UNIVERSIDADE ESTADUAL DE CAMPINAS.

FACULDADE DE ENGENHARIA ELÉTRICA.

DEPARTAMENTO DE COMUNICAÇÕES.

ALGORITMOS PARA DECODIFICAÇÃO DE CÓDIGOS DE BLOCO COM DECISÃO SUAVE E APLICAÇÕES EM SISTEMAS CONCATENADOS GENERALIZADOS.

AUTOR: WALTER GODOY JÚNIOR

ORIENTADOR: PROF. DR. DALTON SOARES ARANTES.

CO-ORIENTADOR: PROF. DR. REGINALDO PALAZZO JÚNIOR.

Este exemplar corresponde a redação final da Tese defendida por Walter Godoy Júnior e aprovada pela Comissão Julgadora em 02 de Abril de 1990.

Dalton S. [illegible]

Tese apresentada ao Departamento de Comunicações da Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, como requisito parcial para a obtenção do grau de Doutor em Engenharia .

Campinas, 1990.

À minha esposa Elena,

Aos meus filhos

Ana Marina e Walter,

À memória dos meus avós

Amália e Marino Tonsig,

Aos meus pais.

AGRADECIMENTOS

Quero expressar minha gratidão ao professor Doutor Dalton Soares Arantes por ter me orientado com muita paciência e dedicação na elaboração desta tese e por ter me mostrado os caminhos da pesquisa científica.

Ao professor Doutor Reginaldo Palazzo Júnior, os meus agradecimentos pela motivadora confiança no meu trabalho, pelas valiosas sugestões, críticas, discussões e paciente orientação.

Desejo ainda agradecer:

Aos professores Doutores Carlos Hartmann e Gur Dial pelas idéias que alimentaram esta tese, pelas longas e frutíferas discussões e pela valiosa bibliografia cedida.

Aos alunos e participantes do projeto MICROSAT do CEFET-PR por terem contribuído, de uma maneira ou de outra, para que esta tese fosse concluída.

Um agradecimento muito especial à minha esposa Elena pelas inúmeras revisões e aos meus filhos Ana Marina e Walter pela enorme paciência que tiveram para conviver comigo durante a elaboração desta tese.

À Coordenação do Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Ao Centro Federal de Educação Tecnológica do Paraná pela confiança em mim depositada e pelo constante apoio.

ÍNDICE

Pág.

Pág.

Pág.

LISTA DE FIGURAS.

Pág.

Pág.

LISTA DE TABELAS

Pág.

RESUMO

Esta tese tem dois objetivos fundamentais: o primeiro é apresentar um sistema original para a decodificação de códigos de bloco com decisão suave. Para tanto, foi desenvolvido um algoritmo fundamentado nos algoritmos de Conjuntos de Informação e de Vizinhos de Zero, sendo feita também uma extensão do algoritmo proposto para os códigos multiníveis.

O segundo objetivo deste trabalho é apresentar um tratamento da concatenação de códigos de bloco usando como instrumento a teoria de reticulados. São analisados alguns tipos de concatenação sob o ponto de vista da Concatenação Generalizada (CG), bem como propostos novos esquemas de codificação e decodificação de códigos concatenados.

CAPÍTULO 1

INTRODUÇÃO.

Neste capítulo é feito um resumo histórico das técnicas de decodificação com decisão suave e de concatenação de códigos, bem como são delineados os objetivos do trabalho.

O conceito de decodificação com decisão suave é fundamentado no uso da informação de confiabilidade sobre cada símbolo da palavra recebida [8]. Tal informação é fornecida pelo circuito do demodulador. Esta técnica pode proporcionar um ganho típico de dois dB em relação ao decodificador abrupto [8].

É bem conhecido que o uso da decisão suave para decodificação do código interno melhora consideravelmente o desempenho dos códigos concatenados [5], daí o interesse manifestado neste trabalho.

A técnica de correção de erros com decisão suave apresenta ainda inúmeros problemas em aberto. Todos os algoritmos existentes tratam, de uma forma ou de outra, de fazer bom uso desta informação extra. Entre os algoritmos mais conhecidos destacam-se o de Chase [7], o de Weldon [52], o algoritmo ótimo de decodificação bit-a-bit de Hartmann-Rudolph [28] e G. Battail [2] e o algoritmo de Massey [32]. Existe, ainda, uma outra classe de algoritmos que utiliza os conjuntos de informação. A decodificação por conjuntos de informação foi introduzida por Prange [40] em 1962. Desde então foram pesquisados e desenvolvidos vários algoritmos que utilizam esta técnica não algébrica de decodificação. Entre eles destacam-se o algoritmo

tradicional de conjuntos de informação [8], o algoritmo de Omura [36] e o algoritmo de conjuntos de informação com síndrome parcial [8].

