosso objetivo aqui é a construção de códigos para canais de comunicação móvel. Desta forma as técnicas utilizadas, o tipo de modulação e a classe de códigos a serem empregados, são escolhidos levando-se em consideração as características e restrições inerentes a esta forma de comunicação.

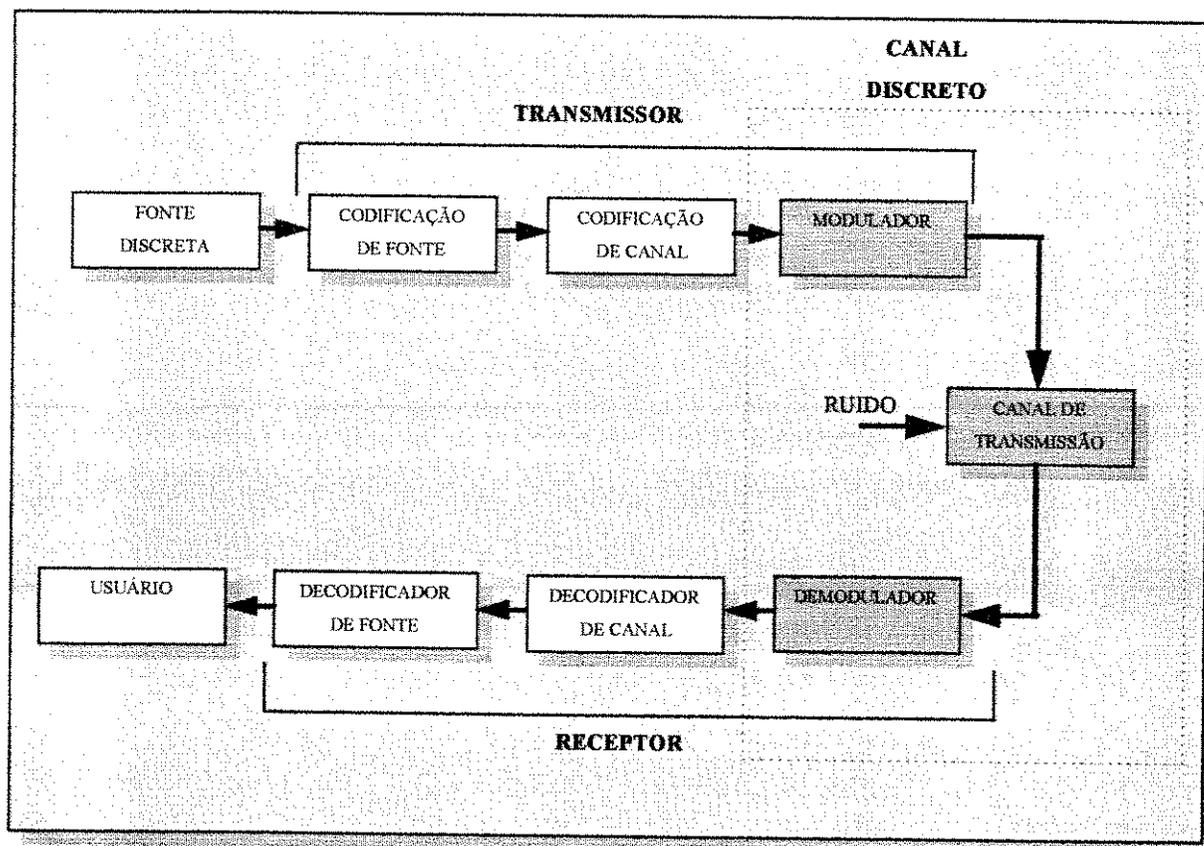


Figura 1.1 - Diagrama de Blocos de um sistema de comunicação digital

O estudo é iniciado com a caracterização do canal a ser utilizado, aqui considerado de desvanecimento do tipo Rayleigh. A seguir a descrição do sistema, bem como considerações sobre as técnicas a serem empregadas são apresentadas. A partir de um estudo estatístico do canal, abordado através do fator de Chernoff, os critérios para a construção de códigos são determinados. Um algoritmo de busca exaustiva baseado nos critérios extraídos do fator de Chernoff é então implementado para a geração de códigos de bloco curtos.

A análise de desempenho dos esquemas codificados, utilizando os códigos gerados, é feita através de simulações computacionais, mostrando a eficiência destes códigos e a concordância dos resultados com a teoria.

1.1 - Estrutura do Trabalho

No Capítulo 2 é feita uma análise matemática sobre o canal de desvanecimento, em que são discutidas as grandezas envolvidas na sua caracterização. A seguir são apresentados conceitos básicos sobre a modelagem matemática de sistemas de comunicação.

No Capítulo 3 uma descrição mais detalhada do sistema a ser utilizado é apresentada. Neste capítulo também são discutidas as técnicas que serão empregadas. Um estudo sobre a métrica de máxima verossimilhança, baseada na probabilidade de transição do canal, e aproximações da métrica ótima para utilização prática também são apresentadas.

No Capítulo 4 é desenvolvido o cálculo do fator de Chernoff condicionado para o sistema em questão. Devido ao cálculo não apresentar uma forma fechada, uma nova interpretação dos resultados é apresentada possibilitando a determinação de critérios para a construção de códigos.

No Capítulo 5 um algoritmo de busca exaustiva baseado nos critérios extraídos do fator de Chernoff é apresentado. A partir dos resultados obtidos através deste algoritmo são apresentadas tabelas contendo os códigos gerados com seus respectivos parâmetros.

No Capítulo 6 são apresentados os resultados de simulação para os esquemas codificados utilizando alguns dos códigos gerados. Através dos resultados de simulação a influência dos critérios de construção e das técnicas utilizadas são analisados.

No Capítulo 7 são apresentadas as conclusões do trabalho, discutindo-se vantagens e desvantagens na utilização dos esquemas codificados e das técnicas empregadas. Algumas sugestões para trabalhos futuros também são comentadas.

Capítulo 2

Conceitos Básicos

Neste capítulo vamos apresentar alguns conceitos básicos da teoria de transmissão digital. Inicialmente vamos caracterizar o canal e o sistema de transmissão, onde através do conceito de banda básica, um modelamento matemático será apresentado. Um esquema de recepção baseado em filtros de correlação também é descrito ao final do capítulo.

O texto apresentado neste capítulo utilizou-se em grande parte de [1] como fonte de referência.

2.1 - Caracterização do Canal com Desvanecimento

Na transmissão através de canais com desvanecimento (do inglês fading) o sinal enviado pelo transmissor, devido à natureza do meio de transmissão, percorre diferentes caminhos até chegar ao receptor. Em cada caminho, o sinal é refletido por obstáculos de naturezas diferentes e que podem variar ao longo do tempo. Devido a este comportamento o canal também é chamado de “canal de múltiplo-percurso”.

Várias são as situações reais onde o canal pode ser caracterizado como sendo um canal de desvanecimento. Na comunicação radio HF, por exemplo, onde a ionosfera é usada como refletor para a comunicação entre dois pontos, a variação dinâmica das suas camadas causada por diversos fatores, é responsável pelo espalhamento do sinal e pela variação no comprimento dos diferentes caminhos percorridos. Também nos sistemas rádio móvel onde existe uma locomoção relativa entre a antena transmissora e a antena receptora, além dos vários obstáculos no percurso que causam reflexões e difrações no sinal enviado produzindo o efeito de múltiplo-percurso, o fenômeno de desvanecimento é observado.

Assim o sinal recebido consiste do somatório de réplicas do sinal enviado com diferentes amplitudes e atrasos e que variam ao longo do tempo. Devido à incerteza e a

impossibilidade de previsão exata do sinal recebido, é razoável caracterizar o canal de maneira estatística.

A fim de modelar matematicamente o canal e seus efeitos sobre um sinal enviado utilizaremos a representação complexa em banda básica, por uma questão de simplicidade.

2.1.1 - Representação Complexa em Banda Básica

Na representação complexa em banda básica, tanto o sinal (aqui considerado de faixa estreita centrado em torno de uma frequência f_c) como o canal, considerado passa faixa, são transformados em um sinal equivalente onde a portadora é suprimida. Desta forma o sistema resultante consiste de um equivalente passa baixa, onde os conteúdos de frequências estão centrados em torno do zero.

Seja $s(t)$ um sinal da forma,

$$s(t) = A \cos(2\pi f_c t + \theta(t)) \quad (2.1)$$

então $s(t)$ pode ser escrito como,

$$s(t) = x(t) \cos(2\pi f_c t) - y(t) \sin(2\pi f_c t) \quad (2.2)$$

Na expressão acima as componentes de sinal em baixa frequência $x(t)$ e $y(t)$, podem ser vistas como uma modulação em amplitude imposta às portadoras $\cos(2\pi f_c)$ e $\sin(2\pi f_c)$ respectivamente. Para o caso do uso de modulação PSK ($\theta(t) = cte = \theta$), $x(t)$ e $y(t)$ são constantes e iguais a $A \cos(\theta)$ e $A \sin(\theta)$ respectivamente.

Seja $s(t)$ o sinal enviado. Então,

$$s(t) = \operatorname{Re} \left\{ (x(t) + jy(t)) e^{j2\pi f_c t} \right\} \quad (2.3)$$

Denotando,

$$s_I(t) = x(t) + jy(t) \quad (2.4)$$

então $s_l(t)$ é a representação complexa em banda básica do sinal. De forma similar pode ser feita a representação do sistema.

2.1.2 - Sinal Recebido e Resposta do Canal

Como o sinal enviado é propagado através de diferentes caminhos, cada um impondo uma atenuação e um atraso que é variante com o tempo, podemos então representar o sinal recebido na forma,

$$v(t) = \sum_n \alpha_n(t) s(t - \tau_n(t)) \quad (2.5)$$

onde $\alpha_n(t)$ é a atenuação causada pelo n-ésimo caminho e $\tau_n(t)$ é o atraso correspondente. Representando $v(t)$ na forma da sua envoltória complexa temos, (de (1) em (2)),

$$v(t) = \text{Re} \left\{ \left[\sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} s_l(t - \tau_n(t)) \right] e^{j2\pi f_c t} \right\} \quad (2.6)$$

Através da expressão anterior vemos que a versão em banda base do sinal recebido pode ser escrita como,

$$r_l(t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} s_l(t - \tau_n(t)) \quad (2.7)$$

Como $r_l(t)$ é a resposta em banda-básica do canal à entrada $s_l(t)$, então o canal em banda base pode ser modelado pela sua resposta impulsiva, variante no tempo,

$$c(\tau; t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n(t)) \quad (2.8)$$

onde $c(\tau; t)$ representa a resposta do canal no instante t devido ao impulso aplicado no instante $t - \tau$.

A fim de se caracterizar os efeitos do canal sobre um sinal enviado, vamos considerar o envio de uma portadora sem modulação em uma frequência f_c ($s_i(t)=1$).

Então o sinal recebido em banda-básica pode ser escrito como,

$$\begin{aligned} r_l(t) &= \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \\ &= \sum_n \alpha_n(t) e^{-j\theta_n(t)} \quad ; \text{ onde } \theta_n(t) = 2\pi f_c \tau_n(t) \quad (2.9) \end{aligned}$$

Através desta expressão vemos que o sinal recebido pode ser visto como a soma de fasores tendo amplitude $\alpha_n(t)$ e fase $\theta_n(t)$. É importante notar que grandes variações dinâmicas no meio são necessárias para que ocorram variações significativas em $\alpha_n(t)$. Em contra partida variações de $1/f_c$ em $\tau_n(t)$ são suficientes para rotacionar a fase $\theta_n(t)$ de 2π radianos. Uma vez que f_c é um valor elevado, pequenas variações de $\tau_n(t)$, (o que corresponde a pequenas variações dinâmicas do meio de transmissão), são suficientes para causar grandes variações na fase dos fasores que compõem o sinal recebido.

O sinal recebido é então o resultado da soma de fasores, que hora podem se somar destrutivamente fazendo com que o sinal recebido $r_l(t)$ seja muito pequeno ou mesmo zero, e hora podem se somar construtivamente resultando em um sinal de grande magnitude.

Quando o número de caminhos é grande, o que ocorre geralmente na prática, o teorema central do limite pode ser aplicado e o sinal $r_l(t)$ pode ser modelado como um processo gaussiano complexo na variável t .

Quando a resposta ao impulso $c(\tau; t)$ é modelada como um processo gaussiano complexo na variável "t", de média nula. A envoltória $|c(\tau; t)|$, em qualquer instante "t", tem distribuição Rayleigh, e a fase correspondente é uniformemente distribuída no intervalo $(-\pi, \pi)$. Neste caso o canal é dito de desvanecimento do tipo Rayleigh.

2.1.3 - Densidades Espectrais e Funções de Correlação do Canal

Modelando a resposta impulsiva $c(\tau; t)$ como um processo aleatório complexo e estacionário no sentido amplo. Define-se a função de autocorrelação de $c(\tau; t)$ como:

$$\phi_c(\tau_1, \tau_2; \Delta t) = \frac{1}{2} E \left[c^*(\tau_1; t) c(\tau_2; t + \Delta t) \right] \quad (2.10)$$

Na maioria dos meios de transmissão a atenuação e a rotação de fase associados a atrasos de caminhos distintos são descorrelacionados. Assim temos,

$$\frac{1}{2} E \left[c^*(\tau_1; t) c(\tau_2; t + \Delta t) \right] = \phi_c(\tau_1; \Delta t) \delta(\tau_1 - \tau_2) \quad (2.11)$$

fazendo $\Delta t = 0$ a autocorrelação resultante $\phi_c(\tau; 0) = \phi_c(\tau)$ representa a potência na saída do canal em função do atraso τ , ($\phi_c(\tau)$ é denominado perfil de intensidade de múltiplo-percurso. Um exemplo típico do comportamento de $\phi_c(\tau)$ pode ser visto na figura abaixo.

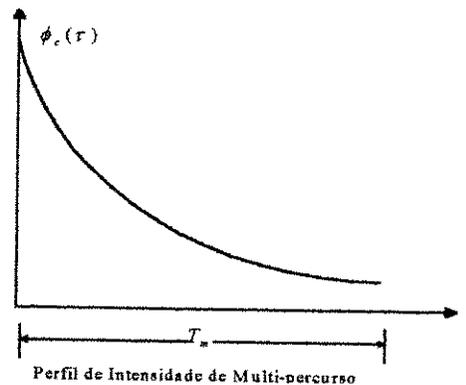


Figura 2.1 - Perfil de Intensidade de múltiplo-percurso

A faixa de valores de τ para os quais $\phi_c(\tau)$ é significativamente diferente de zero, é chamada espalhamento de multi-percurso do canal e denotado por T_m .

A partir de $\phi_c(\tau)$ podemos obter a sua transformada de Fourier $\phi_c(\Delta f)$,

$$\phi_c(\Delta f) = \int_{-\infty}^{\infty} \phi_c(\tau) e^{-j2\pi\Delta f\tau} d\tau \quad (2.12)$$

e uma vez que a transformada representa a função de autocorrelação no domínio da frequência, ela nos fornece a medida da banda de coerência do canal. Assim,

$$(\Delta f)_c \approx \frac{1}{T_m} \quad (2.13)$$

onde $(\Delta f)_c$ representa a banda de coerência do canal. Esta relação pode ser vista na figura abaixo.

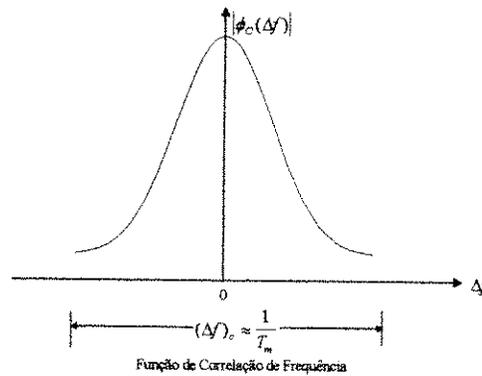


Figura 2.2 - Função de Correlação de Frequência

A banda de coerência fornece uma medida da seletividade do canal, ou seja, se a banda ocupada pelo sinal enviado exceder a banda de coerência do canal, o desvanecimento irá atuar de forma seletiva no sinal, afetando-o com intensidades diferentes dentro do espectro ocupado, resultando em uma distorção severa. Já no caso em que a banda ocupada pelo sinal é menor que $(\Delta f)_c$, então o canal atua de forma uniforme em todo o espectro do sinal enviado, neste caso o canal é dito não seletivo, (do inglês “Flat Fading”).

2.1.4 - Efeito Doppler

Vamos agora investigar o comportamento do canal em relação à variável Δt , (onde Δt representa a diferença temporal entre dois intervalos de observação da resposta do canal). Definindo,

$$\phi_c(\Delta f; \Delta t) = \int_{-\infty}^{\infty} \phi_c(\tau; \Delta t) e^{-j2\pi\Delta f\tau} d\tau \quad (2.14)$$

queremos investigar o efeito da variação temporal do canal no envio de um único tom centrado em uma frequência f . Fazendo-se então $\Delta f = 0$, ($\phi_c(\Delta f; \Delta t) = \phi_c(\Delta t)$) e definindo λ como a frequência Doppler relacionada a variável Δt temos através da transformada de Fourier de $\phi_c(\Delta t)$ que,

$$S_c(\lambda) = \int_{-\infty}^{\infty} \phi_c(\Delta t) e^{-j2\pi\lambda\Delta t} d\Delta t \quad (2.15)$$

A função $S_c(\lambda)$ consiste no espectro de potência que fornece a intensidade do sinal em função da frequência Doppler λ .

Para o caso de envio de um único tom, o canal espalha o espectro do sinal recebido em torno da frequência f . Notamos ainda que se o canal fosse invariante no tempo ($\phi_c(\Delta t) = 1$), $S_c(\lambda)$ se tornaria um impulso indicando que não haveria espalhamento na frequência do sinal enviado.

A faixa de valores para os quais $S_c(\lambda)$ é essencialmente não nulo é denominado espalhamento Doppler do canal e denotado por B_d .

Como $S_c(\lambda)$ é relacionado com $\phi_c(\Delta t)$ através da transformada de Fourier, o recíproco de B_d é uma medida do tempo de coerência do canal, ou seja,

$$(\Delta t)_c \approx \frac{1}{B_d} \quad (2.16)$$

As figuras abaixo mostram a relação entre $S_C(\lambda)$ e $\phi_C(\Delta t)$.

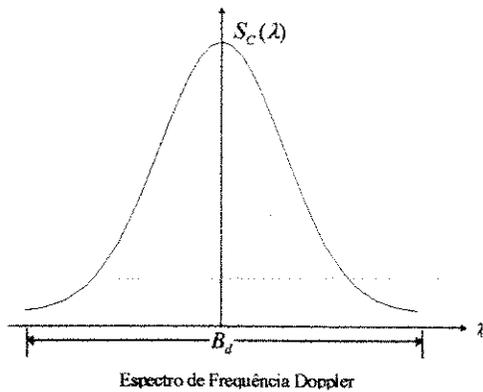


Figura 2.3 - Espectro de Frequência Doppler

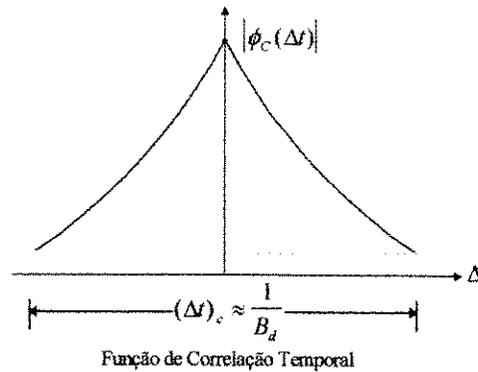


Figura 2.4 - Função de Correlação Temporal

Assim um canal com um grande tempo de coerência possui uma variação lenta o que equivale a dizer que o desvanecimento pode se manter constante durante o envio de vários símbolos de modulação. Neste caso o canal é dito de desvanecimento com variação lenta. Já no caso onde o tempo de coerência é pequeno, o desvanecimento pode variar mesmo durante um único intervalo de modulação e o canal é dito de desvanecimento com variação rápida.

Através da caracterização da banda de coerência e do tempo de coerência do canal, vemos que o comportamento do canal visto pelo sinal enviado, no que se refere a seletividade e a rapidez do desvanecimento, irá depender diretamente da banda ocupada pelo sinal, ou equivalentemente da duração do intervalo de símbolo de modulação.

Seja T o intervalo de modulação, satisfazendo a condição $T \gg T_m$ e seja $W \approx 1/T$ a banda do sinal enviado. Então,

$$W \ll \frac{1}{T_m} \approx (\Delta f)_c \tag{2.17}$$

Isto implica que sendo a banda do sinal muito menor que a banda de coerência do canal, o canal se comporta de maneira não seletiva. Desta forma as componentes em frequência do sinal são atenuadas e rotacionadas de fase da mesma quantidade em toda a extensão do espectro, em outras palavras a transformada de Fourier da resposta impulsiva do canal $C(f;t)$ é uma constante complexa na variável em frequência. Uma vez que $s_i(t)$ tem

seu conteúdo em frequência em torno do zero, então $C(f;t) = C(0;t)$ e o sinal recebido em banda básica é dado por,

$$r_i(t) = C(0;t)s_i(t) \tag{2.18}$$

A função de transferência $C(0;t)$ para um canal não seletivo pode ser modelada na forma fasorial como,

$$C(0;t) = \rho(t)e^{j\Theta(t)} \tag{2.19}$$

onde $\rho(t)$ representa a envoltória e $\Theta(t)$ representa a fase do canal equivalente em banda base e $C(0;t)$ está sendo modelado como um processo gaussiano complexo de média nula.

Suponhamos agora que além de satisfazer a condição $W \ll (\Delta f)_c$, ajustamos o intervalo de modulação de forma a também satisfazermos a condição $T \ll (\Delta t)_c$. Uma vez que T é menor que o tempo de coerência do canal, então estamos caracterizando um canal com desvanecimento de variação lenta, é razoável então assumir que o desvanecimento se mantém constante dentro de um intervalo de observação de alguns símbolos de modulação.

Pode ser mostrado facilmente que quando estamos sobre condição de não seletividade do canal e de variação lenta do desvanecimento, temos que $B_d T_m < 1$, este produto é denominado fator de espalhamento do canal e sob a condição $B_d T_m < 1$ o canal é dito sub-espalhado.

Supondo-se que o canal é sub-espalhado e considerando-se uma janela de observação menor que o tempo de coerência T_m , o ruído multiplicativo pode então ser modelado no receptor como a amostra de um processo gaussiano complexo que é constante durante o intervalo de observação considerado.

Assim podemos rescrever o sinal recebido em banda-básica como,

$$r_i(t) = \rho e^{j\Theta} s_i(t) + n(t) \quad ; 0 \leq t \leq LT \quad (LT \ll T_m) \tag{2.20}$$

onde LT é um múltiplo do intervalo de modulação.

Pode-se mostrar facilmente que “ ρ ” possui distribuição de Rayleigh e “ Θ ” é uniformemente distribuída no intervalo $(-\pi, \pi)$, e $n(t)$ representa o processo gaussiano branco complexo em banda base, que é aditivo ao sinal recebido e pode ser atribuído ao ruído térmico provocado pelos circuitos de recepção.

2.2 - Caracterização Matemática de Sistemas

A caracterização de um dado sistema pode ser conseguida através da sua modelagem matemática, onde o modelo utilizado procura traduzir o seu comportamento ao mesmo tempo que nos permite a obtenção de resultados que são relevantes para a sua análise.

O modelo do canal a ser utilizado pode ser representado esquematicamente através da figura abaixo.

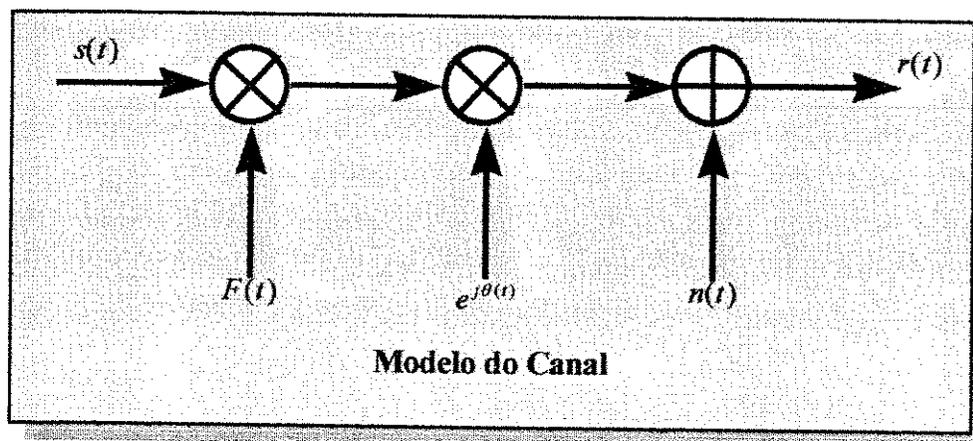


Figura 2.5 - Modelo do Canal

Neste modelo três processos aleatórios estão presentes durante o processo de transmissão e recepção de um sinal.

Processo de Desvanecimento: ($F(t)$): O processo de desvanecimento é considerado de variação lenta, onde por variação lenta entende-se que o valor do processo permanece constante durante a janela temporal de observação utilizada no receptor. Desta forma para um intervalo de N símbolos ($kT \leq t \leq (k+N)T$), $F(t)$ pode ser modelado com a amostra de um processo gaussiano complexo de média nula e variância σ_F^2 .

Se representarmos $F(t)$ na sua forma polar então $F(t) = \rho_k e^{j\Theta_k}$, onde “ ρ_k ” é a amplitude do ruído multiplicativo no intervalo de observação considerado, apresentando distribuição de Rayleigh e “ Θ_k ” a respectiva fase com distribuição uniforme no intervalo $(-\pi, \pi)$.

Processo de Fase aleatória ($e^{j\theta(t)}$): Supõe-se que o canal introduz uma fase aleatória $\theta(t)$ uniformemente distribuída no intervalo $(-\pi, \pi)$. Como no caso do processo desvanecimento, $\theta(t)$ é considerado de variação lenta, ou seja, para o intervalo $(kT \leq t \leq (k+N)T)$, $\theta(t) = \theta_k$, constante.

Processo Gaussiano Branco ($n(t)$): O processo gaussiano é assumido com sendo aditivo complexo de média nula e variância σ_n^2 .

Finalmente consideramos que o canal é sub-espalhado, ou seja, além do processo desvanecimento ser de variação lenta o canal é não seletivo ($B_d T_m < 1$).

2.2.1 - Caracterização do sinal enviado

Consideraremos aqui que a modulação utilizada é do tipo PSK. Esta escolha se justifica porque, como será considerado mais adiante o uso da codificação diferencial no transmissor, a sua realização é facilmente conseguida através do uso de constelações PSK.

Seja $s(t)$ o sinal enviado no intervalo de modulação $kT \leq t \leq (k+1)T$. Assumiremos que o transmissor envia a informação usando M formas de onda $\{s(t) = s_m(t); m = 1, 2, \dots, M\}$ de um alfabeto de modulação PSK e considerando-se a utilização de um pulso retangular em banda-básica da forma $u_T(t)$ para modular o sinal a ser transmitido, $s(t)$ pode ser escrito como,

$$s(t) = A \cos \left\{ 2\pi f_c t + \frac{2\pi}{M}(m-1) \right\} u_T(t) \quad kT \leq t \leq (k+1)T \quad (2.21)$$

onde A é a amplitude constante do sinal transmitido e $\phi = 2\pi/M(m-1)$ é a fase enviada.

Uma maneira conveniente de se representar o sinal para fins do estudo do sistema é através da sua representação complexa. Nesta representação, a parcela do sinal referente à

portadora é suprimida uma vez que não traz nenhuma informação adicional. Desta forma através da representação complexa é possível caracterizar completamente o sinal.

Seja $s(t)$ o sinal representado pela expressão (2.21). Então a sua representação complexa pode ser escrita na forma,

$$\begin{aligned} s_k &= Ae^{j\phi_k} \\ &= \sqrt{2P}e^{j\phi_k} \end{aligned} \tag{2.22}$$

onde P é a potência constante do sinal no intervalo de modulação e “ ϕ_k ” é a fase correspondente enviada. Nota-se que (2.22) traz toda a informação necessária para o processo de decisão no receptor.

Observa-se também que a representação complexa é igual a representação em banda básica, a menos de uma constante.

Como se está considerando que o processo desvanecimento é constante dentro da janela temporal de observação, o mesmo ocorrendo com a fase $\theta(t)$ introduzida pelo canal, o sinal recebido $r(t)$ pode ser escrito como,

$$r(t) = \rho_k e^{j\theta_k} s(t) e^{j\theta_k} + n(t) \quad ; kT \leq t \leq (k+1)T \tag{2.23}$$

A análise de (2.23) mostra que a expressão que define $r(t)$ é igual a expressão obtida para o caso onde o canal é do tipo AWGN, a menos da constante complexa $\rho_k e^{j(\theta_k + \theta_k)}$. Desta forma o receptor de correlação usado para o caso AWGN também pode ser utilizado aqui, pois conforme pode ser visto em [1], [15], neste tipo de receptor toda a estatística referente ao processo gaussiano $n(t)$ que influencia a estatística do sinal recebido $r(t)$ no processo de decisão é preservada na saída dos amostradores, e sendo $\rho_k e^{j(\theta_k + \theta_k)}$ uma constante, esta simplesmente será reproduzida na saída dos amostradores. Logo toda a informação necessária ao processo de decisão pode ser extraída das amostras obtidas no processo de recepção.

2.3 - O Receptor de Correlação

A filosofia por trás do receptor de correlação está em decompor o conjunto de sinais $\{ s_m(t) \}$ através de suas componentes sobre uma base de funções $\{ f_n(t); n = 1, 2, \dots, N \}$ (onde $N \leq M$) que é ortonormal e completa para este conjunto [2], [10]. Através da decomposição o sinal pode ser tratado de forma vetorial (“amostrada”) com componentes em relação as funções da base. Neste processo, se a base é completa então o sinal pode ser completamente caracterizado através da sua representação vetorial.

Na prática este processo equivale a pré-multiplicar o sinal por cada uma das funções da base, integrar o produto no intervalo de existência do sinal e amostrar o valor da integral ao final do intervalo. O valor das amostras em cada um dos correlacionadores corresponde a projeção do sinal na respectiva base.

Seja $S_k = [s_{k1}, s_{k2}, \dots, s_{kN}]$ o vetor de saída correspondendo as amostras nas saídas dos correlacionadores. Então $s(t)$ pode ser reconstruído através da combinação linear das funções da base pré-multiplicadas pelas correspondentes componentes do vetor S_k , ou seja,

$$s(t) = s_{k1}f_1(t) + s_{k2}f_2(t) + \dots + s_{kN}f_N(t) \tag{2.24}$$

Um esquema genérico deste tipo de receptor está mostrado na figura abaixo.

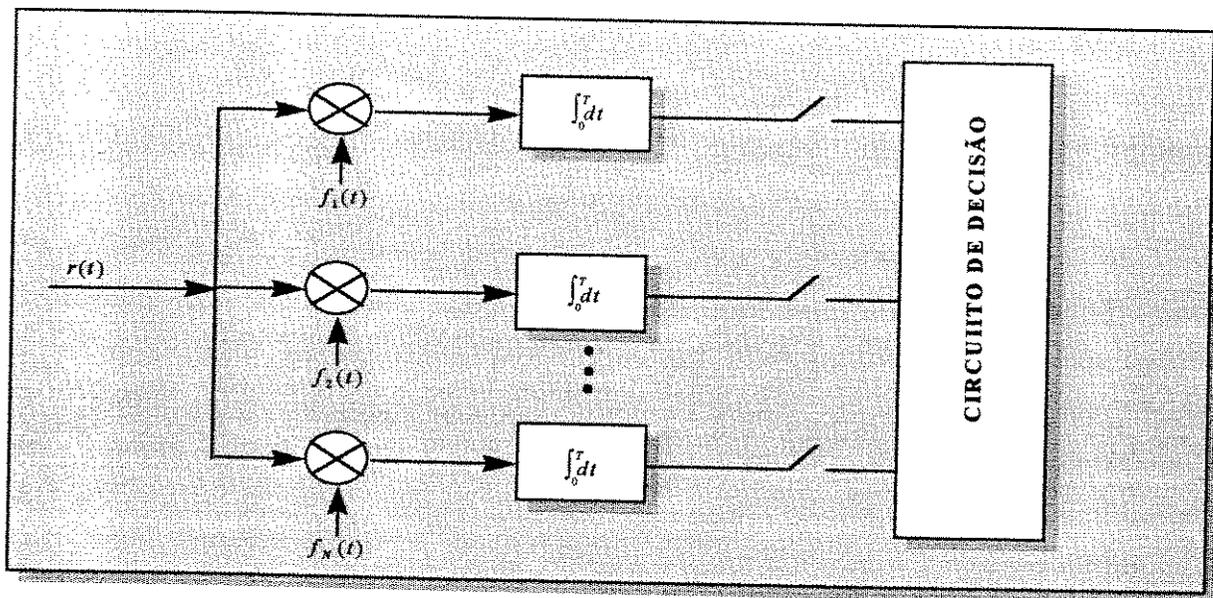


Figura 2.6 - Receptor de Correlação

Figura 2.6 - Receptor de Correlação

Como dito anteriormente neste tipo de receptor toda a estatística relevante ao processo de decisão é preservada nas amostras na saída dos amostradores.

Se representarmos o sinal $s(t)$ no intervalo $kT \leq t \leq (k+1)T$ através de (2.22), então pode-se mostrar facilmente que o receptor equivalente pode ser implementado conforme a figura abaixo.

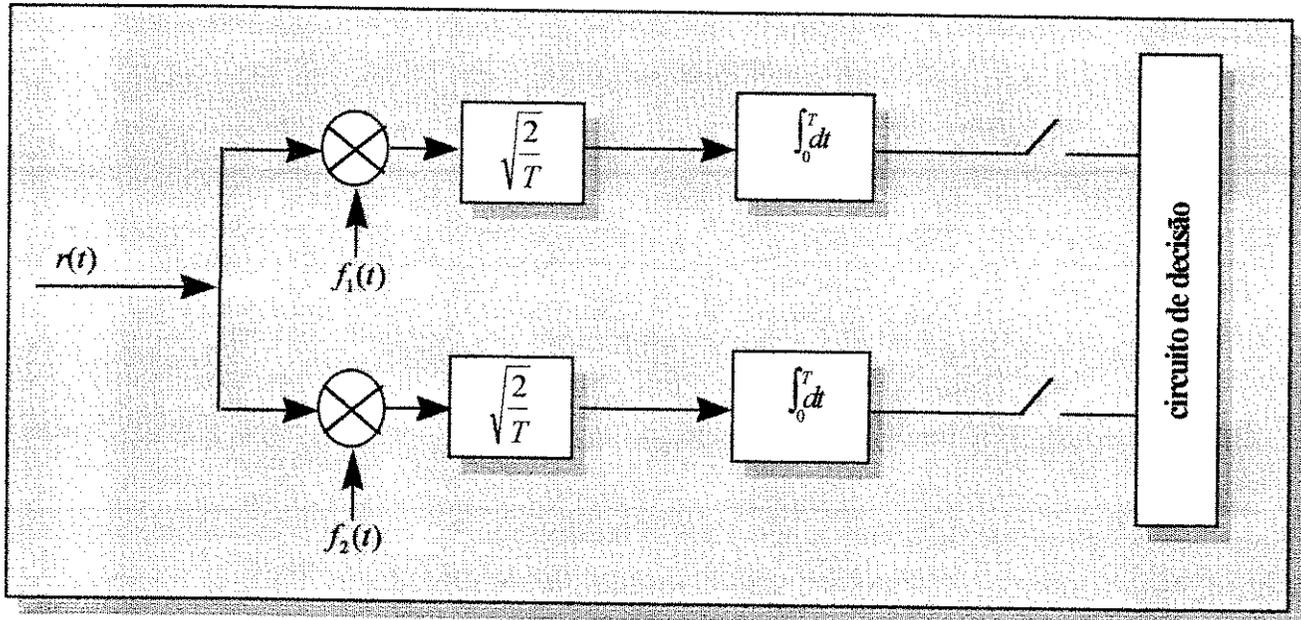


Figura 2.7 - Receptor de Correlação PSK

onde: $f_1(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_c t)$

$f_2(t) = \sqrt{\frac{2}{T}} \text{sen}(2\pi f_c t)$

O uso dos ganhos $\sqrt{2/T}$ utilizados na figura acima consiste apenas de um artifício de normalização para se estar em conformidade com a expressão matemática de (2.22). Assim o sinal r_k amostrado na saída do receptor no instante $(k+1)T$ pode ser escrito como:

$$\begin{aligned} r_k &= F_k e^{j\theta_k} s_k + n_k \\ &= \rho_k e^{j(\theta_k + \theta_k)} \sqrt{2P} e^{j\phi_k} + n_k \end{aligned} \tag{2.25}$$

Considerando-se que $n(t)$ é gaussiano branco de média nula e densidade espectral $N_0/2$, segue de maneira direta que as amostras n_k na saída dos correlacionadores possuem média zero e variância dada por:

$$\sigma_n^2 = \frac{1}{2} E[n_k^2] = \frac{N_0}{T} \quad (2.26)$$

Seja E_s a energia média dos sinais enviados. A fim de que a energia média dos sinais recebidos seja igual a energia no transmissor, a distribuição do ruído Rayleigh é normalizada de tal forma que a energia média na recepção seja igual à energia na transmissão, isto é, sendo E_m a energia média na recepção, então,

$$E_m = E[\rho_k^2 E_s] = E[\rho_k^2] E_s = E_s \quad (2.27)$$

Logo a distribuição de probabilidade do ruído multiplicativo é dada por:

$$p_\rho(\rho) = 2\rho e^{-\rho^2}; \rho \in (0, \infty) \quad (2.28)$$

Os conceitos básicos envolvendo sistemas de comunicação em um canal de desvanecimento foram discutidos neste capítulo. Mais adiante faremos uso dos resultados deste capítulo no desenvolvimento do trabalho.

Capítulo 3

Caracterização do Sistema

O objetivo deste capítulo é caracterizar o sistema que servirá de base para o trabalho a ser desenvolvido. Os conceitos introduzidos no capítulo anterior serão utilizados para a modelagem matemática do sistema. A partir do modelo matemático do sistema, serão extraídos os elementos necessários à obtenção de ferramentas matemáticas que possibilitem a análise e obtenção de critérios para a construção de códigos.

Podemos dizer que o sistema a ser considerado consiste essencialmente de um esquema de modulação PSK sobre o qual fazemos a utilização de 5 técnicas que visam minimizar os efeitos produzidos pela ação do meio de transmissão sobre os sinais enviados.

As técnicas que serão utilizadas e estudadas neste capítulo são:

Codificação/Decodificação diferencial.

Detecção multi-simbólica.

Utilização de informação de estado de canal.

Decisão por máxima verossimilhança.

Embaralhamento/Desembaralhamento dos símbolos de informação.

Existe ainda uma sexta técnica que será utilizada e que consiste no objetivo final deste trabalho, que vem a ser a codificação de bloco. A sua utilização porém está baseada justamente nas características extraídas do sistema que será descrito neste capítulo.

O sistema em estudo pode então ser definido como sendo um sistema de modulação M-DPSK com detecção multi-simbólica, com uso de informação de estado de canal e utilização de decodificação por máxima verossimilhança.

Antes de abordarmos cada uma das técnicas empregadas, é interessante que digamos algumas palavras que justifiquem a sua utilização dentro do contexto estudado.

Nos meio de transmissão que se caracterizam pela presença de fading, especificamente do tipo Rayleigh, onde não se tem uma linha de visada direta entre as antenas transmissora e o receptora, a aquisição e manutenção de uma coerência de fase pode se tornar dificultosa ou mesmo impossível, inviabilizando desta forma o uso de sistemas

coerentes. Nestas situações a utilização de sistemas que utilizem detecção diferencial se apresenta como uma alternativa para este problema, uma vez que estes não necessitam de uma coerência de fase.

É sabido no entanto que esquemas não coerentes sofrem uma considerável degradação em relação aos sistemas coerentes (quando a manutenção de uma coerência de fase é possível). A fim de podermos compensar esta perda devido a não coerência do sistema, e combater os efeitos do canal, mais quatro técnicas são utilizadas; detecção multi-simbólica, utilização de informação de estado de canal, decodificação por máxima verossimilhança e embaralhamento dos símbolos de informação.

Na detecção multi-simbólica o intervalo de observação no qual é realizado o processo de decisão se estende por mais de 2 símbolos (D-PSK usual). Estudos recentes em [3] demonstraram que esta técnica fornece ganhos significativos já para um pequeno acréscimo do intervalo de observação (3 e 4 símbolos) para canais AWGN. A sua eficácia para canais do tipo Rayleigh é um dos tópicos de investigação deste trabalho e será discutido nos próximos capítulos.

A informação de estado de canal, que consiste em uma estimação da amplitude do ruído multiplicativo produzido pelo canal e posteriormente utilizada no processo de decisão, é mais uma técnica que está sendo utilizada com o objetivo de melhorar o desempenho do sistema.

O algoritmo de decisão é baseado nas características estatísticas do canal de forma que a métrica utilizada, baseada na probabilidade de transição do canal, fornece um critério de escolha de máxima verossimilhança no processo de decisão. A utilização desta métrica no entanto pressupõe que o canal não possui memória. Desta forma a técnica de embaralhamento/desembaralhamento da informação transmitida no canal é utilizada para “quebrar” a memória do canal.

Assim as técnicas apresentadas procuram fornecer ao sistema condições de operação que possibilitem o estabelecimento de comunicação com o melhor desempenho possível.

O sistema em consideração é baseado no modelo apresentado em [4]. Esquemáticamente este modelo pode ser representado através da figura 3.1.

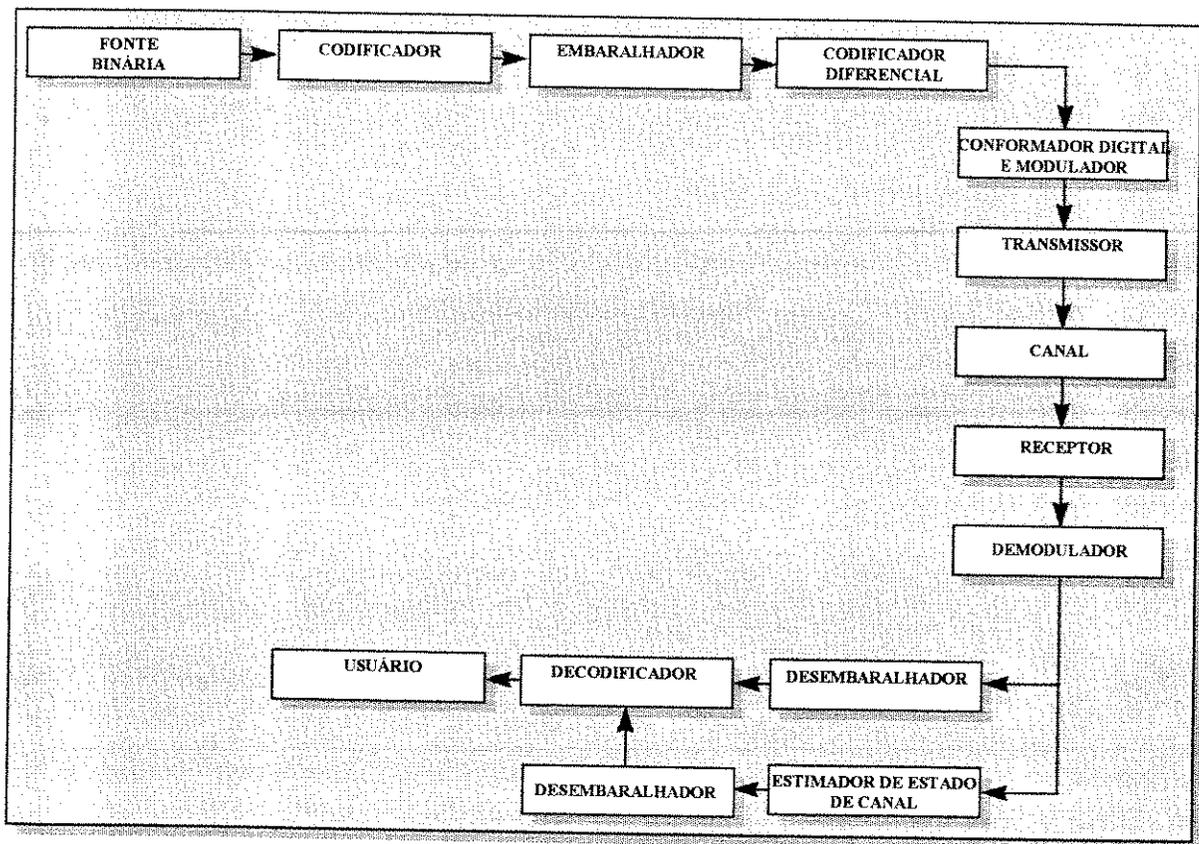


Figura 3.1 - Modelo do Sistema

A informação originada na fonte na forma binária é mapeada nas palavras código através do codificador, os símbolos na saída do codificador são reunidos em grupos de $N-1$ símbolos que passam por um processo de embaralhamento (do inglês “interleaving”) entre grupos. Em seguida os grupos são diferencialmente codificados, conformados e enviados no canal.

No receptor após a passagem por filtros casados e a remoção de alguma frequência Doppler que possa existir, a seqüência recebida de amostras r_k é particionada em grupos de N amostras cada, de tal forma que os grupos se sobreponham em uma amostra (Fig. 3.2). Esta sobreposição é necessária uma vez que o primeiro símbolo de cada grupo é usado como referência no algoritmo de detecção multi-simbólica utilizado em cada grupo. Paralelamente a este processo, a informação de estado de canal é estimada.

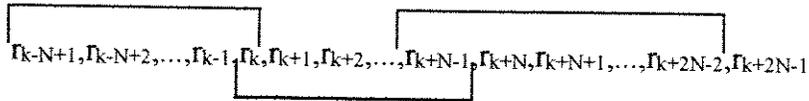


Figura 3.2 - Particionamento entre Grupos

A seqüência de amostras recebidas e a seqüência de valores de informação de estado de canal estimada passam então por um processo de desembaralhamento realizado grupo a grupo (de N-1 amostras). Finalmente após um processo de decodificação por máxima verossimilhança, baseado na seqüência de amostras e na informação de estado de canal, o decodificador fornece na sua saída a informação binária resultante do processo de decisão.

O embaralhamento aqui é assumido como sendo ideal, ou seja, estamos considerando que após o processo de embaralhamento/desembaralhamento os grupos são estatisticamente independentes entre si.

3.1 - Modelagem Matemática do Sistema

Na detecção muti-simbólica a decisão é baseada na observação simultânea de um grupo de símbolos, ao contrário da decisão símbolo a símbolo. Para isto é interessante supor que não há variação na fase da portadora durante o intervalo de observação considerado. Para o canal considerado, tanto o processo de desvanecimento quanto a fase aleatória introduzida pelo canal canal são considerados processos de variação lenta.

3.1.1 - Modelagem do Canal

Assumiremos neste estudo, que dentro o intervalo de existência dos grupos de observação, tanto o processo de desvanecimento quanto a fase aleatória introduzida pelo canal apresentam valores constantes, apenas o ruído aditivo Gaussiano varia de símbolo para símbolo. Esta suposição é razoável uma vez que estamos considerando que o desvanecimento e a fase aleatória introduzida pelo canal são processos de variação lenta.

Com base nestas suposições e para fins de análise, é mais conveniente a utilização de um modelo vetorial do canal compreendendo uma janela temporal correspondente a um grupo de observação.

Seja N o número de símbolos que compõem um grupo de observação e T o intervalo de modulação (correspondendo ao tempo de duração de símbolo). Então para o intervalo de duração de um grupo ($kT \leq t \leq (k + N)T$) o canal pode ser modelado através da Figura 3.3,

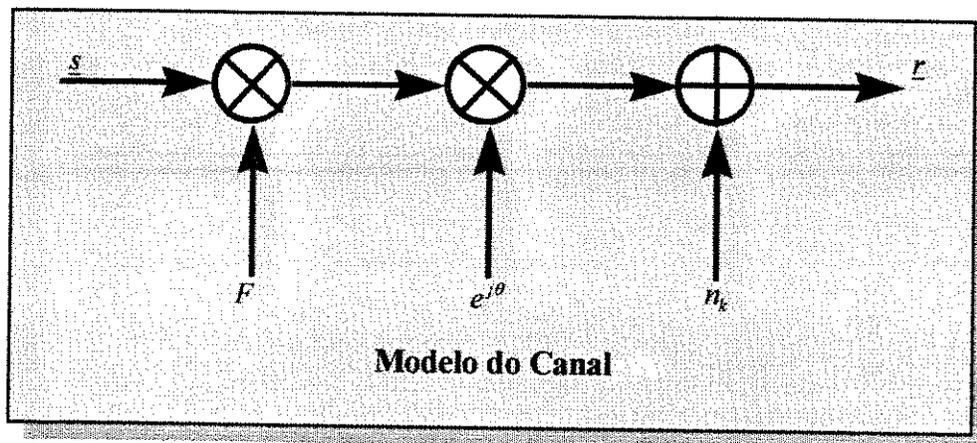


Figura 3.3 - Modelo do Canal

onde \underline{s} e \underline{r} (aqui representados por vetores coluna de N elementos) são as sequências dos símbolos enviados e das amostras recebidas no intervalo $kT \leq t \leq (k + N)T$ respectivamente.

Seja s_k o símbolo enviado no intervalo de modulação $kT \leq t \leq (k + 1)T$. Então temos que a sua representação complexa é dada por

$$s_k = \sqrt{2P} e^{j\phi_k} \tag{3.1}$$

O processo de desvanecimento dentro do intervalo de observação é representado no modelo vetorial através da constante F , onde F representa a amostra de uma gaussiana complexa normalizada ($E\{|F|^2\} = 1$) e de média nula. Escrevendo F na sua forma polar temos

$$F = \rho e^{j\theta} \tag{3.2}$$

onde ρ apresenta distribuição de Rayleigh e Θ é uniformemente distribuída no intervalo de $(-\pi, \pi)$.

O processo de fase aleatória introduzida pelo canal é representado no modelo vetorial pela constante $e^{j\theta}$, onde θ é uniformemente distribuída no intervalo $(-\pi, \pi)$ e constante dentro do intervalo de observação.