Neste trabalho, é proposta uma modificação do algoritmo tradicional de conjuntos de informação, bem como uma generalização do algoritmo de Vizinhos de Zero [30] originalmente desenvolvido para decisão abrupta. A aplicação de limiares de distância mínima torna o algoritmo proposto adaptativo ao canal. Esta adaptabilidade possui a vantagem, em relação aos algoritmos citados, de uma diminuição do tempo médio de decodificação. O algoritmo proposto possui, também, a vantagem de ser facilmente estendido para códigos multiníveis e, conseqüentemente, para sistemas concatenados.

O conceito de concatenação de códigos é conhecido há vários anos [18] e utilizado largamente nas comunicações via satélite. Forney [18] apresentou os conceitos básicos de concatenação e de supercanal, bem como analisou algumas das propriedades fundamentais de sistemas concatenados.

Em 1974, Blokh [4] apresentou o conceito de códigos concatenados generalizados . Em 1976, Zinoviev [56] apresentou um método para a construção de códigos concatenados generalizados, tanto para códigos lineares quanto não lineares. Em 1981, Zinoviev [57] publicou um algoritmo para a decodificação desses códigos. A partir de 1987, Zinoviev, Zyablov e Portnoy estudaram códigos concatenados no espaço Euclidiano [58]. Embora, desde 1966, extensos estudos estejam sendo feitos, a construção e o uso desses códigos para alguns tipos de aplicação permanecem sem solução [5].

Este trabalho tem por objetivo também o tratamento da

concatenação de códigos de bloco, usando como instrumento de avaliação a teoria dos reticulados. É feito um levantamento de alguns tipos de concatenação e especialmente o da Concatenação Generalizada de Códigos (CGC). São realizadas análises e comparações entre sistemas de Modulação Codificada (MD) considerados como casos particulares de CGC. É feita uma breve apresentação da teoria dos reticulados [12,46,47], bem como a sua conexão com a teoria da CGC. No capítulo final desta tese, são apresentados esquemas de decodificação para códigos multiníveis aplicados em modulação codificada .

Em suma, neste trabalho apresenta-se uma contribuição relacionada com os seguintes tópicos:

1-transformações e particularidades dos reticulados que correspondem à operação de Concatenação Generalizada de códigos de bloco;

2-tipos de reticulados e sistemas de concatenação apropriados para se obter esquemas eficientes de decodificação;

3-algoritmos para decodificação suave de códigos de bloco com aplicações em sistemas concatenados.

Para cumprir os objetivos propostos, a tese é dividida em sete capítulos.

O segundo capítulo apresenta uma revisão sobre os conceitos de reticulados, suas construções, bem como os seus parâmetros. É mostrada, também, a relação existente entre códigos corretores de erros e reticulados.

O terceiro capítulo introduz os conceitos fundamentais relativos à Concatenação Generalizada utilizando códigos de bloco. Esse capítulo representa uma contribuição relativa a esquemas de decodificação de códigos concatenados com a

utilização de particularidades, propriedades e transformações dos reticulados.

No quarto e no quinto capítulos, é apresentada uma contribuição ao estudo de algoritmos para a decodificação com decisão suave. A motivação deste estudo é, em grande parte, devida às aplicações que a técnica de decisão suave tem em códigos de bloco [5], em reticulados [46] e em concatenação de códigos.

No sexto capítulo é desenvolvida uma extensão dos algoritmos apresentados no quarto e no quinto capítulos para os códigos multiníveis. Finalmente, o sétimo capítulo analisa os resultados obtidos e apresenta as considerações finais.

CAPÍTULO 2

REVISÃO DE RETICULADOS.

2.1 INTRODUÇÃO.

Neste capítulo serão apresentados os conceitos e definições sobre reticulados utilizados ao longo deste trabalho.

O conceito de reticulados será utilizado aqui, como ferramenta de avaliação de particionamentos de códigos e sinais em sistemas de concatenação generalizada.

Estes conceitos e definições são facilmente encontrados na literatura técnica, tais como [9]-[13], [45-47] e serão aqui reproduzidos para facilidade de exposição.

2.2. RETICULADOS: DEFINIÇÕES E PARÂMETROS.

Definição 2.2.1 [9]: Sejam $x_1,.....,x_n$ vetores linearmente independentes no espaço real Euclidiano de m dimensões $\mathbb{R}^m$, onde $m \geq n$. Então o conjunto de todos os vetores $x = u_1x_1 + ...+u_nx_n$, onde u_1, $....,u_n$ são inteiros, é chamado de reticulado de n dimensões Λ.

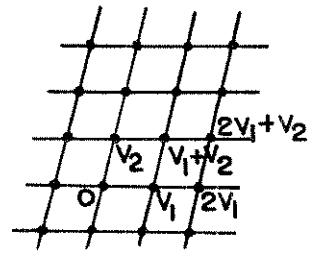
Desta forma, um reticulado Λ é uma representação de um código linear em um espaço Euclidiano, através de números inteiros.