O processo aditivo Gaussiano branco é modelado através das constantes n_k , que representam as amostras de uma Gaussiana complexa de média nula e variância por dimensão definida por:

$$\sigma_n^2 = \frac{1}{2} E\{n_k|^2\} = \frac{N_0}{T} \quad (3.3)$$

É importante salientar que as amostras do ruído Gaussiano aditivo n_k variam de símbolo para símbolo, ao contrário do processo fading e da fase introduzida pelo canal que são constantes durante todo o intervalo de duração do grupo de observação.

A seqüência recebida \underline{r} , pode então ser expressa como,

$$\underline{r} = F e^{j\theta} \underline{s} + \underline{n} \quad (3.4)$$

onde r_{k-i} , s_{k-i} e n_{k-i} são, respectivamente, os i -ésimos elementos $i=0, 1, \dots, N-1$, dos vetores coluna $(N \times 1)$ \underline{r} , \underline{s} , \underline{n} .

3.2 - Métrica de Máxima Verossimilhança

A métrica utilizada no algoritmo de decisão esta baseada na densidade de probabilidade de transição do canal $p(\underline{r}|\underline{s})$. Em outras palavras, a métrica de máxima verossimilhança é aquela que procura maximizar o valor de $p(\underline{r}|\underline{s})$. Isto é, para uma dada seqüência recebida \underline{r} , qual a seqüência \underline{s} que maximiza $p(\underline{r}|\underline{s})$.

Para o caso em estudo estamos considerando que o sistema estima de maneira exata da informação de estado de canal, que consiste no valor da amplitude ρ da amostra do ruído

multiplicativo dentro do intervalo de observação. Isto constitui a chamada informação ideal de estado de canal, que passará a ser denominada CSI (do inglês “Channel State Information”). A obtenção prática da CSI pode se feita como descrito em [12].

Para o caso onde se tem detecção mutli-simbólica e utilização de CSI, foi mostrado em [4] que densidade de probabilidade de transição do canal é dada por,

$$p(\underline{r} | \underline{s}) = \frac{1}{(2\pi\sigma_n^2)^V} \exp\left\{-\frac{1}{2\sigma_n^2} \left[|\underline{r}|^2 + \rho^2|\underline{s}|^2\right]\right\} I_0\left(\rho\sigma_n^{-2}|\underline{r}^T \underline{s}^*\right) \quad (3.5)$$

Uma vez que o logaritmo neperiano é uma função monotonicamente crescente do seu argumento, então maximizar $p(\underline{r} | \underline{s})$ sobre \underline{s} é equivalente a maximizar $\ln p(\underline{r} | \underline{s})$ sobre \underline{s} , o que resulta na seguinte regra de decisão: Escolha a seqüência \underline{s} que maximiza a métrica

$$\eta = -\frac{1}{2\sigma_n^2} \left(\rho^2|\underline{s}|^2\right) + \ln I_0\left(\rho\sigma_n^{-2}|\underline{r}^T \underline{s}^*\right) \quad (3.6)$$

Temos que devido ao fato da constelação utilizada ser do tipo PSK, o primeiro termo da expressão (3.6) é independente do vetor \underline{s} enviado. Desta forma o primeiro termo desta expressão pode ser desprezado. Assim para o caso sistema em estudo a métrica de máxima verossimilhança é dada por,

$$\eta = \ln I_0\left(\rho\sigma_n^{-2}|\underline{r}^T \underline{s}^*\right) \quad (3.7)$$

Esta métrica, no entanto, seria de difícil utilização em uma implementação prática. A expressão $\ln I_0(x)$ pode ser aproximada assintoticamente para pequenos valores do seu argumento por $\ln I_0(x) \cong x^2/4$, e por $\ln I_0(x) = x$ para valores elevados de seu argumento.

Assim após uma normalização por σ_n^2 e utilizando as aproximações para $\ln I_0(x)$, a métrica pode ser simplificada por,

$$\eta = \rho^t \left|\underline{r}^T \underline{s}^*\right|^t \quad (3.8)$$

onde $l = 1$ para altas relações sinal-ruído e $l = 2$ para baixas relações sinal ruído. Embora esta métrica passe a ser sub-ótima e uma certa degradação talvez seja o preço a se pagar, em contra partida ganha-se em simplicidade em termos de implementação para circuitos práticos.

Note-se que em (3.8) manteve-se o valor conhecido da amplitude do fading ρ como parte da métrica, a despeito do fato deste ser independente da vetor enviado. O motivo para isto é que na codificação de bloco, onde as palavras do código constituem-se de seqüências compostas por mais de um grupo de observação, o valor do fading pode variar de grupo para grupo e desta forma a métrica total da seqüência é função do conjunto de valores de amplitudes do fading ao longo da seqüência.

Rescrevendo a métrica (3.8) e utilizando a fórmula do sinal PSK, a métrica pode ser expressa como

$$\eta = \rho^l \left| \sum_{i=0}^{N-1} r_{k-i} e^{j\phi_{k-i}} \right|^l \quad (3.9)$$

Note-se que em (3.9) a amplitude dos sinais foi normalizada por $\sqrt{2P}$. Esta normalização acarreta em uma modificação no valor da variância das amostras do ruído Gaussiano branco aditivo quando (3.9) é utilizada, pois o valor equivalente das amostras passa a ser dado por $n_k/\sqrt{2P}$ (este resultado pode ser mostrado facilmente através do cálculo de $r_{k-i} s_{k-i}$).

Assim a variância equivalente das amostras do ruído Gaussiano branco aditivo passa a ser dada por,

$$\begin{aligned} \sigma_n^2 &= \frac{1}{2} E \left\{ \left| \frac{n_k}{\sqrt{2P}} \right|^2 \right\} = \frac{No}{T} \times \frac{T}{2E_s} \\ \sigma_n^2 &= \frac{No}{2E_s} \end{aligned} \quad (3.10)$$

onde E_s é a energia por símbolo de modulação para uma constelação PSK e assumindo-se que os símbolos são equiprováveis, este valor coincide com a energia média dos símbolos.

A métrica dada por (3.9) resulta então na seguinte regra de decisão

$$\text{Escolha } \hat{\Phi} \text{ se } \left| \sum_{i=0}^{N-1} r_{k-i} e^{j\hat{\phi}_{k-i}} \right|^l \text{ é máxima}$$

onde $\hat{\Phi}$ é uma seqüência particular de fases enviadas.

3.3 - Codificação Diferencial

A métrica (3.9) gera uma ambigüidade no processo de decisão, pois suponha que uma fase ϕ_a foi introduzida (constante para todo o grupo de comprimento N) durante a transmissão. Então

$$\text{Max } \rho^l \left| \sum_{i=0}^{N-1} r_{k-i} e^{-j\hat{\phi}_{k-i}} e^{j\phi_a} \right|^l = \text{Max } \rho^l \left| \sum_{i=0}^{N-1} r_{k-i} e^{-j\hat{\phi}_{k-i}} \right|^l.$$

Logo, embora a decisão seja tomada corretamente em valor absoluto, ela difere da seqüência enviada. Por exemplo suponha que ϕ_a é um múltiplo das fases de informação. Neste caso teríamos uma decisão equivocada da seqüência de envio. Através de um processo de codificação/decodificação diferencial, esta ambigüidade é evitada.

Para um grupo de comprimento N o processo de codificação e decodificação pode ser realizado como se segue.

No codificador diferencial, a cada intervalo de modulação a fase enviada no canal consiste da soma da fase correspondente a informação com a fase enviada no canal no intervalo anterior.

Esquemáticamente temos a representação da Figura 3.4.

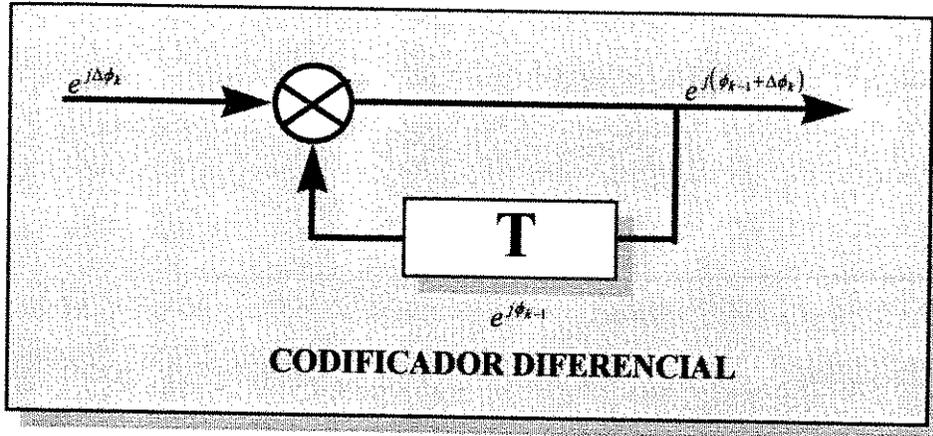


Figura 3.4 - Codificador Diferencial

$\Delta\phi_k$ - Fase correspondente à informação no intervalo $kT = t = (k+1)T$

ϕ_k - Fase enviada no canal no intervalo $kT = t = (k+1)T$

ϕ_{k-1} - Fase enviada no intervalo anterior

A fase enviada no canal em um intervalo genérico de modulação dentro do grupo de observação pode ser escrita como:

$$\phi_{k-i} = \sum_{m=0}^{N-i-2} \Delta\phi_{k-i-m} + \phi_{k-N+1} \quad (3.11)$$

Desta forma a métrica pode ser escrita alternativamente como :

$$\eta = \rho^i \left| \sum_{l=0}^{N-1} r_{k-i} e^{-j(\phi_{k-i} - \phi_{k-N+1})} \right|^i \quad (3.12)$$

Note que nesta nova versão se uma fase ϕ_a for introduzida ao bloco, então todos os símbolos serão modificados ($\phi_{k-i}^i = \phi_{k-i} + \phi_a$), e no processo de decisão ϕ_a é cancelado, uma vez que,

$$\phi_{k-i}^i - \phi_{k-N+1}^i = \phi_{k-i} + \phi_a - (\phi_{k-N+1} + \phi_a) = \phi_{k-i} - \phi_{k-N+1} = \sum_{m=0}^{N-i-2} \Delta\phi_{k-i-m} \quad (3.13)$$

Portanto a métrica de decisão se torna:

$$\eta = \rho^j \left| r_{k-N+1} + \sum_{i=0}^{N-2} r_{k-i} e^{-j \sum_{m=0}^{N-i-2} \Delta \phi_{k-i-m}} \right|^j \quad (3.14)$$

Esta métrica implica que observamos o sinal recebido sobre um intervalo de N símbolos e desta observação fazemos a decisão simultânea sobre $N-1$ fases de informação. Observamos que devido à codificação diferencial a métrica não depende de ϕ_{k-N+1} e este pode sempre ser considerado como $\phi_{k-N+1} = 0$.

3.4 - Métrica de Decisão de Seqüência

Os resultados obtidos até aqui se referem a métrica de decisão para um grupo de observação. A extensão destes resultados para uma seqüência seguem de maneira direta.

Para o caso codificado (codificação de bloco) as palavras do código consistem em seqüências compostas por determinado número de grupos. Neste caso a métrica de decisão da seqüência é o somatório das métricas de cada grupo que compõem a seqüência.

Seja uma seqüência (palavra código) composta por k grupos de informação. Seja o i -ésimo grupo de N símbolos denotado pelo vetor coluna \underline{r}_i e seja a seqüência total de k grupos denotada por \mathbf{r} . Similarmente seja \underline{s}_i o i -ésimo grupo de símbolos de transmissão, e \mathbf{s} a seqüência total transmitida. Assumindo um processo de embaralhamento/desembaralhamento perfeito, os grupos que compõem a seqüência são estatisticamente independentes, e a probabilidade condicional de \mathbf{r} dado \mathbf{s} é

$$p(\mathbf{r} | \mathbf{s}) = \prod_i p(\underline{r}_i | \underline{s}_i) \quad (3.15)$$

Desta forma segue de maneira direta que a métrica total para a seqüência é dada por:

$$\eta = \sum_i \eta_i$$

$$\eta = \sum_i \rho_i^l \left| \underline{r}_i^T \underline{s}_i^{*l} \right|^l \quad (3.16)$$

3.5 - Embaralhamento

Na maioria dos meios de transmissão, o canal atua sobre a informação enviada introduzindo interdependência estatística entre os símbolos sucessivos de modulação. Este fenômeno é conhecido como “memória estatística do canal”. Este comportamento é freqüentemente observado em canais de desvanecimento, onde o desvanecimento pode variar de forma lenta se comparado ao intervalo de duração de símbolo.

Esta memória introduzida pelo canal pode degradar consideravelmente o desempenho de códigos projetados para canais sem memória (e.g. canais AWGN). Um exemplo desta situação pode ser visto em [5].

Técnicas de codificação baseadas nos parâmetros estatísticos da memória do canal já foram propostas. Porém em muitos casos estes parâmetros são variáveis no tempo e de difícil determinação. Desta forma códigos que apresentam um bom desempenho para um determinado conjunto de parâmetros podem apresentar uma grande perda de performance quando há variação destes parâmetros.

Uma técnica que não requer um conhecimento da memória do canal, mas apenas da sua duração aproximada, é o embaralhamento.

A memória do canal, na prática, é função do espaçamento temporal entre os símbolos sucessivos, sendo tanto menor quanto maior é este espaçamento. Então se todos os símbolos de uma dada palavra código forem transladados em intervalos temporais suficientes, ao mesmo tempo em que estes intervalos são preenchidos com símbolos provenientes de outras palavras código, de forma que se mantenha um espaçamento regular entre os símbolos de uma mesma palavra código, então a dependência estatística entre símbolos sucessivos de uma mesma palavra código, pode ser praticamente eliminada e o canal passa a se comportar com sendo sem memória.

Uma maneira possível de implementar o processo de embaralhamento é através de um arranjo matricial, conforme mostrado na Figura 2.7. As palavras na saída do codificador são armazenadas na matriz, linha a linha, sendo que o sentido de leitura para o envio no canal é feito, coluna a coluna. Um embaralhamento de grau M consiste de uma matriz de M linhas (M palavras código). No receptor é realizado o processo inverso (desembaralhamento), e as palavras código originais são reconstituídas.

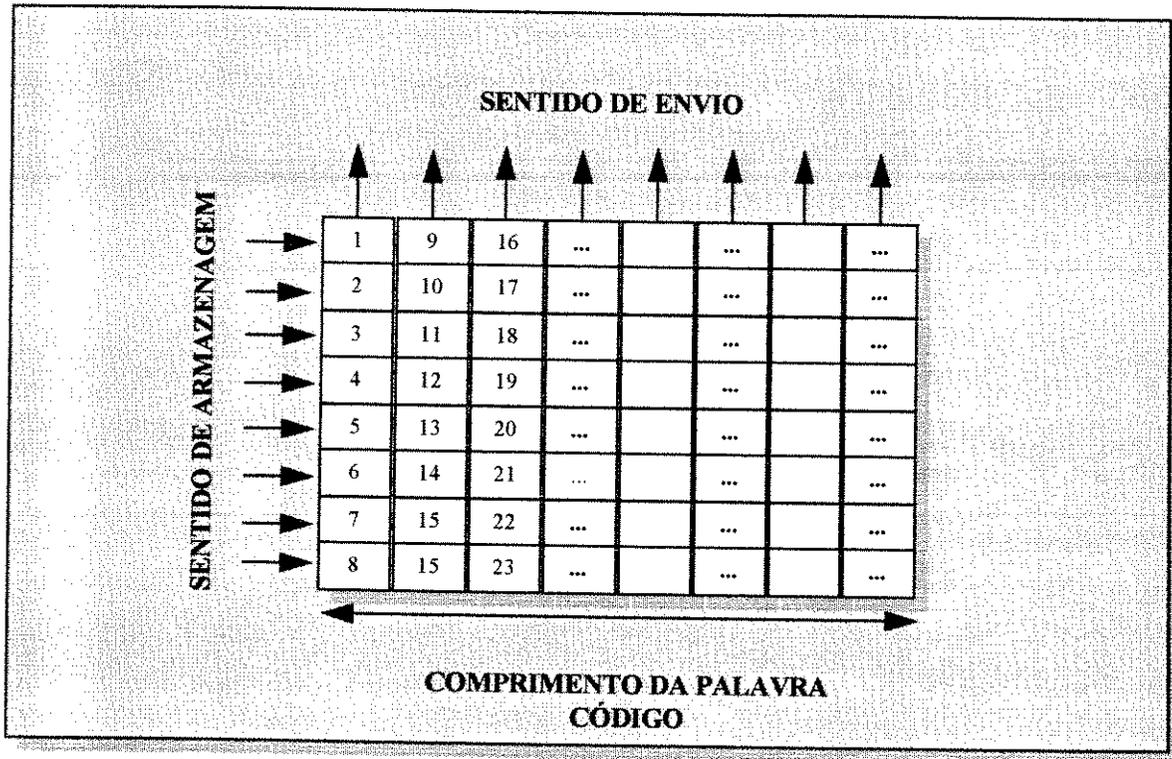


Figura - 2.7 Matriz de Embaralhamento

Desta forma, através do processo de embaralhamento/desembaralhamento, códigos projetados para canais sem memória podem ser implementados em canais com memória.

Uma vez definidos o sistema a ser utilizado, o modelo estatístico do canal e a forma matemática da variável de decisão, o passo seguinte consiste na determinação de limitantes de taxas de codificação e de erro e a extração de critérios que possam ser utilizados para a determinação de bons códigos. Como será visto na próxima parte do trabalho, a determinação destes limitantes se mostrou dificultosa e só foi possível através de análise numérica, o que nos levou a buscar aproximações que conduzissem a resultados algébricos mais simples de serem analisados a fim de se obter critérios que pudessem ser utilizados para a construção de códigos.

Capítulo 4

Limitante de Erro

Um limitante amplamente utilizado no cálculo de probabilidade de erro é o fator de Chernoff. Como se está interessado na determinação de critérios para construção de códigos, o fator de Chernoff fornece uma maneira de expressar algebricamente um limitante para a probabilidade de erro, dando condições de se extrair critérios através da análise de sua expressão matemática, [8], [9].

Seja Δ uma variável aleatória com função densidade de probabilidade $p_{\Delta}(\Delta)$. A probabilidade de Δ assumir valores positivos pode ser limitada pelo fator de Chernoff através da expressão,

$$P(\Delta \geq 0) \leq E_{\Delta}[\text{Exp}(\lambda\Delta)] \quad (4.1)$$

onde λ é uma constante escolhida de forma a minimizar o lado direito de (4.1), conforme [6].

Seja η_C a métrica de decisão correspondente a uma palavra código C enviada¹ e $\eta_{\hat{C}}$ a métrica concorrente correspondendo a uma palavra código \hat{C} . Uma decisão errada será tomada sempre que $\eta_{\hat{C}} > \eta_C$, ou seja sempre que a métrica de uma palavra concorrente superar em valor a métrica correspondente à seqüência enviada. Desta forma definimos Δ como

$$\begin{aligned} \Delta &= \eta_{\hat{C}} - \eta_C \\ \Delta &= \sum_{i \in n} \rho_i^2 (|r_i^T \hat{s}_i^*|^2 - |r_i^T s_i^*|^2) \end{aligned} \quad (4.2)$$

¹ Neste capítulo assumiremos que a métrica de decisão é dada pela expressão (3.16) para baixas relações sinal ruído ($l = 2$).

Onde “ n ” é o conjunto dos grupos para os quais as duas seqüências diferem. Desta forma a probabilidade de erro entre duas palavras código (Probabilidade de erro para a par) pode ser escrita como:

$$P_{E2} = \text{Prob}(C \rightarrow \hat{C}) = \text{Prob}(\Delta \geq 0) \quad (4.3)$$

Aqui $\text{Prob}(C \rightarrow \hat{C})$ é a probabilidade de enviar a palavra C e decidir pela palavra \hat{C} .

Através da utilização da variável Δ é possível determinar o fator de Chernoff que limita a probabilidade par a par, pois através de (4.1) temos que

$$P_{E2} \leq E_{\Delta}[\text{Exp}(\lambda\Delta)] \quad (4.4)$$

Neste ponto duas observações são importantes no desenvolvimento do cálculo:

1^o) Está-se considerando que a amplitude “ ρ ” do ruído multiplicativo é conhecida, desta forma a variável de decisão Δ está condicionada a “ ρ ”.

2^o) Devido à suposição de embaralhamento perfeito, os grupos de observação são estatisticamente independentes.

Utilizando estes dois fatos e a expressão geral para o fator de Chernoff dada por (4.1) temos que, denotando o fator de Chernoff entre duas palavras C e \hat{C} por $C_{\lambda}(C \rightarrow \hat{C})$,

$$C_{\lambda}(C \rightarrow \hat{C}) = E_{\rho}[E_{\Delta}[\text{Exp}(\lambda\Delta)]] \quad (4.5)$$

Devido à independência entre os grupos de observação,

$$\Delta = \sum_{i \in n} \Delta_i; \quad \text{onde } \Delta_i = \rho_i^2 \left(|r_i^T \underline{s}_i^*|^2 - |r_i^T \underline{s}_i|^2 \right), \text{ e}$$

$$C_{\lambda}(C \rightarrow \hat{C}) = \prod_{i \in n} E_{\rho_i}[E_{\Delta_i}[\text{Exp}(\lambda\Delta_i)]] \quad (4.6)$$

Uma vez que o fator de Chernoff de um par de palavras código pode ser escrito como o produto dos fatores de Chernoff de cada grupo de observação, na análise que se segue o cálculo será desenvolvido para um grupo de observação, sendo que para os demais a

expressão resultante só irá variar no que se refere aos termos indexados que assumem valores distintos de grupo para grupo. Por uma questão de simplicidade o índice "i" será suprimido na análise seguinte.

Para um grupo de observação estamos interessados no cálculo de,

$$\text{Min}_{\lambda} E_{\rho}[E_{\Delta}[\text{Exp}(\lambda\Delta)]] \quad (4.7)$$

A determinação de $E_{\rho}[E_{\Delta}[\text{Exp}(\lambda\Delta)]]$ pode ser conseguida, observando-se que Δ se encontra na forma Hermitiana quadrática de variáveis gaussianas complexas [7], ou seja Δ pode ser escrita na forma,

$$\Delta = \underline{W}^{*T} F \underline{W} \quad (4.8)$$

Onde \underline{W} é um vetor coluna de variáveis complexas gaussianas conjuntamente distribuídas e F é uma matriz Hermitiana ($F^T = F^*$).

Definindo:

$$\hat{w} = \rho \hat{s}^{T*} \underline{r} \quad (4.9)$$

$$w = \rho s^{T*} \underline{r} \quad (4.10)$$

Então \hat{w} e w são somas de variáveis aleatórias gaussianas complexas pré multiplicadas por constantes, e portanto também são gaussianas complexas.

Definindo

$$\underline{W} = [\hat{w}, w]^T \quad (4.11)$$

$$F = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (4.12)$$

Temos que



Temos que

$$\Delta = \underline{W}^{*T} F \underline{W} = \rho^2 (|\hat{w}|^2 - |w|^2) \quad (4.13)$$

A expressão $E_{\Delta}[Exp(\lambda\Delta)]$ que passará a ser denotada por $\Delta\Phi(\lambda/\rho)$, i.e.,

$$\Delta\Phi(\lambda/\rho) = E_{\Delta/\rho}[Exp(\lambda\Delta)] \quad (4.14)$$

pode ser vista como a função característica da variável Δ para $j\xi = \lambda$, condicionada a amplitude do ruído multiplicativo “ ρ ”. Desta forma para o cálculo de (4.14) basta determinarmos a função característica de Δ . Este cálculo pode ser feito baseado apenas no conhecimento da matriz de covariâncias do vetor \underline{W} e do vetor média de \underline{W} .

4.1 - Determinação da Matriz de Covariâncias e do Vetor Esperança de \underline{W}

A matriz de covariância denotada aqui por R é definida como

$$R = \frac{1}{2} E[(W - \langle W \rangle)^* (W - \langle W \rangle)^T] \quad (4.15)$$

Determinação do vetor esperança de \underline{W} :

Definindo a quantidade $\delta = \underline{\hat{s}}^T \underline{s}$, podemos escrever o vetor esperança como:

$$E[\underline{W}] = [\rho^2 \delta e^{j(\theta+\Theta)}, \rho^2 N e^{j(\theta+\Theta)}] \quad (4.16)$$

E a matriz R de covariâncias como:

$$R = \frac{\rho^2 \sigma_n^2}{2} \begin{bmatrix} N & \delta^* \\ \delta & N \end{bmatrix} \quad (4.17)$$

Através da análise da matriz de covariância R , vê-se que as variáveis \hat{w} e w do vetor gaussiano complexo \underline{W} são correlacionadas. De forma a simplificar o cálculo podemos aplicar uma transformação linear para obter variáveis independentes e com matriz de covariância igual à matriz identidade, através do uso dos autovalores e autovetores da matriz R .

Seja Λ a matriz de autovalores de R . Então existe uma matriz Ψ tal que $\Lambda = \Psi^* \Psi^T$, e U , a matriz de autovetores de R , tal que $R = U \Lambda U^*$. A transformação

$$\underline{Z} = \Psi^{-1} U^T \underline{W} \quad (4.18)$$

leva o conjunto $\{ \hat{w}, w \}$ ao conjunto $\{ \hat{z}, z \}$, que tem componentes estatisticamente independente e com variância unitária.

Desta forma temos que Δ pode ser escrito na forma:

$$\Delta = \underline{Z}^{*T} (\Psi^{*T} U^T F U^* \Psi) \underline{Z} = \underline{Z}^{*T} T \underline{Z} \quad (4.19)$$

$$T = \Psi^{*T} U^T F U^* \Psi \quad (4.20)$$

A matriz T pode ainda ser diagonalizada através da sua matriz de autovalores Φ e sua matriz de autovetores S , sendo que

$$T = S \Phi S^{*T} \quad (4.21)$$

Introduzindo a transformação,

$$\underline{\Omega} = S^{*T} \underline{Z} \quad (4.22)$$

podemos escrever a variável de decisão Δ na forma,

$$\Delta = \underline{\Omega}^{*T} \Phi \underline{\Omega} = \phi_1 |\omega_1|^2 + \phi_2 |\omega_2|^2 \quad (4.23)$$

onde ω_i são variáveis estatisticamente independentes de variância unitária, e

$$\varpi_i = \alpha_i + j\beta_i \quad (4.24)$$

e

$$\langle \underline{\Omega} \rangle = S^{*T} \Psi^{-1} U^T \langle \underline{W} \rangle \quad (4.25)$$

A função característica de Δ é definida como a transformada de Fourier sobre a sua função densidade de probabilidade. Assim

$$\Delta\Phi(j\xi | \rho) = \int_{-\infty}^{\infty} \text{Exp}(j\xi \Delta) p_{\Delta|\rho}(\Delta) d\Delta = E_{\Delta|\rho}[\text{Exp}(j\xi \Delta)] \quad (4.26)$$

Realizando este cálculo concluímos que $\Delta\Phi(j\xi)$ pode ser escrito como

$$\Delta\Phi(j\xi | \rho) = \prod_{i=1}^2 \frac{1}{(1 - 2j\xi\phi_i)} \text{Exp} \left[\frac{j\xi\phi_i}{(1 - 2j\xi\phi_i)} (\langle \alpha_i \rangle^2 + \langle \beta_i \rangle^2) \right] \quad (4.27)$$

4.2 - Obtenção dos Resultados Numéricos

Através do cálculo dos autovalores de R a matriz Λ é dada por

$$\Lambda = \frac{\rho^2 \sigma_n^2}{2} \begin{bmatrix} N + |\delta|^2 & 0 \\ 0 & N - |\delta|^2 \end{bmatrix} \quad (4.28)$$

A matriz U de autovetores de R é dada por

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} \delta^* & -\delta^* \\ |\delta| & |\delta| \\ 1 & 1 \end{bmatrix} \quad (4.29)$$

A matriz Ψ deve ser tal que $\Lambda = \Psi^* \Psi^T$. Uma solução possível é fazer Ψ de forma que $\psi_i = \lambda^{1/2}$. Assim:

$$\Psi = \frac{\rho \sigma_n^2}{\sqrt{2}} \begin{bmatrix} \sqrt{N+|\delta|} & 0 \\ 0 & \sqrt{N-|\delta|} \end{bmatrix} \quad (4.30)$$

A matriz T é dada por $T = \Psi^{*T} U^T F U^* \Psi$. Desta forma:

$$T = \frac{\rho^2 \sigma_n^2 \sqrt{N^2 - |\delta|^2}}{2} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (4.31)$$

Novamente, através do cálculo dos autovalores de T temos que sua matriz de autovalores é dada por:

$$\Phi = \frac{\rho^2 \sigma_n^2 \sqrt{N^2 - |\delta|^2}}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (4.32)$$

A matriz S de autovalores de T é dada por

$$S = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \quad (4.33)$$

Temos também que $|\langle \underline{\Omega} \rangle|^2$ é dado por

$$|\langle \underline{\Omega} \rangle|^2 = \frac{\rho^2}{\sigma_n^2} \frac{N - \sqrt{N^2 - |\delta|^2}}{N + \sqrt{N^2 - |\delta|^2}}, \quad (4.34)$$

onde

$$\langle \underline{\Omega} \rangle = S^{*T} \Psi^{-1} U^T \langle W \rangle \quad (4.35)$$

Substituindo os valores obtidos na expressão algébrica da função característica e substituindo $j\xi$ pela variável λ temos

$$\phi_2 = -\phi_1$$

Denotando: $\phi_1 = \phi$

$$\Delta\Phi(\lambda | \rho) = \frac{1}{(1-4\lambda^2\phi^2)} \text{Exp} \left\{ \frac{\rho^2}{\sigma_n^2} \left[\frac{-2\sqrt{N^2 - |\delta|^2\phi\lambda + 4N\phi^2\lambda^2}}{(1-4\lambda^2\phi^2)} \right] \right\} \quad (4.36)$$

É importante salientar que $\Delta\Phi(\lambda | \rho)$ é a expressão do fator de Chernoff condicionada a um valor de “ ρ ” o que pode ser visto através de (4.14). Desta forma o fator de Chernoff é alcançado calculando-se a esperança sobre o ruído multiplicativo “ ρ ”. Através da análise de (4.36), vemos porém que para um valor fixo de “ λ ” a função torna-se negativa para certos valores de “ ρ ” no cálculo de (4.7), o que forneceria resultados não válidos (fora da região de convergência do fator de Chernoff) no cálculo da esperança sobre o ruído multiplicativo, invalidando os resultados

Observamos que para um valor fixo de ruído, situação em que a amplitude do ruído “ ρ ” poderia ser vista como um atenuação digamos “ ρ_x ” fixa e conhecida provocada pelo canal, $\Delta\Phi(\lambda | \rho) = \Delta\Phi'(\lambda)$ representaria o fator de Chernoff que caracterizaria este canal e desta forma poderíamos encontrar λ que minimizaria $\Delta\Phi'(\lambda)$.

Se fizermos agora uma varredura em todos os fatores de Chernoff de cada um dos canais caracterizados pela correspondente atenuação “ ρ_x ” onde $\rho_x \in (0, \infty)$ e ponderarmos pela probabilidade de ocorrência destes canais então podemos dizer que este resultado limita a expressão (4.7) para um grupo de observação. Utilizando este artifício o problema inicialmente encontrado no cálculo de (4.7) pode ser resolvido. Note-se que para cada valor de amplitude do ruído “ ρ_x ” no cálculo da esperança sobre o ruído multiplicativo, existe um $\lambda = f(\rho_x)$ que minimiza $\Delta\Phi'(\lambda)$.

Desta forma, o que se está fazendo na realidade é uma média de fatores de Chernoff, pois para cada valor que “ ρ ” assume no cálculo da esperança, o fator de Chernoff correspondente é calculado e o valor obtido é ponderado pela probabilidade de ocorrência do valor de “ ρ_x ” correspondente.

A interpretação deste artifício pode ser vista como sendo uma consequência da variação temporal do canal, acarretando que para cada valor de ruído (caracterizando um canal) existe um fator de Chernoff ótimo distinto.

4.2.1 - Minimização de $\Delta\Phi'(\lambda)$

Uma vez que $\lambda = f(\rho_x)$, uma simplificação na expressão pode ser alcançada observando-se que a quantidade 2ϕ é comum a todos os termos em “ λ “. Podemos supor então que “ λ “ absorve esta quantidade. Rescrevendo $\Delta\Phi'(\lambda)$, temos

$$\Delta\Phi'(\lambda) = \frac{1}{(1-\lambda^2)} \text{Exp} \left\{ \frac{\rho_x^2}{\sigma_n^2} \left[\frac{-\sqrt{N^2 - |\delta|^2} \lambda + N\lambda^2}{(1-\lambda^2)} \right] \right\} \quad (4.37)$$

A minimização de $\Delta\Phi'(\lambda)$ em relação a “ λ “ pode ser obtida através do cálculo de $\frac{\partial \Delta\Phi'(\lambda)}{\partial \lambda} = 0$.

A solução desta expressão conduz a um polinômio de 3º grau onde “ λ “ é uma das três raízes possíveis e deve estar contida no intervalo (0,1) a fim de que $\Delta\Phi'(\lambda) > 0$.

Tem-se:

$$\frac{\partial \Delta\Phi'(\lambda)}{\partial \lambda} = \lambda^3 + \frac{AC}{2} \lambda^2 - (1+BC)\lambda + \frac{AC}{2}, \quad (4.38)$$

$$\text{onde: } A = \sqrt{N^2 - |\delta|^2},$$

$$B = N,$$

$$C = \frac{\rho_x^2}{\sigma_n^2}.$$

Um resultado conhecido é que, dado um polinômio de 3º grau

$$x^3 + ax^2 + bx + c$$

e definindo:

$$p = \frac{a^2}{3} + b \quad q = 2\left(\frac{a}{3}\right)^3 - \frac{ab}{3} + c \quad D = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2$$

O polinômio de 3º grau tem uma raiz real e 2 raízes complexas conjugadas, três raízes reais onde pelo menos 2 são iguais, ou três raízes reais diferentes, se D é positivo, zero ou negativo, respectivamente.

Como já mencionado anteriormente estamos interessados nas raízes reais dentro do intervalo (0,1) como candidatas à minimização de $\Delta\Phi'(\lambda)$.

A fim de se obter as raízes candidatas a minimização de $\Delta\Phi'(\lambda)$ dentro deste intervalo, um estudo do discriminante em (4.38) faz-se necessário.

A análise computacional das curvas que descrevem o comportamento do discriminante para diferentes valores de N , $|\delta|^2$, ρ_x^2 e σ_n^2 demonstram no entanto que o discriminante é instável para valores elevados de ρ_x^2 e valores crescentes de σ_n^2 , ou seja o comportamento das raízes do polinômio (4.38) é instável. Um exemplo deste resultado pode ser visto no gráfico abaixo, onde foi utilizada uma relação sinal ruído de 20dB e o eixo das abcissas corresponde à variação de ρ_x^2 no intervalo de [0,10], e D está representado no eixo das ordenadas.

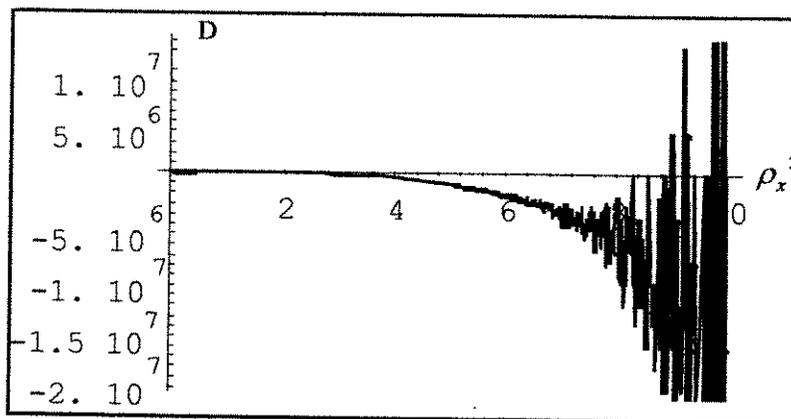


Figura – 4.1 Comportamento de D

Desta forma a determinação de “ λ ” que minimize $\Delta\Phi'(\lambda)$ se torna difícil computacionalmente.

A fim de se obter um resultado que conduza a um limitante talvez menos preciso que o fator de Chernoff para um “ λ ” ótimo, uma vez que este cálculo se mostrou de difícil realização, mas que descreva o comportamento do sistema, podemos escolher um valor de “ λ ” que conduza a uma expressão fechada para seu cálculo. Rescrevendo (4.36) onde

$$\phi = \frac{\rho_x^2 \sigma_n^2 \sqrt{N^2 - |\delta|^2}}{2}, \text{ temos}$$

$$\Delta\Phi'(\lambda) = \left[\frac{1}{1 - \rho_x^4 \sigma_n^4 (N^2 - |\delta|^2) \lambda^2} \right] \text{Exp} \left\{ \lambda \rho_x^4 (N^2 - |\delta|^2) \left[\frac{-1 + \rho_x^2 \sigma_n^2 N \lambda}{1 - \rho_x^4 \sigma_n^4 (N^2 - |\delta|^2) \lambda^2} \right] \right\} \quad (4.39)$$

Através da análise de (4.36), a fim de se manter $\Delta\Phi'(\lambda)$ positivo para todos os valores de “ ρ_x ” no cálculo da esperança sobre o ruído multiplicativo temos que

$$\lambda < \frac{1}{\rho_x^2 \sigma_n^2 \sqrt{N^2 - |\delta|^2}} \quad (4.40)$$

Desta forma fazendo,

$$\lambda = \frac{\alpha}{\rho_x^2 \sigma_n^2}, \quad (4.41)$$

onde α é uma constante a ser maximizada, o cálculo da esperança pode ser realizado e conseguimos garantir (4.40). Efetuando-se este cálculo, finalmente chegamos à expressão

$$E_\rho[\Delta\Phi(\lambda/\rho)] = \frac{1}{\left[1 + \frac{\gamma^2 (N^2 - |\delta|^2)}{4(N\gamma + 1)} \right]} \quad (4.42)$$

onde $\gamma = \frac{1}{\sigma_n^2}$ é a relação sinal ruído.

A expressão (4.42) coincide com o fator de Chernoff para o caso sem informação de estado de canal. Isto decorre do fato que durante o cálculo de (4.42), e utilizando (4.41), “ λ ” absorve o termo em “ ρ_x ” e desta forma os resultados com e sem CSI coincidem. Este fato ocorrerá sempre que tivermos variáveis de decisão onde o ruído multiplicativo puder ser absorvido por “ λ ”, ou seja, sempre que a variável de decisão $\Delta_{c/CSI} = f(\rho)\Delta_{s/CSI}$, como é o caso considerado.

Este resultado mostra que o fator de Chernoff para o caso com CSI pode ser limitado pelo fator de Chernoff sem CSI sempre que a relação entre as métricas com e sem CSI satisfizerem a relação acima citada.

4.2.2 - Aproximação do fator de Chernoff para uma seqüência

Para o caso de uma palavra código que corresponde a uma seqüência de grupos, através de (4.6) teríamos simplesmente o produtório de (4.42) para os n grupos que diferem entre as duas seqüências consideradas no fator de Chernoff.

No caso de duas seqüências porém existe uma restrição com relação ao resultado obtido. Como visto o resultado obtido em (4.41) foi realizado considerando-se um único grupo e fazendo-se $\lambda = f(\rho_x)$. Porém para o caso de uma seqüência de grupos, para cada grupo temos uma variável de ruído “ ρ_i ” que é estatisticamente independente dos demais grupos. Assim “ λ ” varia de grupo para grupo, o que contradiz a definição de fator de Chernoff estabelecida em (4.6).

Como a determinação do fator de Chernoff para duas seqüências utilizando um “ λ ” constante para todos os grupos não é alcançável, que interpretação podemos tirar do cálculo se utilizarmos um “ λ_i ” que varia de grupo para grupo ?

Partindo da definição de fator de Chernoff onde

$$\text{Prob}(\Delta \geq 0) \leq \mathbf{C}_\lambda (C \rightarrow \hat{C}) = \prod_{i \in n} E_{\rho_i} [E_{\Delta_i} [\text{Exp}(\lambda \Delta_i)]] , \text{ onde } \Delta = \sum_{i \in n} \Delta_i \quad (4.43)$$

e Δ_i 's são estatisticamente independentes, temos que

$$\text{Prob}(\Delta \geq 0) = \text{Prob} \left(\sum_{i \in n} \Delta_i \geq 0 \right) \quad (4.44)$$

Desta forma todas as combinações de Δ_i 's tal que a o somatório que define Δ seja maior que "0" constituem os termos que definem $\text{Prob}(\Delta \geq 0)$, ou seja podemos escrever

$$\text{Prob}(\Delta \geq 0) = \text{Prob}(\Delta_1 \geq 0, \Delta_2 \geq 0, \dots, \Delta_n \geq 0) + \text{termos} \quad (4.45)$$

onde em "termos" estão subentendidas todas as combinações de valores de Δ_i 's tal que $\Delta \geq 0$.

Devido à independência entre grupos podemos escrever para o primeiro termo em (4.45),

$$\text{Prob}(\Delta_1 \geq 0, \Delta_2 \geq 0, \dots, \Delta_n \geq 0) = \prod_{i \in n} P(\Delta_i \geq 0) \quad (4.46)$$

Além disso,

$$P(\Delta_i \geq 0) \leq E_{\rho_i} [\Delta \Phi(\lambda | \rho_i)], \quad (4.47)$$

Então o primeiro termo da probabilidade de erro pode ser limitado por

$$\text{Prob}(\Delta_1 \geq 0, \Delta_2 \geq 0, \dots, \Delta_n \geq 0) \leq \prod_{i \in n} E_{\rho_i} [\Delta \Phi(\lambda | \rho_i)] \quad (4.48)$$

Desta forma podemos dizer que a extensão de (4.42) para um par de seqüências que divergem em n grupos, pode ser interpretado como um limitante para o primeiro termo da probabilidade de erro.

Este resultado, embora constitua um limitante muito fraco para probabilidade de erro par a par entre seqüências, pode ser visto como uma ferramenta na investigação dos critérios a serem utilizados para construção de códigos, pois ele traduz ao menos em parte o comportamento do sistema considerado. Denotando então o resultado da extensão do cálculo de (4.42) para uma seqüência por LC (limitante de comportamento), podemos escrever para uma par de seqüências que divergem em n grupos:

$$LC = \prod_{i \in n} \frac{1}{\left[1 + \frac{\gamma^2 (N^2 - |\delta_i|^2)}{4(N\gamma + 1)} \right]} \quad (4.49)$$

É importante salientar que a comparação dos resultados obtidos poderiam nos levar a concluir que, uma vez que (4.49) é um limitante para a probabilidade de erro total para o caso sem CSI e um limitante apenas para o primeiro termo para o caso com CSI, o desempenho de sistemas utilizando CSI baseados na métrica (3.16) deveria ser pior que para o caso sem CSI. Esta comparação no entanto não seria justa, pois (4.49), que para o caso sem CSI consiste no fator de Chernoff exato, para o caso com CSI consiste de uma extensão do cálculo, que procura caracterizar em parte o comportamento do sistema. Sua utilidade está na extração de critérios para construção de códigos e não como limitante de probabilidade de erro, uma vez que uma série de aproximações e considerações foram utilizadas para que pudéssemos chegar a este resultado.

Capítulo 5

Construção de Códigos

Neste capítulo abordaremos o problema da construção de códigos de bloco para o sistema de modulação M-DPSK com uso de CSI para canais com desvanecimento Rayleigh.

Os códigos construídos e o método utilizado para a sua construção estão baseados nos resultados obtidos no capítulo anterior. Será mostrado que através do limitante de comportamento (LC), podemos extrair critérios que nos permitem estabelecer regras para a construção de códigos.

O método utilizado para a geração dos códigos é o de busca exaustiva, ou seja, uma vez estabelecidas as propriedades matemáticas desejáveis às seqüências que irão compor as palavras do código, é iniciado um processo de busca destas seqüências. Desta forma o código resultante constitui-se de um subconjunto de todas as seqüências possíveis para o comprimento e a cardinalidade da constelação PSK especificados. Uma vez tendo sido estabelecido este subconjunto de seqüências, que na realidade consiste de uma tabela de palavras que definem o código, o transmissor pode então mapear a informação oriunda da fonte nas palavras do código. Naturalmente, tendo o receptor o conhecimento desta tabela, este irá decidir qual a palavra mais provável de haver sido enviada baseando-se em um processo de decisão por máxima verossimilhança. (uso da métrica de máxima verossimilhança (3.16)).

5.1 - Análise do Limitante de Comportamento (LC)

No capítulo anterior foi mostrado que para sistemas de modulação M-DPSK com uso de CSI em canais de desvanecimento Rayleigh, o limitante LC é uma ferramenta matemática que estabelece uma relação entre a probabilidade de erro no processo de decisão entre duas seqüências em função de parâmetros matemáticos que relacionam estas seqüências.

Rescrevendo LC temos:

$$LC = \prod_{i \in n} \frac{1}{\left[1 + \frac{\gamma^2 (N^2 - |\delta_i|^2)}{4(N\gamma + 1)} \right]} \quad (5.1)$$

Para relações sinal ruído de interesse ($\gamma \gg 1$) LC pode ser aproximado assintoticamente por

$$LC = \left(\frac{4N}{\gamma} \right)^{L_n} \frac{1}{D\delta} \quad (5.2)$$

onde

$$D\delta = \prod_{i \in n} (N^2 - |\delta_i|^2) \quad (5.3)$$

e L_n é a cardinalidade do conjunto n .

O limitante de comportamento LC estabelece um relação em termos de desempenho entre duas seqüências. A fim de que possamos estender esta análise para o estudo de desempenho de um código, teríamos que computar as contribuições de todas as seqüências que compõem o código. Podemos considerar no entanto que assintoticamente o comportamento será dominado pelas seqüências que contribuem de maneira mais expressiva em LC ou seja aquelas produzem os maiores valores de LC dentre as seqüências que constituem o código. Desta forma definindo $\min(\|n\|)$ por L como o subconjunto de seqüências que apresenta a menor cardinalidade em termos de número de grupos, e o correspondente $D\delta$ por $D\delta(L)$ como os mínimos assumidos entre os pares de seqüências que constituem o código, e definindo α_L como o número médio de seqüências do código que produzem estes valores em LC, então podemos que dizer que assintoticamente o comportamento do código em termos de probabilidade de erro pode ser descrito aplicando-se o limitante de união. Denotaremos este resultado por LCC (limitante de comportamento de código).

Assim LCC pode ser escrito como:

$$LCC = \alpha_1 \left(\frac{4N}{\gamma} \right)^L \frac{1}{D\delta(L)} \quad (5.4)$$

Um estudo da expressão de LCC mostra que o desempenho do código é função da similaridade entre suas seqüências, ou seja, quanto mais similares são as seqüências que compõem o código, maior a probabilidade de erro no processo de decisão.

Da análise de LCC vemos que o primeiro termo da expressão mostra que quanto maior for o número de grupos (maior valor de L) para os quais as seqüências divergem, tanto melhor será o desempenho do código. Este termo pode ser visto como a distância de Hamming mínima do código se os grupos forem tomados como elemento de medida desta distância, ou seja, se cada grupo for visto como um “novo símbolo”. Note que o termo que se refere à distância de Hamming é função da relação sinal ruído, ou seja, sua contribuição para o desempenho do código está diretamente relacionada à relação sinal ruído utilizada. Desta forma constatamos que a distância de Hamming é um dos critérios importantes para busca de bons códigos, como se poderia esperar.

O termo $D\delta$ pode ser visto como um refinamento desta medida de distância entre seqüências, agora em termos de grupo. Dado que dois grupos divergem em algum ponto entre as seqüências, de quanto é esta medida de divergência.

Para dois grupos divergentes que divergem na i -ésima posição de seqüência, esta medida é dada por $N^2 - |\delta_i|^2$, sendo que para as seqüências em questão o resultado é dado pelo produto desta medida ao longo dos grupos divergentes ($D\delta$). Mais adiante será mostrado que quanto maior a dissimilaridade entre grupos e conseqüentemente entre seqüências, maior será este produto. Também, uma vez que o mesmo encontra-se no denominador da expressão (5.4) melhor será o desempenho do código. Note que $D\delta$, ao contrário da distância de Hamming, não é função da relação sinal ruído utilizada, sendo função apenas das características matemáticas das seqüências consideradas. O termo $D\delta$ pode ser visto então como uma nova medida de distância.

Através da análise de (5.4) podemos extrair alguns resultados que permitem uma avaliação preliminar da influência dos critérios relevantes ($L, D\delta(L)$) na construção de códigos.

Um estudo mais minucioso da expressão de LCC mostra que a influência da distância $D\delta(L)$ é inversamente proporcional à distância de Hamming L entre seqüências.

Considere um esquema com distância de Hamming L e $D\delta(L) = D\delta_1$. Suponha agora que o código é reprojeto de forma a fornecer $D\delta(L) = D\delta_2$ porém com mesmo valor de L . Assumindo que α_L permanece constante para os dois códigos, então o ganho assintótico devido um aumento de $D\delta(L)$ de um valor $D\delta_1$ para $D\delta_2$ para um mesmo valor de LCC pode ser escrito como segue

$$\Delta g_{\infty} \equiv RSR_1 - RSR_2 |_{LCC_1=LCC_2} = \frac{10}{L} \text{Log} \frac{D\delta_2}{D\delta_1} \quad (5.5)$$

onde RSR é a relação sinal ruído dada em termos da energia por símbolo. Esta relação também foi observada em [11] para esquemas PSK sem uso de CSI.

Suponha agora que $D\delta_2 = 2D\delta_1$. Para uma distância de Hamming $L=2$, o ganho assintótico prevê um ganho de codificação de 1.5dB, enquanto que o mesmo aumento em $D\delta(L)$ para uma distância de Hamming $L=10$ fornece um ganho de apenas 0.3dB.

Este resultado mostra que a maximização de $D\delta(L)$ é mais efetiva para códigos com baixos valores de L do que para códigos que apresentem elevados valores de L .

5.2 - Estudo do Comportamento de δ

A quantidade δ é definida como o produto interno entre o vetor de sinais enviado \underline{s} em um grupo de observação de comprimento N e o transposto conjugado do vetor de sinais estimado $\underline{\hat{s}}$ para o correspondente grupo, ou seja,

$$\delta = \underline{\hat{s}}^{T*} \cdot \underline{s} \quad (5.6)$$

Como visto anteriormente, a métrica de decisão para um grupo pode ser expressa como,

$$\begin{aligned} \eta &= \rho^2 \left| \sum_{i=0}^{N-1} r_{k-i} e^{j(\phi_{k-i} - \phi_{k-N+1})} \right|^2 \\ &= \rho^2 \left| r_{k-N+1} + r_{k-i} \sum_{i=0}^{N-2} e^{j \sum_{m=0}^{N-i-2} \Delta \phi_{k-i-m}} \right|^2 \end{aligned} \quad (5.7)$$

Através da análise desta expressão verificamos que a mesma independe do primeiro símbolo enviado no grupo de observação (devido ao cancelamento do mesmo), logo este pode sempre ser assumido como e^{j0} .