É importante notar que, de acordo com [12], é conveniente usar $m>n$ coordenadas para descrever um reticulado de n dimensões.

Alguns exemplos simples de reticulados são mostrados na

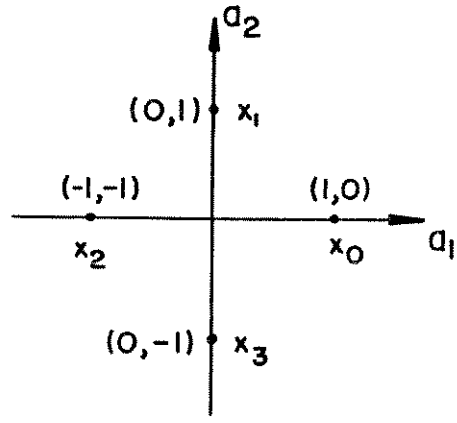
figura 2.2.1.

Neste trabalho, para representar um código de bloco linear, será adotada a notação utilizada em [5], ou seja C(q;n,M,d), onde q é a cardinalidade do alfabeto, n é o comprimento do código, d é a distância mínima de Hamming e M é a cardinalidade do código C.

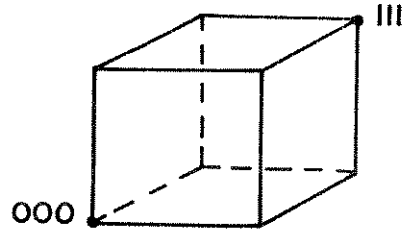


Célula básica do reticulado.

(a).



(b).



(c).

Fig. 2.2.1 Exemplos de reticulados e seus subconjuntos.

(a) Reticulado em $\mathbb{R}^2$.

(b) Modulação 4PSK em $\mathbb{R}^2$- subconjunto de um reticulado.

(c) Representação do código C(2;3,2,3) em $\mathbb{R}^3$-subconjunto de um reticulado.

Através da teoria de reticulados, pode ser tratada uma grande classe de constelações de sistemas modulados, bem como a representação de códigos no espaço Euclidiano n-dimensional. Uma das aplicações da associação de reticulados e códigos é facilitar o desenvolvimento de algoritmos para decodificação de códigos utilizando a técnica de decisão suave [13].

Assim como os códigos lineares, um reticulado pode ser gerado por matrizes denominadas geradoras, como segue.

Definição 2.2.2 [9]: A **matriz geradora** M, de dimensão k x n, do reticulado Λ será representada por $M = (x_j^{(i)})$, $1 \leq i \leq k$ e $1 \leq j \leq n$, onde $x^{(i)}$ é o i-ésimo vetor linearmente independente.

A obtenção de reticulados pode ser vista como uma subclasse de problemas de um problema mais geral denominado de **empacotamento esférico.** Como neste trabalho serão analisados somente empacotamentos reticulados, será adotada a mesma representação Λ tanto para os reticulados como para os empacotamentos esféricos.

Definição 2.2.3 [47]: O **empacotamento esférico** no $\mathbb{R}^n$ é o arranjo de infinitas esferas de mesmo tamanho que não se interceptam. Mais precisamente, um empacotamento esférico Λ no $\mathbb{R}^n$, de raio ρ, consiste de uma seqüência infinita de pontos (centros das esferas) $x^{(1)}, x^{(2)}, \ldots\ldots$ no $\mathbb{R}^n$ tais que $dist(x^{(i)}, x^{(j)}) = \|x^{(i)} - x^{(j)}\| \geq 2\rho$, para todo $i \neq j$.

Assim, se as esferas de raio ρ são posicionadas sobre os centros $x^{(1)}$, $x^{(2)}$,....., as esferas não se interceptam.

Quando os centros das esferas $x^{(i)}$ formam um grupo Abeliano, Λ é chamado de **empacotamento** em um reticulado.

Definição 2.2.4 [47]: A **dimensão** k de um empacotamento de um reticulado Λ é o número máximo de centros linearmente independentes em Λ.

Sejam $x^{(i)} = (x_1^{(i)},\ldots,x_n^{(i)})$, para $1 \leq i < k$, centros linearmente independentes que geram Λ. Como $\Lambda \subseteq \mathbb{R}^n$, então $k \leq n$.

Definição 2.2.5 [47]: O determinante de Λ é definido como:

$$\det \Lambda = (\det MM^T)^{1/2}.$$

Quando o reticulado possui a mesma dimensão do espaço no qual está inserido, isto é, quando k=n, M é uma matriz quadrada n X n não singular e $\det \Lambda = |\det M|$.