Então δ pode se expresso como

$$\delta = \left[e^{-j0}, e^{-j\hat{\phi}_{k-N+2}}, \dots, e^{-j\hat{\phi}_k} \right] \begin{bmatrix} e^{j0} \\ e^{j\hat{\phi}_{k-N+2}} \\ \vdots \\ e^{j\hat{\phi}_k} \end{bmatrix} \quad (5.8)$$

ou seja

$$\delta = 1 + \sum_{i=0}^{N-2} e^{j \sum_{m=0}^{N-i-2} (\Delta \phi_{k-i-m} - \Delta \phi_{k-i-m})} \quad (5.9)$$

denotando

$$\alpha_i = \sum_{m=0}^{N-i-2} (\Delta \phi_{k-i-m} - \Delta \phi_{k-i-m}) \quad (5.10)$$

Então, pode-se escrever

$$\delta = 1 + \sum_{i=0}^{N-2} e^{j\alpha_i} \quad (5.11)$$

Como as operações de soma e subtração entre as fases são realizadas sob módulo 2π , pode-se mostrar que os valores possíveis de α_i , são os próprios valores de fase utilizadas na constelação PSK.

É sabido da álgebra que o produto interno entre dois vetores, é uma medida de similaridade entre estes dois vetores. Sendo δ definido como o produto interno entre \underline{s} e $\underline{\hat{s}}$, este é uma medida da semelhança entre o vetor enviado e o vetor concorrente. Da mesma

forma $|\delta|^2$ é uma medida desta semelhança. Note-se que quanto maior a semelhança entre \underline{s} e $\underline{\hat{s}}$ maior será o valor de $|\delta|^2$.

A medida $N^2 - |\delta|^2$ possui uma relação inversa com a similaridade, ou seja, quanto mais similares são os vetores (maior o valor de $|\delta|^2$), menor será o valor de $N^2 - |\delta|^2$. É importante salientar que $N^2 - |\delta|^2$ é uma medida sempre positiva, atingindo o valor nulo quando se tem o maior grau de similaridade entre os vetores, isto é, quando $\underline{s} = \underline{\hat{s}}$ (neste caso $|\delta|^2 = N^2$ e $N^2 - |\delta|^2 = 0$).

Assim podemos dizer que $N^2 - |\delta|^2$ é uma medida da distância entre dois vetores.

Um caso interessante ocorre quando $N=2$, nesta situação a quantidade $N^2 - |\delta|^2$ corresponde ao módulo da distância Euclidiana ao quadrado entre os símbolos de informação enviados e a distância $D\delta$ se torna a distância produto.

Seja o sinal enviado $s_k = e^{j(\phi_{k-1} + \Delta\phi_k)}$ no intervalo $kT \leq t \leq (k+1)T$ e $\hat{s}_k = e^{j(\phi_{k-1} + \hat{\Delta}\phi_k)}$ o sinal concorrente correspondente, temos que,

$$\begin{aligned} \delta &= \left| 1 + e^{j(\Delta\phi_k - \hat{\Delta}\phi_k)} \right|^2 \\ &= 2 - 2 \operatorname{Re} \left\{ e^{j(\Delta\phi_k - \hat{\Delta}\phi_k)} \right\} \end{aligned} \quad (5.12)$$

Denotaremos agora o sinal enviado e o sinal concorrente correspondente por x e \hat{x} , respectivamente. O módulo da distância Euclidiana ao quadrado pode ser escrito como

$$|x - \hat{x}|^2 = |x|^2 - 2 \operatorname{Re} \{ x * \hat{x} \} + |\hat{x}|^2 \quad (5.13)$$

Assumindo que os sinais tem sua energia normalizada ($|x|^2 = |\hat{x}|^2 = 1$), para constelação PSK, temos então

$$\begin{aligned} |x - \hat{x}|^2 &= 2 - 2 \operatorname{Re} \left\{ e^{j(\Delta\phi_k - \hat{\Delta}\phi_k)} \right\} \\ &= N^2 - |\delta|^2 \end{aligned} \quad (5.14)$$

Assim para o cálculo da seqüência, temos que

$$D\delta = \prod_{i \in n} (N^2 - |\delta_i|^2) = \prod_{i \in n} |x_i - \hat{x}_i|^2 \quad (5.15)$$

Desta forma a distância produto é um caso especial da distância $D\delta$.

Este resultado mostra ainda que para o caso M-DPSK usual ($N=2$), os critérios de construção de códigos são os mesmos que os obtidos para sistemas MPSK com CSI e M-DPSK sem CSI, quando os critérios são obtidos através do fator do cálculo do fator de Chernoff [11].

Para uma situação geral onde $N > 2$ e com o uso de um alfabeto q-PSK, os valores que $D\delta$ pode assumir irão variar conforme o comprimento N dos grupos de observação utilizados, do comprimento das seqüências do código e a cardinalidade do alfabeto considerado.

No estudo realizado, não foi possível encontrar uma metodologia ou ainda identificar uma estrutura matemática que possibilitasse determinar os valores possíveis de serem assumidos por $D\delta$, bem como a freqüência com que estes valores ocorrem para uma situação específica de comprimento de grupo, seqüência e constelação PSK utilizada.

Baseado ainda no limitante LCC podemos observar que a medida que o valor de N (comprimento do grupo de observação) aumenta, a distância de Hamming alcançável se reduz de L para o caso onde $N=2$ para $L/(N-1)$ para um comprimento $N > 2$. Este fato demonstra uma perda de desempenho para situações onde a distância de Hamming é o fator predominante para o controle da probabilidade de erro.

Ainda baseado em LCC, observamos que o valor N aparece no numerador da expressão que define LCC elevado ao valor de L , o que nos leva a concluir que um aumento do valor de N é responsável por uma contribuição negativa em termos de desempenho, ou seja, a medida que utilizamos mais símbolos para formar os grupos de observação, cai o desempenho do código.

Então podemos nos questionar: Quanto se perde ou se ganha ao aumentar-se o número de símbolos por grupo? Existe uma condição onde a distância $D\delta$ pode compensar as perdas em termos de distância de Hamming e devidas ao numerador N em LCC, onde se torne vantajoso utilizar $N > 2$?

Para que pudessemos responder a estas perguntas teríamos que conhecer como $D\delta$ se comporta para diferentes valores de comprimento de grupo, seqüência e constelação

utilizados. Como dito anteriormente esta análise não foi possível, desta forma tentaremos responder a algumas destas questões através de resultados de simulação como será visto no próximo capítulo.

Concluimos através da análise realizada, que para relações sinal ruído elevadas (como geralmente será o caso em situações práticas), a maximização da distância de Hamming L e da distância $D\delta$ são os critérios mais importantes para a construção de códigos de bloco. É interessante notar que estes resultados não se restringem apenas à construção de códigos de bloco, sendo que poderiam ser igualmente utilizados para construção de códigos convolucionais ou de treliça, por exemplo.

5.3 - Algoritmo para Construção de Códigos

O algoritmo desenvolvido para a construção de códigos de bloco é baseado no processo de busca exaustiva, os critérios de busca são distância de Hamming e distância $D\delta$ entre seqüências.

Primeiramente é definido o tamanho do grupo de observação N a ser utilizado, o número de grupos por seqüência “ K ” e a constelação PSK a ser utilizada. Desta forma as palavras do código possuem comprimento $K(N-1)$. Isto se deve ao fato da utilização da sobreposição entre grupos.

A palavra toda nula é sempre considerada como pertencente ao código a ser gerado. Desta forma todos os códigos gerados possuem inicialmente a palavra toda nula ($[e^{j0}, e^{j0}, \dots, e^{j0}]$).

O processo de busca inicia-se utilizando o maior valor de distância de Hamming L e de distância $D\delta$ alcançáveis para os comprimentos e constelação especificados.

Inicia-se então o processo de geração de todas as seqüências possíveis. À medida que as seqüências são geradas, elas são comparadas às seqüências que compõem o código segundo os critérios de distâncias L e $D\delta$ especificados. Caso a seqüência comparada satisfaça os critérios, ela é então incorporada ao código, caso contrário a seqüência é eliminada. Uma nova seqüência é gerada e o processo de comparação se repete.

Uma vez esgotadas todas as seqüências, o código gerado pode ser armazenado. Os valores de L e $D\delta$ são ajustados para os próximos valores abaixo dos anteriores e o processo

de busca reinicia. Relembramos aqui que para um mesmo valor de L geralmente existe uma faixa de valores possíveis de $D\delta$. Assim em muitas situações somente o valor de $D\delta$ é ajustado.

O algoritmo pode ainda ser modificado de forma a atingir uma taxa de codificação especificada, assim a cada vez que se tem a varredura das seqüências para um dado valor de L e $D\delta$, ao final do processo é verificado se a taxa foi atingida, desta forma podemos encontrar os melhores códigos para uma dada taxa especificada. Para os casos onde a taxa não é possível de ser atingida, podemos obter os códigos que mais se aproximam das características desejadas.

Esquemáticamente o algoritmo pode se representado como se segue

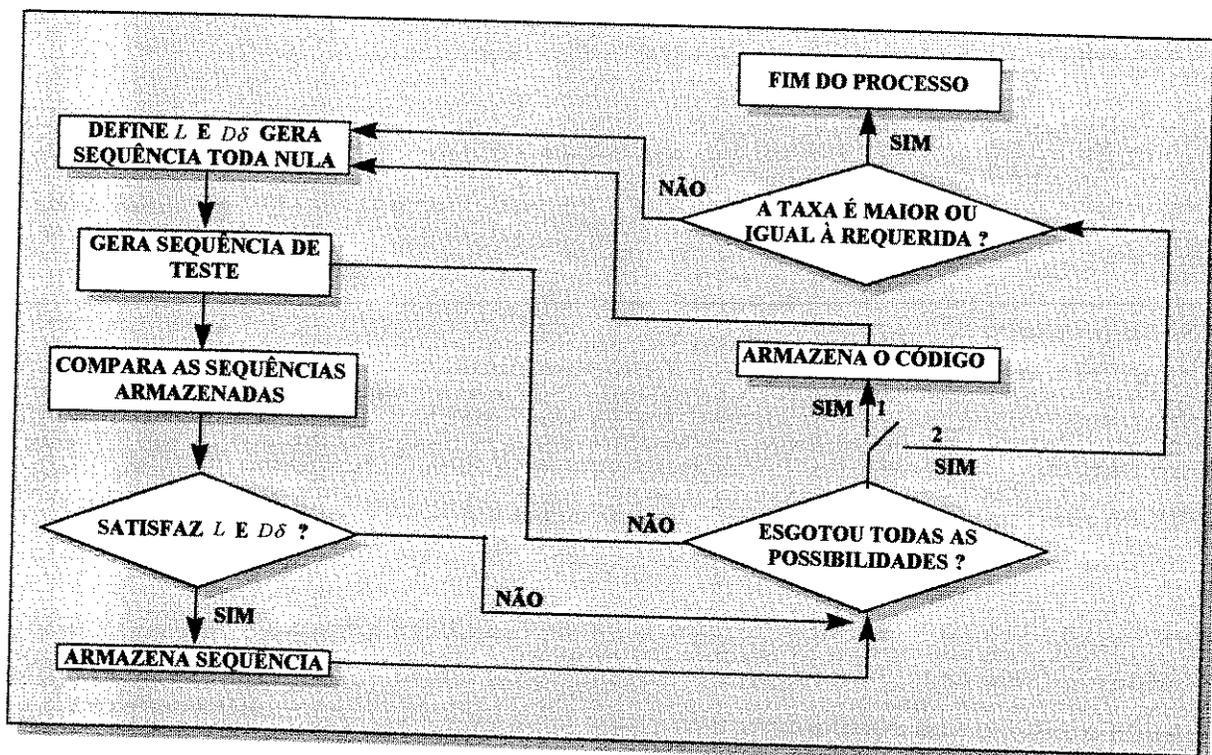


Figura 5.1 - Algoritmo de Construção de Códigos

Note-se que esquema acima estão mostradas as duas possibilidades de operação do algoritmo de busca de códigos.

Baseados no algoritmo descrito acima os programas implementados tem como objetivo a geração de bons códigos de bloco curtos.

Existe uma restrição computacional quando o comprimento dos códigos se torna muito longo. Devido ao fato dos programas utilizarem busca exaustiva para geração dos

códigos, temos que códigos de comprimento muito longo tornam-se proibitivos quando este tipo de metodologia é empregada. Seja por exemplo o caso onde $N=4$, $K=8$ e uma constelação 16-PSK. Nesta situação existem $7.922816251 \times 10^{28}$ seqüências de teste possíveis, o que torna proibitiva a utilização desta metodologia.

Foram gerados códigos para constelações 4-PSK e 8-PSK com comprimentos de grupos de observação $N=2$ e $N=3$, bem como número de grupos (K) variando deste 2 até 10 grupos em alguns casos.

5.4 - Códigos Gerados

As tabelas abaixo mostram os parâmetros dos melhores códigos obtidos para os casos estudados. A Tabela 1 mostra os códigos obtidos com o uso da constelação 4-PSK e $N=2$. A Tabela 2 mostra os códigos obtidos com o uso da constelação 8-PSK e $N=2$. A Tabela 3 mostra os códigos obtidos para constelação 4-PSK e $N=3$. A taxa R mostrada nas tabelas é dada em bits/símbolo de modulação.

K	L	$D\delta$	R
2	2	4	1,00
3	2	8	1,00
4	3	16	1,00
5	3	16	1,00
6	4	16	1,00
7	3	16	1,14
8	4	16	1,00
10	5	32	1,00

Tabela 1 - Códigos de Modulação 4-DPSK, $N=2$

K	L	$D\delta$	R
2	2	1,50	1,50
3	3	0,68	1,00
3	2	4,00	1,33
4	2	4,00	1,50
5	4	0,80	1,04
5	3	1,17	1,29
6	5	0,47	0,89
7	4	0,11	1,00

Tabela 2 - Códigos de Modulação 8-DPSK, $N=2$

k	L	$D\delta$	R
2	2	16	1,00
3	2	64	1,00
4	3	128	0,94
4	2	64	1,25

Tabela 3 - Códigos de Modulação 4-DPSK, $N=3$

Além dos códigos apresentados nas tabelas acima, outros códigos foram gerados para fim de comparação de desempenho e serão apresentados no próximo capítulo, onde os resultados de simulação são analisados.

Um algoritmo similar de busca de códigos de bloco curtos para detecção coerente foi descrito em [13]. Como o algoritmo aqui mostrado permite liberdade de variação de $D\delta$, foi possível determinar alguns códigos melhores que os encontrados em [13].

Os programas implementados para geração dos códigos foram escritos em linguagem C e uma listagem destes programas encontra-se em anexo ao final deste trabalho. Também é apresentada uma listagem dos códigos contidos nas tabelas acima.

Capítulo 6

Resultados de simulação

Neste capítulo serão apresentados os resultados de desempenho obtidos através de simulação para vários códigos construídos a partir da metodologia descrita no Capítulo 5. Nosso objetivo aqui é investigar a influência dos parâmetros de construção (Dh , $D\delta$ e N), no desempenho do sistema codificado, bem como verificar a concordância dos resultados de simulação com os resultados previstos pela teoria. Será analisada ainda a influência do uso de CSI e a utilização da métrica (3.16) para $l = 1$ e $l = 2$.

Os resultados apresentados ao longo do capítulo estão na forma de gráficos comparativos, onde dois ou mais esquemas de codificação são mostrados de maneira que se possa estabelecer uma relação entre o parâmetro comparado e o desempenho obtido.

As simulações foram realizadas utilizando-se embaralhamento perfeito entre grupos, embora na prática isto não seja totalmente possível, podemos dizer que os resultados de desempenho constituem-se em uma boa aproximação da realidade.

A relação sinal ruído é dada em termos da energia média por bit de informação, enquanto a probabilidade de erro é dada em termos de erro de palavra. Desta forma, se por um lado estamos sendo otimistas ao considerar um embaralhamento perfeito entre grupos, por outro estamos sendo pessimistas quanto à ocorrência de erros, uma vez que a probabilidade de erro de palavra se constitui em um limitante superior para a probabilidade de erro de bit.

O programa de simulação foi escrito em linguagem C e uma listagem do programa encontra-se em anexo ao final deste trabalho.

6.1 - Métrica de Máxima Verossimilhança

Como mostrado no Capítulo 4, a métrica de máxima verossimilhança é aproximada por,

$$\eta = \sum_i \rho_i \left| \mathbf{r}_i^T \underline{\mathbf{s}}_i^* \right|^l \quad (6.1)$$

onde $l=2$ para baixa RSR e $l=1$ para alta RSR. Os resultados de simulação comprovam este resultado. Porém para todos os códigos testados os resultados mostraram que a faixa de valores para os quais a métrica para $l=2$ apresenta ganhos quando a informação de estado de canal é utilizada está em uma região onde não há ganhos de codificação ($RSR \leq 5$ dB).

O gráfico abaixo mostra os resultados de simulação para dois sistemas codificados onde estão sendo comparados os desempenhos com e sem uso de CSI para a métrica $l=2$. É mostrado também o comportamento de um esquema D-BPSK não codificado para fim de comparação com os esquemas codificados.

Note-se que o ganho para o uso de CSI é quase desprezível nesta região, sendo que fora desta região observou-se que não existe ganho, e até uma pequena degradação quando a CSI é utilizada. A influência do uso de CSI parece estar associada à relação sinal-ruído utilizada, quanto mais ruidoso é o canal mais eficaz é a sua utilização.

Com base nestes resultados, a métrica para altas RSR mostra-se como a melhor alternativa, e será utilizada como métrica de decisão nos demais resultados a serem apresentados ao longo deste capítulo.

Os resultados que serão apresentados neste capítulo se referem a códigos construídos para modulação 4-PSK e com taxa de codificação de 1bit/símbolo. Esta escolha se deve ao fato de que para esta constelação, foram encontrados códigos com taxas inteiras tornando fácil a comparação em termos de desempenho com esquemas não codificados. Desta forma, para alguns dos gráficos apresentados é mostrado o desempenho comparativo de um esquema D-BPSK não codificado, que apresenta mesma taxa de codificação (1 bit/símbolo), para fim de comparação de desempenho com os esquemas codificados.

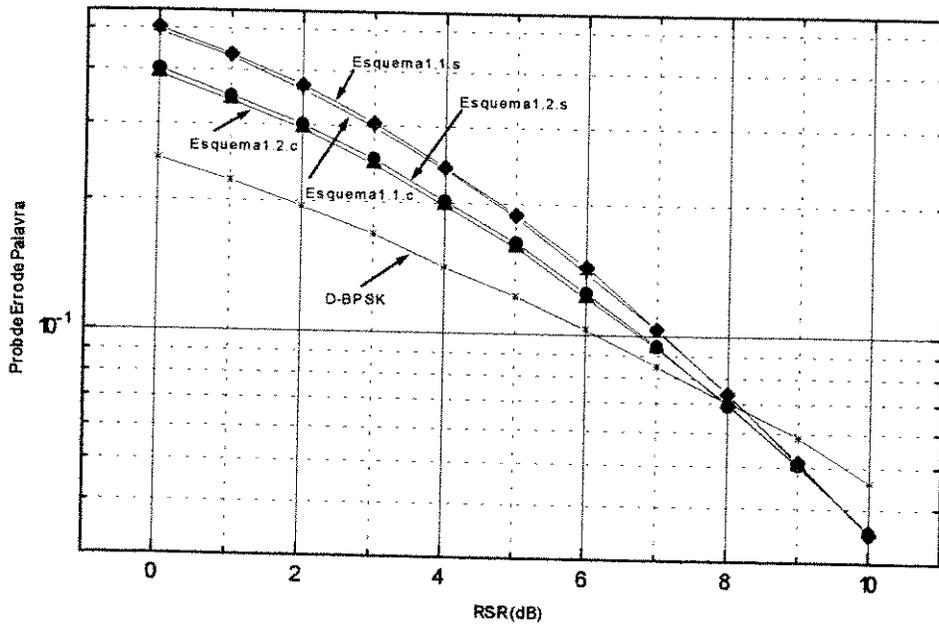


Figura 6.1 - Métrica para baixas RSR

Os esquemas codificados na figura se referem aos seguintes códigos:

Esquema 1.1.c/s: Modulação 4-PSK, $Dh=2$, $D\delta=4$, $N=2$, $K=2$, $R=1$ bit/símbolo;

Esquema 1.2.c/s: Modulação 4-PSK, $Dh=2$, $D\delta=8$, $N=2$, $K=3$, $R=1$ bit/símbolo;

Considerando-se as implementações com e sem CSI respectivamente.

6.2 - Influência de $D\delta$

Aqui é mostrada a influência de $D\delta$ para dois códigos que apresentam os mesmos parâmetros, porém com $D\delta_2 = 2D\delta_1$. Nesta situação a teoria prevê um ganho assintótico de aproximadamente 1.5 dB (expressão 5.5), observamos no entanto que para estes dois códigos o ganho chega a 2 dB. Este resultado pode ser explicado pelo fato de estarmos utilizando a métrica para altas RSR, e o cálculo do limitante de comportamento ter sido desenvolvido utilizando-se a métrica para baixas RSR. Isto mostra no entanto que os parâmetros apontados como de importância para construção de códigos, são ainda mais determinantes quando a métrica para altas RSR é utilizada.

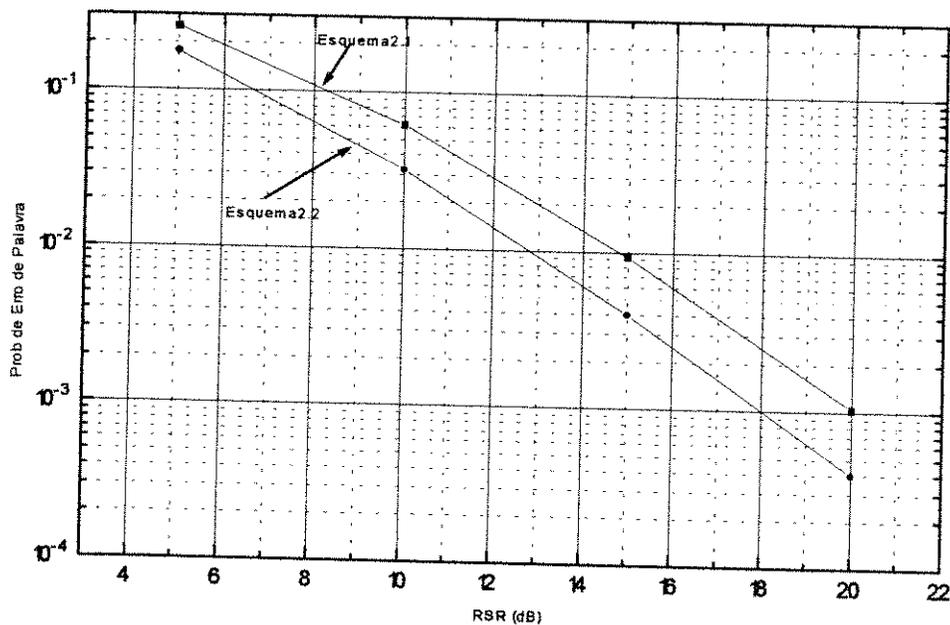


Figura 6.2 - Influência de $D\delta$

Os esquemas codificados na figura se referem aos seguintes códigos

Esquema 3.1: Modulação 4-PSK, $Dh=2$, $D\delta=4$, $N=2$, $K=3$, $R=1$ bit/símbolo;

Esquema 3.2: Modulação 4-PSK, $Dh=2$, $D\delta=8$, $N=2$, $K=3$, $R=1$ bit/símbolo;

ambos com uso de CSI.

6.3 - Influência de Dh

O gráfico abaixo mostra a influência da distância de Hamming entre dois esquemas codificados que possuem os mesmos parâmetros mas onde a distância de Hamming difere de uma unidade. Conforme previsto pela teoria vemos que o ganho é função da RSR.

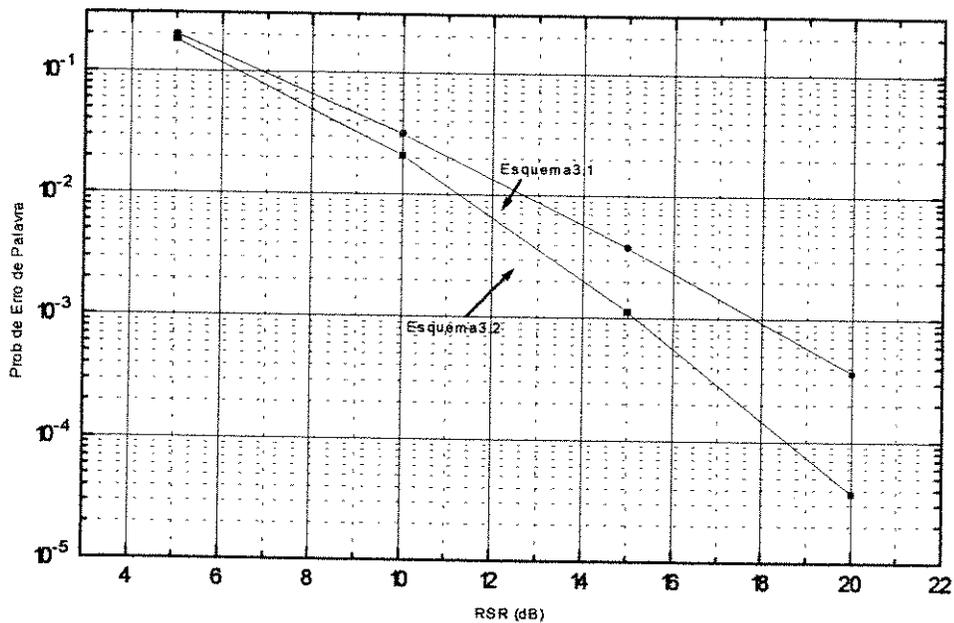


Figura 6.3 - Influência de Dh

Os esquemas codificados na figura se referem aos seguintes códigos:

Esquema 3.1: Modulação 4-PSK, $Dh=2$, $D\delta=16$, $N=2$, $K=4$, $R=1$ bit/símbolo;

Esquema 3.2: Modulação 4-PSK, $Dh=3$, $D\delta=16$, $N=2$, $K=4$, $R=1$ bit/símbolo;

ambos com uso de CSI.

6.4 - Influência de N

Como dito anteriormente, não foi possível determinar de maneira teórica a influência do aumento do número de símbolos por grupo na construção de códigos. Buscou-se então comparar códigos que apresentassem os mesmos parâmetros, porém com valores distintos de N como uma forma de verificar a influência de N na construção de códigos. Os resultados mostraram que para os códigos testados a utilização de $N=3$ acarreta em uma perda de desempenho em relação a códigos com $N=2$ quando códigos com mesmos parâmetros são testados.

Um exemplo para esta situação pode ser visto através da figura abaixo.

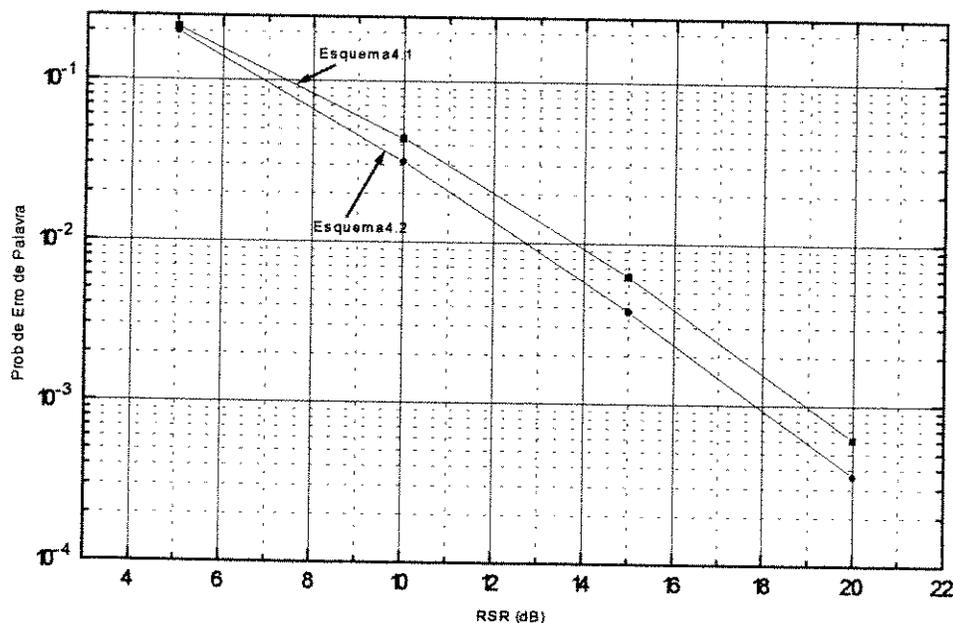


Figura 6.4 - Influência de N

Os esquemas codificados na figura se referem aos seguintes códigos:

Esquema 4.1: Modulação 4-PSK, $Dh=2$, $D\delta=16$, $N=3$, $K=4$, $R=1$ bit/símbolo;

Esquema 4.2: Modulação 4-PSK, $Dh=2$, $D\delta=16$, $N=2$, $K=4$, $R=1$ bit/símbolo;

ambos com uso de CSI.

6.5 - Influência do uso de CSI

O gráfico abaixo mostra a influência do uso de CSI quando a métrica para altas RSR é utilizada. Observamos que a sua influência é tanto maior quanto maior é o comprimento do código utilizado. Isto pode ser explicado pelo fato de que, quanto mais grupos entram na métrica de decisão, maior é a contribuição da CSI no cômputo geral da métrica.

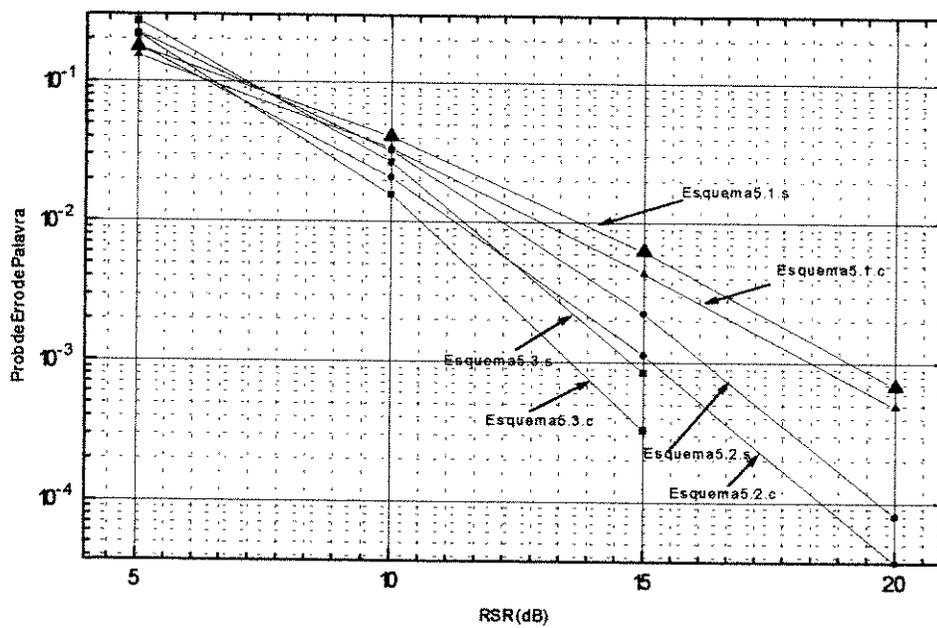


Figura 6.5 - Influência do Uso de CSI

Os esquemas codificados na figura se referem aos seguintes códigos:

Esquema 5.1.c/s: Modulação 4-PSK, $Dh=2$, $D\delta=4$, $N=2$, $K=2$, $R=1$ bit/símbolo;
 Esquema 5.2.c/s: Modulação 4-PSK, $Dh=3$, $D\delta=16$, $N=2$, $K=4$, $R=1$ bit/símbolo;
 Esquema 5.3.c/s: Modulação 4-PSK, $Dh=4$, $D\delta=16$, $N=2$, $K=6$, $R=1$ bit/símbolo;
 com e sem CSI respectivamente.

6.6 - Melhores Códigos 1

Apresentamos aqui os resultados de simulação para os melhores códigos obtidos para constelação 4-PSK com uso de CSI. Na Figura 6.6 é mostrado também a curva do sistema D-BPSK não codificado para fim de comparação com os esquemas codificados.

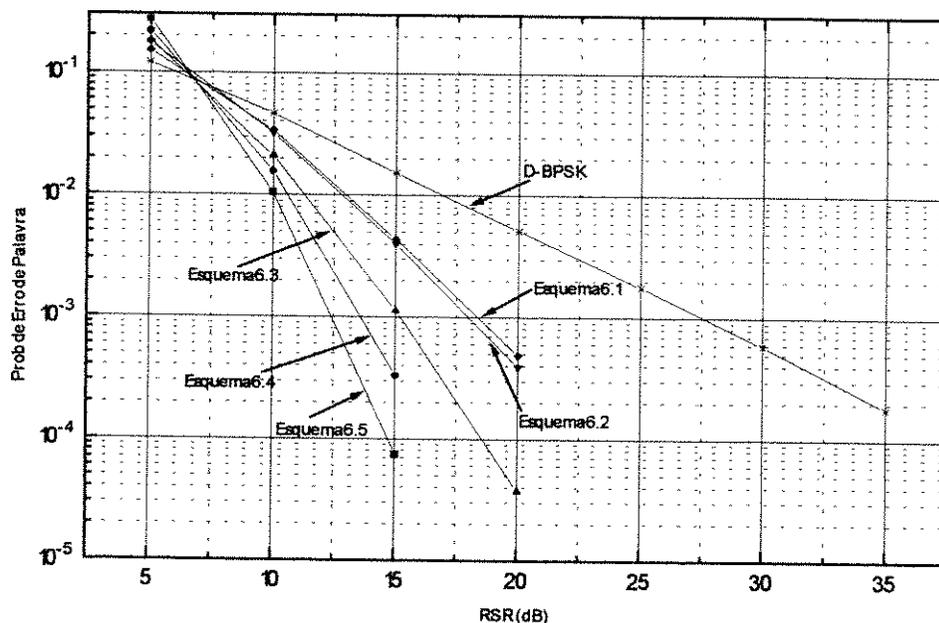


Figura 6.6 - Melhores Códigos 1

Os esquemas codificados na figura se referem aos seguintes códigos:

- Esquema 6.1: Modulação 4-PSK, $Dh=2$, $D\delta=4$, $N=2$, $K=2$, $R=1$ bit/símbolo;
- Esquema 6.2: Modulação 4-PSK, $Dh=2$, $D\delta=8$, $N=2$, $K=3$, $R=1$ bit/símbolo;
- Esquema 6.3: Modulação 4-PSK, $Dh=3$, $D\delta=16$, $N=2$, $K=4$, $R=1$ bit/símbolo;
- Esquema 6.4: Modulação 4-PSK, $Dh=4$, $D\delta=16$, $N=2$, $K=6$, $R=1$ bit/símbolo;
- Esquema 6.5: Modulação 4-PSK, $Dh=5$, $D\delta=32$, $N=2$, $K=10$, $R=1$ bit/símbolo;

6.7 -Melhores Códigos 2

No item anterior a comparação entre um esquema D-BPSK não codificado e o desempenho dos códigos obtidos não é justa uma vez que estamos comparando probabilidade de erro de palavra (dos esquemas codificados), com probabilidade de erro de bit para o esquema não codificado.

Neste item os resultados anteriores da Figura 6.6, para os esquemas codificados são mantidos. Porém a comparação agora é feita em termos de probabilidade de erro de palavra para os esquemas não codificados, através do uso da fórmula $P_{EP} = 1 - (1 - P_{EB})^K$, onde P_{EP} é a probabilidade de erro de palavra equivalente, P_{EB} a probabilidade de erro de bit para um esquema D-BPSK não codificado e K denota o número de bits da palavra código do código a ser comparado. Assim observa-se estabelecer o desempenho equivalente de um esquema não codificado em termos de erro de palavra.

No gráfico abaixo usamos a notação P.E.P.E.N.C.x para denotar probabilidade de erro de palavra equivalente de um esquema não codificado, sendo 'x' o comprimento da palavra código equivalente. Para os códigos comparados na Figura 6.7, os Esquemas 6.1, 6.2, 6.3, 6.4, e 6.5 apresentam comprimentos de palavra iguais a 2, 3, 4, 6, e 10, respectivamente.

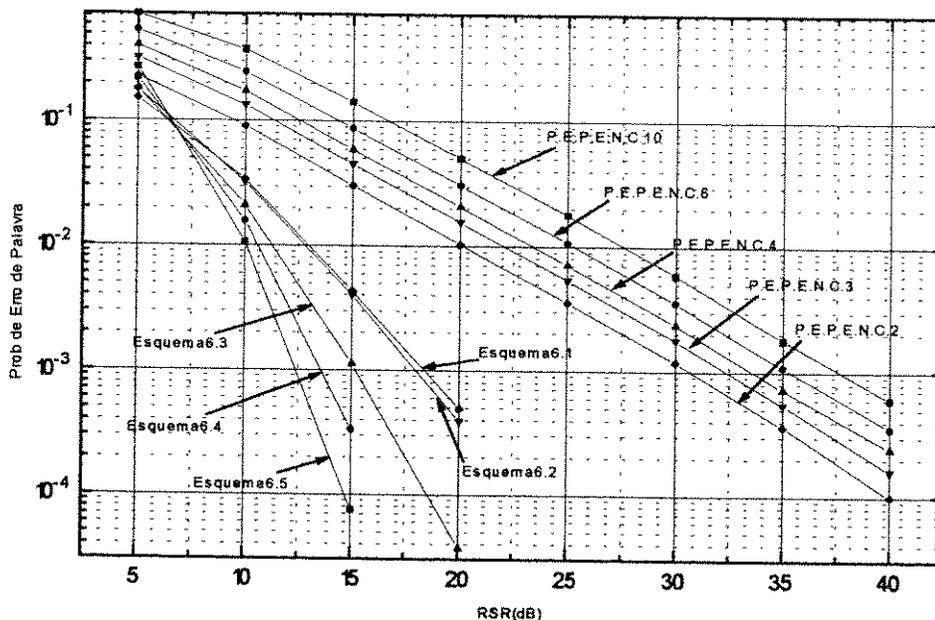


Figura 6.7 - Melhores códigos 2

É importante salientar que os exemplos apresentados neste capítulo nem sempre se constituíram nos melhores códigos para os parâmetros especificados, mas foram escolhidos de forma a ilustrar as características e a influência do uso destes parâmetros em diferentes esquemas de codificação. Também, o comportamento mostrado nestes exemplos se verificou para outros códigos simulados, correspondendo a uma tendência geral de comportamento e não a casos isolados.

Capítulo 7

Conclusão

Neste trabalho foi desenvolvido o cálculo do fator de Chernoff condicionado a um valor de amplitude de desvanecimento (ρ_i) para o modelo de canal considerado. Devido as restrições impostas pelo cálculo de (4.36), a obtenção de uma expressão fechada do fator de Chernoff para presença de CSI não foi possível. A partir de uma nova interpretação deste cálculo determinaram-se critérios de projeto para construção de códigos.

É importante salientar que a expressão final obtida, (4.49), coincide com o fator de Chernoff para o caso sem CSI [Ventura]. Esta coincidência não é gratuita. É fácil mostrar que sempre que na métrica de decisão, o fator correspondente ao desvanecimento puder ser incorporado na constante λ , os resultados com e sem CSI coincidirão, e os critérios de construção de bons códigos serão os mesmos para os dois casos. Os resultados de simulação confirmaram esta suposição. Entretanto os resultados de desempenho superaram o ganho assintótico previsto pela expressão (4.49). Isto se deve em parte ao fato de cálculo ter sido desenvolvido utilizando-se a métrica para baixas RSR, e as simulações terem sido na sua maioria realizadas utilizando a métrica para altas RSR.. Entretanto este resultado também mostra que o comportamento assintótico não é totalmente previsto por LC, sugerindo que a influência dos parâmetros D_H e D_δ é ainda maior que a prevista.

A utilização da métrica para altas RSR ((3.16) $\Rightarrow l = 1$) se deve ao fato de que os resultados de simulação mostraram que nas regiões de interesse, isto é, onde há ganhos observáveis de codificação, a métrica para altas RSR se apresenta como a melhor opção. Como a métrica para baixas RSR ($l = 2$) coincide com a métrica gaussiana, fica assim comprovado que nem sempre ela implicará em ganhos relativos à um sistema sem CSI.

A extensão do intervalo de observação para $N > 2$, mostrou que para os casos estudados, há perda de desempenho quando códigos que possuem os mesmos parâmetros são comparados como os seus correspondentes utilizando $N = 2$. Esta perda pode ser explicada pelo fato de que ao se agrupar símbolos em intervalos de observação de comprimento $N > 2$,

a distância de Hamming alcançável cai de um fator de L para $L/(N-1)$. Em contra partida, na detecção multi-simbólica, os valores alcançáveis de D_s podem aumentar significativamente. Sendo assim, é difícil prever para que taxas existem códigos que apresentem um ganho na utilização de detecção multi-simbólica. O efeito do aumento de N poderia ser obtido através do cálculo da taxa de corte do modelo de canal considerado (ou através de um estudo mais detalhado do comportamento de D_s). Estes estudos permanecem em aberto.

Embora não tenha sido realizado um estudo mais detalhado da estrutura dos códigos gerados, alguns códigos parecem apresentar uma estrutura possível de ser organizada através de uma treliça, o que pode reduzir grandemente a complexidade e o tempo gasto no processo de decodificação.

Anexo A

Listagem dos Programas de geração de códigos e do programa de simulação

```
//gera os melhores codigos para constelação 4-PSK,N=2

#include <alloc.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define CARD 4
#define SEQCOMP 10 //comprimento da sequência (em grupos)

FILE *ptr;
void main(void){

int i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i,k,l,m,x=1,a,A[SEQCOMP],
    **criterio,limiarH,
    seqcompH,seqcompD,num,resul[4],
    vH=SEQCOMP+1,temp,re=0,im=0,H;
float vD=262150.,dprod=1.,limiarD;
num=(int)pow(2.,(double)SEQCOMP);
seqcompH=SEQCOMP;
seqcompD=SEQCOMP*2;
clrscr();
do{ if(x>1)
    {
        for(i=0;i<x;i++)
            farfree(criterio[i]);
        farfree(criterio);
    }
    x=1;
    limiarH=seqcompH;
    limiarD=(float)pow(2.,(double)seqcompD);

    criterio=(int**)farmalloc(sizeof(*criterio));
    if(criterio==NULL){printf("\n nao consegui alocar CRITERIO");
        exit(1);}
    criterio[0]=(int*)farmalloc(SEQCOMP*sizeof(int));
    if(criterio[0]==NULL){printf("\n nao consegui alocar CRITERIO[0]");
        exit(1);}

    for(m=0;m<SEQCOMP;m++)
        criterio[0][m]=0;
    printf("%d,%f\n",limiarH,limiarD);
for(i1=0;i1<CARD;i1++){ //alterar o numero de lacos de acordo com
for(i2=0;i2<CARD;i2++){ //o numero total de simbolos
for(i3=0;i3<CARD;i3++){
    for(i4=0;i4<CARD;i4++){
    for(i5=0;i5<CARD;i5++){
    for(i6=0;i6<CARD;i6++){
    for(i7=0;i7<CARD;i7++){
    for(i8=0;i8<CARD;i8++){
    for(i9=0;i9<CARD;i9++){
    for(i9=0;i9<CARD;i10++){

        A[0]=i1;A[1]=i2;A[2]=i3; //alterar de acordo com o número de simbolos
        A[3]=i4;A[4]=i5;A[5]=i6;
        A[6]=i7;A[7]=i8;A[8]=i9;A[9]=i10;
        for(i=0;i<x;i++){
            H=0;
            for(k=0;k<SEQCOMP;k++){

                a=criterio[i][k]-A[k];
                if(a!=0) H+=1;
```

```

        a%=CARD;
        if(a<0) a=CARD+a;
        switch(a){
        case 0:re+=1;break;
        case 1:im+=1;break;
        case 2:re-=1;break;
        case 3:im-=1;break;}

        temp=(float)(4.-((1+re)*(1+re)+im*im));
        if(temp==0) temp=1;
        dprod*=(float)temp;
        re=0;
        im=0;
    }
    if(H<vH)
    vH=H;

    if(dprod<vD)
        vD=dprod;
    dprod=1;
}

if(vD>=limiarD && vH>=limiarH)
{
    critério=(int**)farrealloc(critério,(unsigned long)(x+1)*sizeof(*critério));
        if(critério==NULL){printf("\n nao consegui alocar CRITERIO");
        exit(1);}
        critério[x]=(int*)farmalloc(SEQCOMP*sizeof(int));
        if(critério[x]==NULL){printf("\n nao consegui alocar CRITERIO[%d]", x);
        exit(1);}
    for(m=0;m<SEQCOMP;m++)
    critério[x][m]=A[m];
    x++;}
vD=66000.,vH=SEQCOMP+1;
} } } } } } //alterar de acordo com o numero de laços
seqcompD=1;
if(seqcompD<seqcompH)
{seqcompH-=1;
seqcompD=2*seqcompH;
}
printf("%d\n",x);
} while( x < num );

for(i=0;i<x;i++){
for(k=0;k<SEQCOMP;k++)
printf("%d ",critério[i][k]);
printf(" ");
}
printf("\n");
printf("distancia produto=%f\n",limiarD);
printf("distancia de Hamming=%d\n",limiarH);
printf("num de seq encontradas=%d \n",x);

/*#####*/
//guarda o codigo mapeado no arquivo CODIGO.M
/* ptr=fopen("c:\cristhof\COD4N2.M","aw+t");
fprintf(ptr,"\n\n");
fprintf(ptr,"distancia produto=%f\n",limiarD);
fprintf(ptr,"distancia de Hamming=%d\n",limiarH);
fprintf(ptr,"comprimento do bloco %d\n",SEQCOMP);
fprintf(ptr,"numero de sequencias encontradas=%d\n",x-1);
for(i=0;i<x;i++){ if(!(i%3))
        fprintf(ptr,"\n");
        fprintf(ptr,"[");
        for(k=0;k<SEQCOMP;k++)
        fprintf(ptr,"%d",critério[i][k]);
        fprintf(ptr,"]");
        fprintf(ptr," ");}

fclose(ptr);
/*#####*/
for(i=0;i<x;i++)
    farfree(critério[i]);
    farfree(critério);

```

}

Gera os melhores códigos para constelação 8-PSK, N=2.

```

#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <alloc.h>
#define CARD 8
#define SEQCOMP 7 //comprimento da sequencia (em grupos)
#define DT 4 //numero de valores distintos que (N^2-|d|^2) assume
#define DH 5 //define a distancia de Hamming

FILE *ptr;
void main(void){

int w,i1,i2,i3,i4,i5,i6,i7,
    i,k,l,m,x,a,A[SEQCOMP],
    **criterio,limiarH,num,
    vH=SEQCOMP+1,H;
double vD=66000.,dprod=1.,temp,limiarD,re=0.,im=0.;
double a0 = 0., a1= 1.,a2=sqrt(2.)/2.;
num=(int)pow(4.,(double)SEQCOMP);
limiarH=DH;
/*#####*/
//rotina/calcula todos os valores de Dd para uma dada distância de Hamming

double *DD,acumula
,buf,
c0,c1,c2,c3; //ei's-->valores de (n^2-|d|^2)
c0=4.;
c1=2.+ sqrt(2.);
c2=2.;
c3=2.- sqrt(2.);

DD=(double*)malloc(sizeof(double));DD[0]=0.;
int l1,l2,l3,l4,l5,l6,u,B[DH],b,xd=1,seta,t,j;
//variar o numero de lacos de acordo com a distancia de hamming
for(l1=0;l1<DT;l1++){
for(l2=0;l2<DT;l2++){
for(l3=0;l3<DT;l3++){
for(l4=0;l4<DT;l4++){
for(l5=0;l5<DT;l5++){
// for(l6=0;l6<DT;l6++){

//variar o numero de B[i]'s de acordo com o numero de lacos
B[0]=l1;B[1]=l2;B[2]=l3;
B[3]=l4;B[4]=l5;//B[5]=l6;
acumula=1.;
for(t=0;t<DH;t++){
b=B[t];
switch(b){
case 0:acumula*=c0;break;
case 1:acumula*=c1;break;
case 2:acumula*=c2;break;
case 3:acumula*=c3;break;}
}

seta=0;
for(t=0;t<xd;t++)
{
if(acumula==DD[t])
seta=1;
}
if (seta==0){
DD=(double*)realloc(DD,(xd+1)*(sizeof(double)));
DD[xd]=acumula;
xd++;}
} } } } //alterar de acordo com
//o numero de laços
for(u=1;u<xd-1;u++){

```

```

        for(t=u+1;t<xd;t++)
            if (DD[u]<DD[t])
                {buf=DD[t];
                DD[t]=DD[u];
                DD[u]=buf;
                }
        for(t=1;t<xd;t++)
        printf("DD[%d]=%f ",t,DD[t]);
        printf("xd=%d",xd);
/*****
//rotina principal
clrscr();
x=1;
j=40 ;
do{ if(x>1)
    {
    for(i=0;i<x;i++)
    farfree(criterio[i]);
    farfree(criterio);
    }
    x=1;
    limiarD=DD[j];
    criterio=(int**)farmalloc(sizeof*criterio);
    if(criterio==NULL){printf("\n nao consegui alocar CRITERIO");
    exit(1);}
    criterio[0]=(int*)farmalloc(SEQCOMP*sizeof(int));
    if(criterio[0]==NULL){printf("\n nao consegui alocar CRITERIO[0]");
    exit(1);}
    for(m=0;m<SEQCOMP;m++)
    criterio[0][m]=0;
    printf("%d,%f,%d\n",limiarH,limiarD,j);
for(w=0;w<2;w++){
for(i1=w;i1<7+w;i1+=2){ //alterar o numero de lacos de acordo com
for(i2=w;i2<7+w;i2+=2){ //o numero total de simbolos=SEQCOMP
for(i3=w;i3<7+w;i3+=2){
for(i4=w;i4<7+w;i4+=2){
for(i5=w;i5<7+w;i5+=2){
for(i6=w;i6<7+w;i6+=2){
for(i7=w;i7<7+w;i7+=2){
A[0]=i1;A[1]=i2;A[2]=i3;
A[3]=i4;A[4]=i5;A[5]=i6;A[6]=i7; //alterar de acordo com o numero de simb

for(i=0;i<x;i++){
H=0;
for(k=0;k<SEQCOMP;k++){
a=criterio[i][k]-A[k];
if(a!=0) H+=1;
a%=CARD;
if(a<0) a=CARD+a;
switch(a){
case 0:re+=1.;break;
case 1:{re+=a2;im+=a2;};break;
case 2:im+=1.;break;
case 3:{re-=a2;im+=a2;};break;
case 4:re-=1.;break;
case 5:{re-=a2;im-=a2;};break;
case 6:im-=1.;break;
case 7:{re+=a2;im-=a2;};break;}
temp=(double)(4.-((1.+re)*(1.+re)+im*im));
if(temp==0.) temp=1;
dprod*=temp;
re=0.;
im=0.;
} //fim do laço k
if(H<vH)
vH=H;
if(dprod<vD)
vD=dprod;
dprod=1.;
} //fim do laço i
if(vD>=limiarD && vH>=limiarH)

```

```

    {   criterio=(int**)farrealloc(criterio,(unsigned long)(x+1)*sizeof(*criterio));
        if(criterio==NULL){printf("\n nao consegui alocar CRITERIO");
exit(1);}
        criterio[x]=(int*)farmalloc(SEQCOMP*sizeof(int));
        if(criterio[x]==NULL){printf("\n nao consegui alocar CRITERIO[%d]", x);
exit(1);}
        for(m=0;m<SEQCOMP;m++)
        criterio[x][m]=A[m];
        x++;
    }
    vD=66000.;vH=SEQCOMP+1;
    } } } } } //alterar de acordo com o numerode simbolos
                // de laços=SEQCOMP
        j++;
        printf("%d\n",x);
    }while( x < num && j < xd);
        free(DD);
    for(i=0;i<x;i++){
        for(k=0;k<SEQCOMP;k++)
            printf("%d ",criterio[i][k]);
            printf(" ");
        }
        printf("\n");
        printf("distancia produto=%f\n",limiarD);
        printf("distancia de Hamming=%d\n",limiarH);
        printf("num de seq encontradas=%d \n",x);
        printf("numero de valores de Dd encontrados=%d",xd-1);
/*#####*/
//guarda o codigo mapeado no arquivo CODIGO.M
/* ptr=fopen("c:\cristhof\codigo.M", "aw+t");
    fprintf(ptr, "\n\n");
    fprintf(ptr, "distancia produto=%f\n",limiarD);
    fprintf(ptr, "distancia de Hamming=%d\n",limiarH);
    fprintf(ptr, "numero de sequencias encontradas=%d\n",x-1);
    for(i=0;i<x;i++){
        fprintf(ptr, "[");
        for(k=0;k<COL;k++)
            fprintf(ptr, "%d",criterio[i][k]);
            fprintf(ptr, "];");
            fprintf(ptr, " ");}

fclose(ptr); */
/*#####*/
    for(i=0;i<x;i++)
        farfree(criterio[i]);
        farfree(criterio);
    }

```

Gera códigos para N=3 modulação 4-PSK

```

#include <process.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#define CARD 4
#define COMP 2 //numero de simbolos/bloco
#define SEQCOMP 4 //numero de blocos
#define COL COMP*SEQCOMP

FILE *ptr;
long xxx;
void main(void){
int resul[4];
int i1,i2,i3,i4,i5,i6,i7,i8,i,k,l,m,x=1,a,A[COL],
num=(int)pow(2.,(double)COL),
**criterio, //especificacao da distancia de Hamming
pal=4;//especifica#ao do num palavras por linha no arquivo de saida
int ih, Dh,Dih,DDih,temp,re=0,im=0;
float vD=16777216.,dprod=1.,limiarD;
double pot=7. //3.*(double)COL;
long int u;

```

```

clrscr();
do{limiarD=(float)pow(2.,pot );getch(); //especificacao de Dd
  if(x>1)
    {
      for(i=0;i<x;i++) farfree(criterio[i]);
      farfree(criterio);
    }
  x=1;
  Dh=COL;
  criterio=(int**)farmalloc(sizeof(*criterio));
  if(criterio==NULL) {
    printf("\n nao consegui alocar CRITERIO1");
    exit(1);
  }
  printf("\n%d",COL*sizeof(int));
  criterio[0]=(int*)farmalloc(COL*sizeof(int));
  if(criterio[0]==NULL) {
    printf("\n nao consegui alocar CRITERIO[0]");
    exit(1);
  }
  for(m=0;m<COL;m++) criterio[0][m]=0;
  xxx=0;
  for(i1=0;i1<CARD;i1++) {
    //alterar o numero de lacos de acordo com
    for(i2=0;i2<CARD;i2++){ //o numero total de simbolos
      for(i3=0;i3<CARD;i3++){
        for(i4=0;i4<CARD;i4++){
          for(i5=0;i5<CARD;i5++){
            for(i6=0;i6<CARD;i6++){
              for(i7=0;i7<CARD;i7++){
                for(i8=0;i8<CARD;i8++){
                  A[0]=i1;A[1]=i2;A[2]=i3; //alterar de acordo com o numero de simb
                  A[3]=i4;A[4]=i5;A[5]=i6;
                  A[6]=i7;A[7]=i8;
                  Dih=COL;ih=0;
                  for(i=0;i<x;i++){
                    DDih=0;
                    for(k=0;k<SEQCOMP;k++){
                      for(l=k*COMP;l<k*COMP+COMP;l++){
                        a=criterio[i][l]-A[l];
                        if(a!=0) ih=1;
                        a%=CARD;
                        if(a<0) a=CARD+a;
                        switch(a){
                          case 0:re+=1;break;
                          case 1:im+=1;break;
                          case 2:re-=1;break;
                          case 3:im-=1;break;
                        }
                      }//fim do laço l
                      temp=(float)((COMP+1)*(COMP+1)-((1+re)*(1+re)+im*im));
                      DDih+=ih;
                      ih=0;
                      if(temp==0) temp=1;
                      dprod*=(float)temp;
                      re=0;
                      im=0;
                    }//fim do laço k
                    if(DDih<Dih)
                      Dih=DDih;
                    if(dprod<vD) vD=dprod;
                    dprod=1.;
                  }//fim do laço i
                if(vD>=limiarD) {
                  criterio=(int**)farrealloc(criterio,(x+1)*sizeof(*criterio));
                  if(criterio==NULL){printf("\n nao consegui alocar CRITERIO2"); exit(1);}
                  criterio[x]=(int*)farmalloc(COL*sizeof(int));
                  if(criterio[x]==NULL){printf("\n nao consegui alocar CRITERIO[%d]2", x); exit(1);}
                  printf("\n%d .. %u",x,farcoreleft());
                  u=farcoreleft();
                  for(m=0;m<COL;m++) criterio[x][m]=A[m];
                  x++;getch();
                }
                if(Dih<Dh)

```

```

                Dh=Dih;
                }//fim do laço if(limiarD)
                vD=1677216.;
            } } } } } //alterar de acordo com o numero de laços
            printf("Dd=%f,Dh=%d\nx=%d\n",limiarD,Dh,x);
            pot=1.;
        } while(pot>1.&& x < num);

        for(i=0;i<x;i++) {
            for(k=0;k<COL;k++) printf("%d ",criterio[i][k]);
            printf(" ");
        }
        printf("\n");
        printf("distancia produto=%f\n",limiarD);
        printf("distancia de Haming=%d\n",Dh);
        printf("num de seq encontradas=%d \n",x);
        /*#####*/
        //guarda o codigo mapeado no arquivo CODIGON3.M
        ptr=fopen("c:\cristhof\codigoN3.M","aw+t");
        fprintf(ptr,"n\n");
        fprintf(ptr,"N=%d,K=%d,Taxa= ",COMP,SEQCOMP);
        fprintf(ptr,"Ddelta=%f\n",limiarD);
        fprintf(ptr,"distancia de Haming=%d\n",Dh);
        fprintf(ptr,"numero de sequencias encontradas=%d\n",x);
        for(i=0;i<x;i++){ if(!(i%pal))
            fprintf(ptr,"n");
            fprintf(ptr,"[");
            for(k=0;k<COL;k++)
                fprintf(ptr,"%d",criterio[i][k]);
            fprintf(ptr,"]");
            fprintf(ptr," ");}
        fclose(ptr);
        for(i=0;i<x;i++)
            free(criterio[i]);
        free(criterio);
    }

```

Apartir das especificacoes do codigo(Dd,L,N,DisHamming) gera o arquivo de entrada para simulac3 (programa de simulacão)
//programa para uso apenas 4-PSK

```

#include <alloc.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define CARD 4
#define COMP 1 //número de símbolos de inf no grupo
#define SEQCOMP 9 //número de grupos
#define COL COMP*SEQCOMP //número total de símbolos

FILE *ptr;
void main(void){
char i1,i2,i3,i4,i5,i6,i7,i8,i9;
int i,k,l,m,x=1,a;
char A[COL],**criterio ;
int re=0,im=0,temp,resul[4],vHL=COL,vH,ih,limiarH=4;//<- valor de Dh aplicado
float vD=262150.,dprod=1.,limiarD=64.;//<- valor de Dd aplicado
double a0=0.,a1=1.;
clrscr();
    criterio=(char**)farmalloc(sizeof(*criterio));
    if(criterio==NULL){printf("\n nao consegui alocar CRITERIO");
    exit(1);}
    criterio[0]=(char*)farmalloc(COL*sizeof(int));
    if(criterio[0]==NULL){printf("\n nao consegui alocar CRITERIO[0]");
    exit(1);}

    for(m=0;m<COL,m++)
        criterio[0][m]=0;

    for(i1=0;i1<CARD;i1++){
        for(i2=0;i2<CARD;i2++){
            for(i3=0;i3<CARD;i3++){

```

```

for(i4=0;i4<CARD;i4++){
for(i5=0;i5<CARD;i5++){
for(i6=0;i6<CARD;i6++){
for(i7=0;i7<CARD;i7++){
for(i8=0;i8<CARD;i8++){
for(i9=0;i9<CARD;i9++){
    A[0]=i1;A[1]=i2;A[2]=i3;A[3]=i4;
    A[4]=i5;A[5]=i6;A[6]=i7;A[7]=i8;A[8]=i9;
    for(i=0;i<x;i++){
        vH=0;
        for(k=0;k<SEQCOMP;k++){
            ih=0;
            for(l=k*COMP;l<k*COMP+COMP;l++){
                a=criterio[i][l]-A[l];
                a%=CARD;if(a!=0)ih=1;
                if(a<0) a=CARD+a;
                switch(a){
                    case 0:re+=1;break;
                    case 1:im+=1;break;
                    case 2:re-=1;break;
                    case 3:im-=1;break;}
            }//fim do laço l
            temp=(float)(COMP+1)*(COMP+1)-((1+re)*(1+re)+im*im);
            vH+=ih;
            if(temp==0) temp=1;
            dprod*=(float)temp;
            re=0;
            im=0;
        }//fim do laço k
    }if(dprod<vD)
    vD=dprod;
    dprod=1.;
if(vH<vHL)
    vHL=vH;
}//fim do laço i

if(vD==limiarD && vHL>=limiarH)
{
    criterio=(char**)farrealloc(criterio,(unsigned long)(x+1)*sizeof(*criterio));
    if(criterio==NULL){printf("\n nao consegui alocar CRITERIO");
        exit(1);}
    criterio[x]=(char*)farmalloc(COL*sizeof(int));
    if(criterio[x]==NULL){printf("\n nao consegui alocar CRITERIO[%d]", x);
        exit(1);}
for(m=0;m<COL;m++)
    criterio[x][m]=A[m];
    x++;printf("x=%d\n",x);
}
vD=262150.,vHL=COL;
} } } } } } } }
for(i=0;i<x;i++){
    for(k=0;k<COL;k++)
        printf("%d ",criterio[i][k]);
        printf(" ");
    }
    printf("\n");
    printf("limiarD=%f\n",limiarD);
    printf("Distancia de Hamming=%d\n",limiarH);
    printf("num de seq encontradas=%d\n",x);
    /*#####*/
//guarda o codigo mapeado no arquivo DADOS.M
ptr=fopen("c:\Veristhof\Dados.M","aw+t");
fprintf(ptr,"\n\n");
fprintf(ptr,"distancia produto=%f\n",limiarD);
fprintf(ptr,"distancia de Hamming=%d\n",limiarH);
fprintf(ptr,"comprimento do bloco %d\n",SEQCOMP);
fprintf(ptr,"numero de sequencias encontradas=%d\n",x-1);
for(i=0;i<x;i++){ if(!(i%3))
        fprintf(ptr,"\n");
        fprintf(ptr,"[");
        for(k=0;k<SEQCOMP;k++)
            fprintf(ptr,"%d",criterio[i][k]);
        fprintf(ptr,"]");

```

```

        fprintf(ptr, " ");}
fclose(ptr);
/*****
// gera o arquivo de entrada para o programa de simulação

ptr=fopen("c:\crissthof\gl.M", "wb");
for(i=0;i<x;i++){
for(k=0;k<SEQCOMP;k++){
    resul[0]=fwrite( &a1, sizeof( a1), 1, ptr);
    resul[1]=fwrite( &a0, sizeof( a1), 1, ptr);
    if((resul[0]!=1)||(resul[1]!=1))
        {printf(" \nnao consegui gravar1");
         exit(1);}
for(l=k*COMP;l<k*COMP+COMP;l++){
a=criterio[i][l];
switch(a){
case 0: resul[2]=fwrite( &a1, sizeof( a1), 1, ptr);
        resul[3]=fwrite( &a0, sizeof( a1), 1, ptr);
        if((resul[2]!=1)||(resul[3]!=1))
            {printf(" \nnao consegui gravar2");
             exit(1);}
        break;
case 1: resul[2]=fwrite( &a0, sizeof( a1), 1, ptr);
        resul[3]=fwrite( &a1, sizeof( a1), 1, ptr);
        if((resul[2]!=1)||(resul[3]!=1))
            {printf(" \nnao consegui gravar3");
             exit(1);}
        break;
case 2: a1=-a1; resul[2]=fwrite( &a1, sizeof( a1), 1, ptr);a1=-a1;
        resul[3]=fwrite( &a0, sizeof( a1), 1, ptr);
        if((resul[2]!=1)||(resul[3]!=1))
            {printf(" \nnao consegui gravar4");
             exit(1);}
        break;
case 3:  resul[2]=fwrite( &a0, sizeof( a1), 1, ptr);
        a1=-a1;resul[3]=fwrite( &a1, sizeof( a1), 1, ptr); a1=-a1;
        if((resul[2]!=1)||(resul[3]!=1))
            {printf(" \nnao consegui gravar5");
             exit(1);}
        break;}
        }//fim do laço k
        }//fim do laço l
    }//fim do laço i
fclose(ptr);
for(i=0;i<x;i++){
    farfree(criterio[i]);
    farfree(criterio);
}

```

Programa de simulação - simulac3

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <complex.h>
#include <alloc.h>
#include <fstream.h> //biblioteca adicional para geração das rotinas ruído
#include <iostream.h>

#define PI M_PI
#define PI2 M_PI_2
#define P3I2 M_PI+M_PI_2

//estas variáveis devem ter as dimensões do arquivo gl.m
#define M 16 //numero total de palavras
#define N 2 //
#define K 4 //numero de grupos
#define NK N*K

```

```

int E[ 40];
double B[ 40];
complex **C2;
/*****
// Rotinas para geração de ruído

#define NMAX 20000
#define VALORMAX 1000
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7

double Uniforme(long *idum);
double Gaussiana(long *idum);
double Rayleigh(long *idum);
////////////////////////////////////
// Funções do programa

// Uniforme - gera um numero real com distribuicao uniforme entre (0,1)
double Uniforme(long *idum)
{
    int j;
    long k;
    static long idum2=123456789;
    static long iy=0;
    static long iv[NTAB];
    double temp;

    if (*idum<=0) // inicializacao
    {
        // (realizada somente na primeira chamada)

        if(-(*idum)<1) *idum=1;
        else *idum = -(*idum);
        idum2=(*idum);
        for(j=NTAB+7;j>=0;j--) // le^ a shuffle table depois de 8 warm-ups
        {
            k=(*idum)/IQ1;
            *idum=IA1*( *idum-k*IQ1)-k*IR1;
            if(*idum<0) *idum+= IM1;
            if (j<NTAB) iv[j]=*idum;
        }
        iy=iv[0];
    }
    k=(*idum)/IQ1;
    *idum=IA1*( *idum-k*IQ1)-k*IR1; // inicio do algoritmo
    if(*idum<0) *idum+= IM1; // calcula idum=(IA1*idum)%IM1 sem problema
    k=idum2/IQ2; // de overflow, pelo metodo de Schrage
    idum2=IA2*(idum2-k*IQ2)-k*IR2; // calcula idum2=(IA2*idum2)%IM2
    if (idum2<0) idum2+=IM1;; // sem overflow da mesma forma
    j=iy/NDIV; // sera' na faixa de 0 .. NTAB-1
    iy=iv[j]-idum2;
    iv[j]=*idum;
    if(iy<1) iy+=IMM1;
    if((temp=AM*iy)>RNMX) return (RNMX);
    else return (temp);
}

// Gaussiana - gera um numero real com distribuicao gaussiana com média nula e variância unitária

double Gaussiana(long *idum)
{
    static int iset=0;

```

```

static double gset;
double fac, rsq, v1, v2;

if(iset==0)
{
    do{
        v1=2.0*Uniforme(idum)-1.0;
        v2=2.0*Uniforme(idum)-1.0;
        rsq=v1*v1+v2*v2;
    } while((rsq>=1.0)||((rsq==0.0)));
    fac=sqrt(-2.0*(log(rsq))/rsq);
    gset=v1*fac;
    iset=1;
    return (v2*fac);
}
else
{
    iset=0;
    return (gset);
}
}
// Fim das funções Ruído
/*****
int sistema( void)
{
    double x1, y1, a1, a2, snr10,o,En;
    int lin, col, pos;
    long idum;
    int snr, passo, range,resul[2];
    int l, indice;
    int s, m, n;
    double m_max, m_total;
    long int i,iter,e;
    complex m_parcial, F[ K], O[ K], Nn[ NK], R[ NK];
/*****
// carrega a matriz codigo de gl.M em C2
clrscr();
C2=(complex far **)farmalloc( M*sizeof(*C2));
if(C2==NULL)
printf("nao consegui alocar a matriz C2");
for(lin=0;lin<M;lin++){
C2[lin]=(complex far *)farmalloc((NK)*sizeof(complex));
if(C2[lin]==NULL)
printf("nao consegui alocar a linha C[%d]",lin);}
FILE *fp;
fp = fopen("c:\\cristhof\\gl.M", "rb");
if(fp==NULL)
{
    puts("\nErro na abertura do arquivo!!!");
    exit(1);
}
for( lin = 0; lin < M; lin++){
    for( col = 0; col < NK; col++){
        resul[0]=fread( &x1, sizeof( x1), 1, fp);
        resul[1]=fread( &y1, sizeof( x1), 1, fp);
        if((resul[0]!=1)||((resul[1]!=1)){
            printf("\n nao consegui ler");
            exit(1);}
        C2[lin][col]=complex(x1,y1); }
}
printf("Entre com a relacao sinal/ruído inicial: ");
scanf("%d", &snr);
printf("Entre com o passo: ");
scanf("%d", &passo);
printf("Entre com o range: ");
scanf("%d", &range);
printf("Entre com o numero de iteracoes ");
scanf("%d",&iter);

pos = -1;
for( s = snr; s <= range; s += passo)
{

```

```

printf("\nSNR = %d\n", s);
pos++; // pos 0, 1, 2, 3 ... do vetor relacao sinal ruido.
E[pos] = s;

snr10=(double)s / 10.;
En=sqrt(pow(10.,snr10)); //energia do sinal mandado no canal
e = 0;

for( i = 1; i <= iter; i+=1) // Laco de iteracao.
{
    if( !(i % 1000))
        printf("%ld\n", i);
    idum=-rand();
    l=(int)( 16 . * Uniforme(&idum)); // Escolhe palavra aleatoriamente.

    for( col = 0; col < K; col++) // Gera vetor 'fading' e 'fase aleatoria'
    {
        idum=-rand();
        x1 =(1./sqrt(2.))* Gaussiana(&idum);
        idum=-rand();
        y1 =(1./sqrt(2.))* Gaussiana(&idum);
        F[ col] = complex( x1, y1); // Vetor 'F' - Gaussiano
        idum=-rand();
        o=Uniforme(&idum)-1.;
        a1 = cos(M_PI*o);
        a2 = sin(M_PI*o);

        O[ col] = complex( a1, a2); // Vetor 'O' - Unif. Distr.(-PI,PI)
    }
    for( col = 0; col < NK; col++) // Gera vetor 'Gaussiano aditivo'
    {
        idum=-rand();
        x1 =(1./sqrt(2.))* Gaussiana(&idum);
        idum=-rand();
        y1 =(1./sqrt(2. ))* Gaussiana(&idum);
        Nn[ col] = complex( x1, y1); // Vetor 'Nn' - Gaussiano aditivo
    }
    for( m = 0; m < K; m++) // Calculo do vetor de chegada.
        for( n = 0; n < N; n++)
        {
            col = m*N + n;
            R[ col] = En*C2[ l][ col] * F[ m] * O[ m] + Nn[ col];
        }
    m_max = 0.;
    for( lin = 0; lin < M; lin++) // Varre Palavras
    {
        m_total = 0.;
        for( m = 0; m < K; m++) // Varredura dos Blocos
        {
            m_parcial = complex( 0., 0.);
            for( n = 0; n < N; n++) // Varredura dos simbolos
            {
                col = m * N + n;
                m_parcial += R[ col] * conj( En*C2[ lin][ col]);
            }
            m_total += abs( m_parcial )*abs( F[ m]);
        } // FOR 'M'
        if( m_total > m_max) // Guarda o indice da palavra de metrica
        {
            // maxima.
            m_max = m_total;
            indice = lin;
        }
    } // FOR 'LIN'
    if( l != indice) // Acumula erro.
        e++;
} // Fim FOR ITER'

B[ pos] = (double)e / (double)iter; // b = e/w; B=[B,b];
} // Fim FOR 'SNR'
for(lin=0;lin<M;lin++)
    farfree(C2[lin]);
farfree(C2);

```

```

return( pos+1);
} // Fim funcao SISTEMA.
/*****
//Funcao guarda resultados de simulacao em SIMUR.M
void grava_results( int pos)
{
FILE *fp = NULL;
int i;

fp = fopen("C:\\CRISTHOF\\DADOS.M", "aw+t");
if( fp == NULL)
{
printf("\n\nNao foi possivel gerar, ou abrir, o arquivo 'SIMUR.M'\n");
puts("Pressione qualquer tecla\n");
getch();
exit( 1);
}
fprintf( fp, "% Tipo do sistema simulado %\n");
fprintf( fp, "SNR=[");
for( i = 0; i < pos; i++)
{
fprintf( fp, "%d", E[ i]);
if( i != (pos - 1))
fprintf( fp, ",");
}
fprintf( fp, "];\n");
fprintf( fp, "PES=[");
for( i = 0; i < pos; i++)
{
fprintf( fp, "%g", B[ i]);
if( i != (pos - 1))
fprintf( fp, ",");
}
fprintf( fp, "];\n\n");

fclose( fp);

} // Fim funcao GRAVA_RESULTS
/*****
void main( void)
{
int pos;
time_t t;
srand((unsigned) time(&t));
pos = sistema();
grava_results( pos);
}

```

Anexo B

Listagem dos principais códigos encontrados

distância produto=4
distância de Hamming=2
número de grupos 2
números de sequencias encontradas=4

[00] [11] [22] [33]
...

distância produto=8
distância de Hamming=2
número de grupos 3
números de sequencias encontradas=8

[000] [012] [120]
[132] [201] [213]
[321] [333]

distância produto=16
distância de Hamming=3
número de grupos 4
números de sequencias encontradas=16

[0000] [0112] [0231]
[0323] [1021] [1133]
[1210] [1302] [2013]
[2101] [2222] [2330]
[3032] [3120] [3203]
[3311]

distância produto=16
distância de Hamming=3
número de grupos 5
números de sequencias encontradas=32

[00000] [00112] [00231]
[00323] [01021] [01133]
[01210] [01302] [02013]
[02101] [02222] [02330]
[03032] [03120] [03203]
[03311] [20011] [20103]
[20220] [20332] [21030]
[21122] [21201] [21313]
[22002] [22110] [22233]
[22321] [23023] [23131]
[23212] [23300]

distância produto=16
 distância de Hamming=4
 número de grupos 6
 números de sequencias encontradas=64

[000000] [001111] [002222]
 [003333] [010123] [011032]
 [012301] [013210] [020231]
 [021320] [022013] [023102]
 [030312] [031203] [032130]
 [033021] [100132] [101023]
 [102310] [103201] [110011]
 [111100] [112233] [113322]
 [120303] [121212] [122121]
 [123030] [130220] [131331]
 [132002] [133113] [200213]
 [201302] [202031] [203120]
 [210330] [211221] [212112]
 [213003] [220022] [221133]
 [222200] [223311] [230101]
 [231010] [232323] [233232]
 [300321] [301230] [302103]
 [303012] [310202] [311313]
 [312020] [313131] [320110]
 [321001] [322332] [323223]
 [330033] [331122] [332211]
 [333300]

distância produto=16
 distância de Hamming=4
 número de grupos 8
 números de sequencias encontradas=256

[00000000] [00001111] [00002222]
 [00003333] [00010123] [00011032]
 [00012301] [00013210] [00020231]
 [00021320] [00022013] [00023102]
 [00030312] [00031203] [00032130]
 [00033021] [00100132] [00101023]
 [00102310] [00103201] [00110011]
 [00111100] [00112233] [00113322]
 [00120303] [00121212] [00122121]
 [00123030] [00130220] [00131331]
 [00132002] [00133113] [00200213]
 [00201302] [00202031] [00203120]
 [00210330] [00211221] [00212112]
 [00213003] [00220022] [00221133]
 [00222200] [00223311] [00230101]
 [00231010] [00232323] [00233232]
 [00300321] [00301230] [00302103]
 [00303012] [00310202] [00311313]
 [00312020] [00313131] [00320110]
 [00321001] [00322332] [00323223]
 [00330033] [00331122] [00332211]
 [00333300] [11000011] [11001100]
 [11002233] [11003322] [11010132]
 [11011023] [11012310] [11013201]
 [11020220] [11021331] [11022002]

[11023113] [11030303] [11031212]
[11032121] [11033030] [11100123]
[11101032] [11102301] [11103210]
[11110000] [11111111] [11112222]
[11113333] [11120312] [11121203]
[11122130] [11123021] [11130231]
[11131320] [11132013] [11133102]
[11200202] [11201313] [11202020]
[11203131] [11210321] [11211230]
[11212103] [11213012] [11220033]
[11221122] [11222211] [11223300]
[11230110] [11231001] [11232332]
[11233223] [11300330] [11301221]
[11302112] [11303003] [11310213]
[11311302] [11312031] [11313120]
[11320101] [11321010] [11322323]
[11323232] [11330022] [11331133]
[11332200] [11333311] [22000022]
[22001133] [22002200] [22003311]
[22010101] [22011010] [22012323]
[22013232] [22020213] [22021302]
[22022031] [22023120] [22030330]
[22031221] [22032112] [22033003]
[22100110] [22101001] [22102332]
[22103223] [22110033] [22111122]
[22112211] [22113300] [22120321]
[22121230] [22122103] [22123012]
[22130202] [22131313] [22132020]
[22133131] [22200231] [22201320]
[22202013] [22203102] [22210312]
[22211203] [22212130] [22213021]
[22220000] [22221111] [22222222]
[22223333] [22230123] [22231032]
[22232301] [22233210] [22300303]
[22301212] [22302121] [22303030]
[22310220] [22311331] [22312002]
[22313113] [22320132] [22321023]
[22322310] [22323201] [22330011]
[22331100] [22332233] [22333322]
[33000033] [33001122] [33002211]
[33003300] [33010110] [33011001]
[33012332] [33013223] [33020202]
[33021313] [33022020] [33023131]
[33030321] [33031230] [33032103]
[33033012] [33100101] [33101010]
[33102323] [33103232] [33110022]
[33111133] [33112200] [33113311]
[33120330] [33121221] [33122112]
[33123003] [33130213] [33131302]
[33132031] [33133120] [33200220]
[33201331] [33202002] [33203113]
[33210303] [33211212] [33212121]
[33213030] [33220011] [33221100]
[33222233] [33223322] [33230132]
[33231023] [33232310] [33233201]
[33300312] [33301203] [33302130]
[33303021] [33310231] [33311320]
[33312013] [33313102] [33320123]
[33321032] [33322301] [33323210]

[33330000] [33331111] [33332222]
[33333333]

distância produto=32
distância de Hamming=5
número de grupos 10
números de sequencias encontradas=1024

[0000000000] [0000011111] [0000022222]
[0000033333] [0001100112] [0001111003]
[0001122330] [0001133221] [0002200223]
[0002211332] [0002222001] [0002233110]
[0003300331] [0003311220] [0003322113]
[0003333002] [0010101021] [0010110130]
[0010123203] [0010132312] [0011001133]
[0011010022] [0011023311] [0011032200]
[0012301202] [0012310313] [0012323020]
[0012332131] [0013201310] [0013210201]
[0013223132] [0013232023] [0020202032]
[0020213123] [0020220210] [0020231301]
[0021302120] [0021313031] [0021320302]
[0021331213] [0022002211] [0022013300]
[0022020033] [0022031122] [0023102303]
[0023113212] [0023120121] [0023131030]
[0030303013] [0030312102] [0030321231]
[0030330320] [0031203101] [0031212010]
[0031221323] [0031230232] [0032103230]
[0032112321] [0032121012] [0032130103]
[0033003322] [0033012233] [0033021100]
[0033030011] [0100101210] [0100110301]
[0100123032] [0100132123] [0101001302]
[0101010213] [0101023120] [0101032031]
[0102301033] [0102310122] [0102323211]
[0102332300] [0103201121] [0103210030]
[0103223303] [0103232212] [0110000231]
[0110011320] [0110022013] [0110033102]
[0111100323] [0111111232] [0111122101]
[0111133010] [0112200012] [0112211103]
[0112222230] [0112233321] [0113300100]
[0113311011] [0113322322] [0113333233]
[0120303222] [0120312333] [0120321000]
[0120330111] [0121203330] [0121212221]
[0121221112] [0121230003] [0122103001]
[0122112110] [0122121223] [0122130332]
[0123003113] [0123012002] [0123021331]
[0123030220] [0130202203] [0130213312]
[0130220021] [0130231130] [0131302311]
[0131313200] [0131320133] [0131331022]
[0132002020] [0132013131] [0132020202]
[0132031313] [0133102132] [0133113023]
[0133120310] [0133131201] [0200202320]
[0200213231] [0200220102] [0200231013]
[0201302232] [0201313323] [0201320010]
[0201331101] [0202002103] [0202013012]
[0202020321] [0202031230] [0203102011]
[0203113100] [0203120233] [0203131322]

[0210303301] [0210312210] [0210321123]
[0210330032] [0211203213] [0211212302]
[0211221031] [0211230120] [0212103122]
[0212112033] [0212121300] [0212130211]
[0213003030] [0213012121] [0213021212]
[0213030303] [0220000312] [0220011203]
[0220022130] [0220033021] [0221100200]
[0221111311] [0221122022] [0221133133]
[0222200131] [0222211020] [0222222313]
[0222233202] [0223300023] [0223311132]
[0223322201] [0223333310] [0230101333]
[0230110222] [0230123111] [0230132000]
[0231001221] [0231010330] [0231023003]
[0231032112] [0232301110] [0232310001]
[0232323332] [0232332223] [0233201002]
[0233210113] [0233223220] [0233232331]
[0300303130] [0300312021] [0300321312]
[0300330203] [0301203022] [0301212133]
[0301221200] [0301230311] [0302103313]
[0302112202] [0302121131] [0302130020]
[0303003201] [0303012310] [0303021023]
[0303030132] [0310202111] [0310213000]
[0310220333] [0310231222] [0311302003]
[0311313112] [0311320221] [0311331330]
[0312002332] [0312013223] [0312020110]
[0312031001] [0313102220] [0313113331]
[0313120002] [0313131113] [0320101102]
[0320110013] [0320123320] [0320132231]
[0321001010] [0321010101] [0321023232]
[0321032323] [0322301321] [0322310230]
[0322323103] [0322332012] [0323201233]
[0323210322] [0323223011] [0323232100]
[0330000123] [0330011032] [0330022301]
[0330033210] [0331100031] [0331111120]
[0331122213] [0331133302] [0332200300]
[0332211211] [0332222122] [0332233033]
[0333300212] [0333311303] [0333322030]
[0333333121] [1000102101] [1000113010]
[1000120323] [1000131232] [1001002013]
[1001013102] [1001020231] [1001031320]
[1002302322] [1002313233] [1002320100]
[1002331011] [1003202230] [1003213321]
[1003220012] [1003231103] [1010003120]
[1010012031] [1010021302] [1010030213]
[1011103032] [1011112123] [1011121210]
[1011130301] [1012203303] [1012212212]
[1012221121] [1012230030] [1013303211]
[1013312300] [1013321033] [1013330122]
[1020300133] [1020311022] [1020322311]
[1020333200] [1021200021] [1021211130]
[1021222203] [1021233312] [1022100310]
[1022111201] [1022122132] [1022133023]
[1023000202] [1023011313] [1023022020]
[1023033131] [1030201112] [1030210003]
[1030223330] [1030232221] [1031301000]
[1031310111] [1031323222] [1031332333]
[1032001331] [1032010220] [1032023113]
[1032032002] [1033101223] [1033110332]
[1033123001] [1033132110] [1100003311]

[1100012200] [1100021133] [1100030022]
[1101103203] [1101112312] [1101121021]
[1101130130] [1102203132] [1102212023]
[1102221310] [1102230201] [1103303020]
[1103312131] [1103321202] [1103330313]
[1110102330] [1110113221] [1110120112]
[1110131003] [1111002222] [1111013333]
[1111020000] [1111031111] [1112302113]
[1112313002] [1112320331] [1112331220]
[1113202001] [1113213110] [1113220223]
[1113231332] [1120201323] [1120210232]
[1120223101] [1120232010] [1121301231]
[1121310320] [1121323013] [1121332102]
[1122001100] [1122010011] [1122023322]
[1122032233] [1123101012] [1123110103]
[1123123230] [1123132321] [1130300302]
[1130311213] [1130322120] [1130333031]
[1131200210] [1131211301] [1131222032]
[1131233123] [1132100121] [1132111030]
[1132122303] [1132133212] [1133000033]
[1133011122] [1133022211] [1133033300]
[1200300221] [1200311330] [1200322003]
[1200333112] [1201200333] [1201211222]
[1201222111] [1201233000] [1202100002]
[1202111113] [1202122220] [1202133331]
[1203000110] [1203011001] [1203022332]
[1203033223] [1210201200] [1210210311]
[1210223022] [1210232133] [1211301312]
[1211310203] [1211323130] [1211332021]
[1212001023] [1212010132] [1212023201]
[1212032310] [1213101131] [1213110020]
[1213123313] [1213132202] [1220102213]
[1220113302] [1220120031] [1220131120]
[1221002301] [1221013210] [1221020123]
[1221031032] [1222302030] [1222313121]
[1222320212] [1222331303] [1223202122]
[1223213033] [1223220300] [1223231211]
[1230003232] [1230012323] [1230021010]
[1230030101] [1231103320] [1231112231]
[1231121102] [1231130013] [1232203011]
[1232212100] [1232221233] [1232230322]
[1233303103] [1233312012] [1233321321]
[1233330230] [1300201031] [1300210120]
[1300223213] [1300232302] [1301301123]
[1301310032] [1301323301] [1301332210]
[1302001212] [1302010303] [1302023030]
[1302032121] [1303101300] [1303110211]
[1303123122] [1303132033] [1310300010]
[1310311101] [1310322232] [1310333323]
[1311200102] [1311211013] [1311222320]
[1311233231] [1312100233] [1312111322]
[1312122011] [1312133100] [1313000321]
[1313011230] [1313022103] [1313033012]
[1320003003] [1320012112] [1320021221]
[1320030330] [1321103111] [1321112000]
[1321121333] [1321130222] [1322203220]
[1322212331] [1322221002] [1322230113]
[1323303332] [1323312223] [1323321110]
[1323330001] [1330102022] [1330113133]

[1330120200] [1330131311] [1331002130]
[1331013021] [1331020312] [1331031203]
[1332302201] [1332313310] [1332320023]
[1332331132] [1333202313] [1333213202]
[1333220131] [1333231020] [2000203202]
[2000212313] [2000221020] [2000230131]
[2001303310] [2001312201] [2001321132]
[2001330023] [2002003021] [2002012130]
[2002021203] [2002030312] [2003103133]
[2003112022] [2003121311] [2003130200]
[2010302223] [2010313332] [2010320001]
[2010331110] [2011202331] [2011213220]
[2011220113] [2011231002] [2012102000]
[2012113111] [2012120222] [2012131333]
[2013002112] [2013013003] [2013020330]
[2013031221] [2020001230] [2020010321]
[2020023012] [2020032103] [2021101322]
[2021110233] [2021123100] [2021132011]
[2022201013] [2022210102] [2022223231]
[2022232320] [2023301101] [2023310010]
[2023323323] [2023332232] [2030100211]
[2030111300] [2030122033] [2030133122]
[2031000303] [2031011212] [2031022121]
[2031033030] [2032300032] [2032311123]
[2032322210] [2032333301] [2033200120]
[2033211031] [2033222302] [2033233213]
[2100302012] [2100313103] [2100320230]
[2100331321] [2101202100] [2101213011]
[2101220322] [2101231233] [2102102231]
[2102113320] [2102120013] [2102131102]
[2103002323] [2103013232] [2103020101]
[2103031010] [2110203033] [2110212122]
[2110221211] [2110230300] [2111303121]
[2111312030] [2111321303] [2111330212]
[2112003210] [2112012301] [2112021032]
[2112030123] [2113103302] [2113112213]
[2113121120] [2113130031] [2120100020]
[2120111131] [2120122202] [2120133313]
[2121000132] [2121011023] [2121022310]
[2121033201] [2122300203] [2122311312]
[2122322021] [2122333130] [2123200311]
[2123211200] [2123222133] [2123233022]
[2130001001] [2130010110] [2130023223]
[2130032332] [2131101113] [2131110002]
[2131123331] [2131132220] [2132201222]
[2132210333] [2132223000] [2132232111]
[2133301330] [2133310221] [2133323112]
[2133332003] [2200001122] [2200010033]
[2200023300] [2200032211] [2201101030]
[2201110121] [2201123212] [2201132303]
[2202201301] [2202210210] [2202223123]
[2202232032] [2203301213] [2203310302]
[2203323031] [2203332120] [2210100103]
[2210111012] [2210122321] [2210133230]
[2211000011] [2211011100] [2211022233]
[2211033322] [2212300320] [2212311231]
[2212322102] [2212333013] [2213200232]
[2213211323] [2213222010] [2213233101]
[2220203110] [2220212001] [2220221332]

[2220230223] [2221303002] [2221312113]
[2221321220] [2221330331] [2222003333]
[2222012222] [2222021111] [2222030000]
[2223103221] [2223112330] [2223121003]
[2223130112] [2230302131] [2230313020]
[2230320313] [2230331202] [2231202023]
[2231213132] [2231220201] [2231231310]
[2232102312] [2232113203] [2232120130]
[2232131021] [2233002200] [2233013311]
[2233020022] [2233031133] [2300100332]
[2300111223] [2300122110] [2300133001]
[2301000220] [2301011331] [2301022002]
[2301033113] [2302300111] [2302311000]
[2302322333] [2302333222] [2303200003]
[2303211112] [2303222221] [2303233330]
[2310001313] [2310010202] [2310023131]
[2310032020] [2311101201] [2311110310]
[2311123023] [2311132132] [2312201130]
[2312210021] [2312223312] [2312232203]
[2313301022] [2313310133] [2313323200]
[2313332311] [2320302300] [2320313211]
[2320320122] [2320331033] [2321202212]
[2321213303] [2321220030] [2321231121]
[2322102123] [2322113032] [2322120301]
[2322131210] [2323002031] [2323013120]
[2323020213] [2323031302] [2330203321]
[2330212230] [2330221103] [2330230012]
[2331303233] [2331312322] [2331321011]
[2331330100] [2332003102] [2332012013]
[2332021320] [2332030231] [2333103010]
[2333112101] [2333121232] [2333130323]
[3000301303] [3000310212] [3000323121]
[3000332030] [3001201211] [3001210300]
[3001223033] [3001232122] [3002101120]
[3002110031] [3002123302] [3002132213]
[3003001032] [3003010123] [3003023210]
[3003032301] [3010200322] [3010211233]
[3010222100] [3010233011] [3011300230]
[3011311321] [3011322012] [3011333103]
[3012000101] [3012011010] [3012022323]
[3012033232] [3013100013] [3013111102]
[3013122231] [3013133320] [3020103331]
[3020112220] [3020121113] [3020130002]
[3021003223] [3021012332] [3021021001]
[3021030110] [3022303112] [3022312003]
[3022321330] [3022330221] [3023203000]
[3023212111] [3023221222] [3023230333]
[3030002310] [3030013201] [3030020132]
[3030031023] [3031102202] [3031113313]
[3031120020] [3031131131] [3032202133]
[3032213022] [3032220311] [3032231200]
[3033302021] [3033313130] [3033320203]
[3033331312] [3100200113] [3100211002]
[3100222331] [3100233220] [3101300001]
[3101311110] [3101322223] [3101333332]
[3102000330] [3102011221] [3102022112]
[3102033003] [3103100222] [3103111333]
[3103122000] [3103133111] [3110301132]
[3110310023] [3110323310] [3110332201]

[3111201020] [3111210131] [3111223202]
 [3111232313] [3112101311] [3112110200]
 [3112123133] [3112132022] [3113001203]
 [3113010312] [3113023021] [3113032130]
 [3120002121] [3120013030] [3120020303]
 [3120031212] [3121102033] [3121113122]
 [3121120211] [3121131300] [3122202302]
 [3122213213] [3122220120] [3122231031]
 [3123302210] [3123313301] [3123320032]
 [3123331123] [3130103100] [3130112011]
 [3130121322] [3130130233] [3131003012]
 [3131012103] [3131021230] [3131030321]
 [3132303323] [3132312232] [3132321101]
 [3132330010] [3133203231] [3133212320]
 [3133221013] [3133230102] [3200103023]
 [3200112132] [3200121201] [3200130310]
 [3201003131] [3201012020] [3201021313]
 [3201030202] [3202303200] [3202312311]
 [3202321022] [3202330133] [3203203312]
 [3203212203] [3203221130] [3203230021]
 [3210002002] [3210013113] [3210020220]
 [3210031331] [3211102110] [3211113001]
 [3211120332] [3211131223] [3212202221]
 [3212213330] [3212220003] [3212231112]
 [3213302333] [3213313222] [3213320111]
 [3213331000] [3220301011] [3220310100]
 [3220323233] [3220332322] [3221201103]
 [3221210012] [3221223321] [3221232230]
 [3222101232] [3222110323] [3222123010]
 [3222132101] [3223001320] [3223010231]
 [3223023102] [3223032013] [3230200030]
 [3230211121] [3230222212] [3230233303]
 [3231300122] [3231311033] [3231322300]
 [3231333211] [3232000213] [3232011302]
 [3232022031] [3232033120] [3233100301]
 [3233111210] [3233122123] [3233133032]
 [3300002233] [3300013322] [3300020011]
 [3300031100] [3301102321] [3301113230]
 [3301120103] [3301131012] [3302202010]
 [3302213101] [3302220232] [3302231323]
 [3303302102] [3303313013] [3303320320]
 [3303331231] [3310103212] [3310112303]
 [3310121030] [3310130121] [3311003300]
 [3311012211] [3311021122] [3311030033]
 [3312303031] [3312312120] [3312321213]
 [3312330302] [3313203123] [3313212032]
 [3313221301] [3313230210] [3320200201]
 [3320211310] [3320222023] [3320233132]
 [3321300313] [3321311202] [3321322131]
 [3321333020] [3322000022] [3322011133]
 [3322022200] [3322033311] [3323100130]
 [3323111021] [3323122312] [3323133203]
 [3330301220] [3330310331] [3330323002]
 [3330332113] [3331201332] [3331210223]
 [3331223110] [3331232001] [3332101003]
 [3332110112] [3332123221] [3332132330]
 [3333001111] [3333010000] [3333023333]
 [3333032222]

#####

N=3

distância $D_\delta = 16$

distância de Hamming=2

número de grupos 2

números de sequencias encontradas=16

[0000] [0101] [0202]
 [0303] [1010] [1111]
 [1212] [1313] [2020]
 [2121] [2222] [2323]
 [3030] [3131] [3232]
 [3333]

distância $D_\delta = 64$

distância de Hamming=2

número de grupos 3

números de sequencias encontradas=64

[000000] [000202] [002020]
 [002222] [010101] [010303]
 [012121] [012323] [020002]
 [020200] [022022] [022220]
 [030103] [030301] [032123]
 [032321] [101010] [101212]
 [103030] [103232] [111111]
 [111313] [113131] [113333]
 [121012] [121210] [123032]
 [123230] [131113] [131311]
 [133133] [133331] [200020]
 [200222] [202000] [202202]
 [210121] [210323] [212101]
 [212303] [220022] [220220]
 [222002] [222200] [230123]
 [230321] [232103] [232301]
 [301030] [301232] [303010]
 [303212] [311131] [311333]
 [313111] [313313] [321032]
 [321230] [323012] [323210]
 [331133] [331331] [333113]
 [333311]

distância $D_\delta = 128$

distância de Hamming=3

número de grupos 4

números de sequencias encontradas=182

[00000000] [00010102] [00020210]
 [00030312] [00101020] [00111122]
 [00121230] [00131332] [00202001]
 [00212103] [00222311] [00232213]
 [00303021] [00313123] [00323231]
 [01000120] [01010222] [01020001]
 [01031003] [01101200] [01110302]
 [01121110] [01132023] [01202121]

[01212212] [01221333] [01232331]
[01303230] [01313332] [01323013]
[01333111] [02000201] [02010003]
[02020121] [02031123] [02100310]
[02111030] [02121311] [02131202]
[02202233] [02212022] [02222113]
[02233000] [02302312] [02333232]
[03001002] [03010320] [03021100]
[03030233] [03100111] [03111203]
[03120012] [03132122] [03201313]
[03212330] [03223032] [03233201]
[03312221] [03322010] [03333323]
[10001212] [10011011] [10021323]
[10030021] [10100231] [10112013]
[10120303] [10132300] [10201102]
[10210133] [10223222] [10233110]
[10332130] [11001310] [11012030]
[11022102] [11032211] [11103003]
[11113121] [11123320] [11130112]
[11200300] [11210031] [11231022]
[11301232] [12000323] [12011112]
[12021233] [12031331] [12102101]
[12112232] [12120213] [12200110]
[12213220] [12230002] [12302003]
[12323300] [13002021] [13011231]
[13021032] [13032313] [13103322]
[13113001] [13122223] [13130030]
[13203112] [13220200] [13231111]
[13313203] [20001103] [20012310]
[20023012] [20033100] [20102230]
[20110020] [20120132] [20130311]
[20201023] [20211213] [20222031]
[20233333] [20300202] [20331322]
[21002333] [21012201] [21020331]
[21031220] [21103212] [21203011]
[21211010] [21223302] [21230130]
[21300013] [21320203] [22003031]
[22013233] [22022011] [22101113]
[22110100] [22130221] [22202320]
[22211301] [22220023] [22323130]
[22332203] [23003113] [23030122]
[23102033] [23113211] [23121000]
[23132210] [23200230] [23210332]
[23221221] [23301120] [23312302]
[23322112] [30012132] [30102321]
[30123101] [30131131] [30301300]
[30310010] [31003311] [31022222]
[31100032] [31213102] [31221101]
[31301031] [31310220] [32013023]
[32022303] [32122131] [32300322]
[32310101] [32321033] [32331121]
[33023330] [33110223] [33133103]
[33201222] [33302111]

Bibliografia

- [1] **J. G. Proakis**, Digital Communications, New York, Mc Grall-Hill, 1995.
- [2] **B. P. Lathi**, Random Signals and Communication Theory, International Textbook Company, 1968.
- [3] **D. Divsalar e M. K. Simon**, "Multiple-Symbol Differential Detection of MPSK", IEEE Transaction on Communication, vol. 38, NO.3, pp 300-308, Março 1990.
- [4] **D. Divsalar e M. K. Simon**, "Maximum Likelihood Diferencial Detection of Uncoded and Trellis coded Amplitude Phase Modulation over AWGN and Fading Channels: metric performance", IEEE Transactions on Communications. , vol 42, NO.1, pp. 76-89, Janeiro 1994.
- [5] **A. J. Viterbi, J. K. Omura**, Principles of Digital Communication and Coding, Mc Graw-Hill, 1979.
- [6] **J. Ventura-Traveset, G. Caire, E. Biglieri**, "Impact of Diversity Reception on Fading Channels with Coded Modulation. Part II: Differencial Block Detection", <http://hp0tlc.polito.it/group.html>.
- [7] **M. Schwartz. W. R. Bennete Stein**, Communication System and Techiques, Mc Grall-Hill, 1966.
- [8] **J. M. Wozencraft, I. M. Jacobs**, Principles of Communication Engineering, Wiley, 1965.
- [9] **E. Biglieri, D. Divsalar, P. J. Mclane, M. K. Simon**, Introduction to Trellis-Coded Modulation with Aplications, New York, Macmillan Publishing Company, 1991.

-
- [10] **S. Haykin**, Digital Communications, Wiley, 1988.
- [11] **S. H. Jamali, T. Le-Ngoc**, Coded Modulation Techniques for Fading Channels, Boston, Kluwer, 1994.
- [12] **G. Kaplan e S. Shamai (Shitz)**, "On information Rates in Fading Channels with Partial Side Information", ETT, vol.6, NO.6, pp. 665-669, Novembro-Dezembro 1995.
- [13] **C. D. de Alencar e J. Portugheis**, "Construção de Códigos de Bloco M-PSK para canais com desvanecimento", Anais do 13º SBT, pp. 662-666, Águas de Lindóia, SP, Setembro 1995.
- [14] **S. Benedetto, E. Biglieri e V. Castellani**, Digital Transmission Theory, Prentice-Hall, 1987.