Outro parâmetro importante na teoria dos reticulados é a energia de seus elementos.

Definição 2.2.6: A energia, norma ou módulo de um vetor $x = (x_1,..,x_m) \in \mathbb{R}^m$ é dada por:

$$||x|| = \sqrt{x.x} = \left(\sum x_i^2\right)^{1/2} \qquad (2.1)$$

Definição 2.2.7 [47]: A densidade δ de qualquer empacotamento esférico (reticulado ou não) é a fração do espaço $\mathbb{R}^n$ que é coberta pelas esferas.

Para um empacotamento esférico reticulado Λ de raio ρ e dimensão n, a densidade é dada por [47]:

$$\delta = V_n \rho^n / \det \Lambda \ , \tag{2.2}$$

onde

$$V_n := \Pi^{n/2} / \Gamma((n/2)+1) \tag{2.3}$$

é o volume de uma esfera de raio unitário, onde Γ é a função gama .

Definição 2.2.8 [47]: O número de contatos $\tau(x)$ da esfera centrada em x é o número de esferas vizinhas, isto é, o número de esferas que tocam a esfera centrada em x.

O valor máximo de $\tau(x)$ para $x \in \Lambda$ é representado por τ_{max}. Para um empacotamento de um reticulado, $\tau(x)=\tau_{max}=\tau$, independente da escolha de x [47].

Basicamente, são dois os problemas fundamentais de empacotamentos esféricos: a determinação do empacotamento mais denso e a determinação do maior número de contatos.

Exemplo de reticulado ótimo: Os reticulados n dimensionais D_n em $\mathbb{R}^n$, representam os reticulados mais densos possíveis para n=3,4 e 5 [47]. Para estes reticulados k=n e a

matriz geradora M é :

$$M = 1/\sqrt{2} \cdot \begin{bmatrix} 2 & 0 & 0 & . & . & . & 0 \\ 1 & 1 & 0 & . & . & . & 0 \\ 1 & 0 & 1 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ 1 & 0 & 0 & . & . & . & 1 \end{bmatrix}$$

sendo $\rho = 1/2$, $det = 2^{-(n-2)/2}$ e $\tau = 2n(n-1)$. Para n=4 tem-se o reticulado de quatro dimensões gerado pelos vetores:

$$x^{(1)} = 1/\sqrt{2} \cdot (2,0,0,0);$$

$$x^{(2)} = 1/\sqrt{2} \cdot (1,1,0,0);$$

$$x^{(3)} = 1/\sqrt{2} \cdot (1,0,1,0);$$

$$x^{(4)} = 1/\sqrt{2} \cdot (1,0,0,1).$$

A densidade deste empacotamento reticulado é igual a 0,616186, o número de contatos é 24 e o raio das esferas 0,5.

Definição 2.2.9 [47]: O **arranjo de coordenadas** de um ponto em $\mathbb{R}^n$ que possui coordenadas inteiras é formado colocando-se em colunas os valores das coordenadas do ponto em base binária. A primeira linha do arranjo (de ordem 2^0) compreende os primeiros dígitos das coordenadas e assim possui zeros para as coordenadas cujo valor é um número par e um para aquelas cujo valor é um número ímpar. As 2's,3's,4's, etc. linhas (respectivamente de ordens 2^1,2^2,2^3, etc.) de maneira similar, compreendem os 2's,3's,4's, etc. dígitos das

coordenadas. A notação complementar é usada para inteiros negativos, supondo que o maior valor positivo de uma coordenada é $2^{n-1}-1$.

Exemplo de arranjo de coordenadas.

2	-3	5	0	1	4	1	
0	1	1	0	1	0	1	$2^0 = 1$
1	0	0	0	0	0	0	$2^1 = 2$
0	1	1	0	0	1	0	$2^2 = 4$
0	1	0	0	0	0	0	$2^3 = 8$

Como será feito uso de alguns tipos de construção de reticulados nos Capítulos 3 e 6 deste trabalho, serão, a seguir, apresentadas as definições das construções que serão utilizadas.

Definição 2.2.10 Construção A [12]: Seja $C(2;n,M,d)$ um código binário. Então $\mathbf{x} = (x_1,\ldots x_n)$, de coordenadas inteiras, será o centro de uma esfera, se e somente se $\mathbf{x}$ for congruente (módulo 2) a uma palavra código de C, isto é, se a primeira linha do arranjo de coordenadas de $\mathbf{x}$ pertence a C. O empacotamento esférico de um reticulado é obtido, se e somente se C for um código de grupo.

Exemplo de construção A.

A seguir é mostrado um exemplo de construção A. Neste exemplo a construção A é feita através de um arranjo de coordenadas onde a primeira linha é uma palavra do código $C(2;4,4,2)$ e a segunda linha é uma palavra não codificada.

$$\begin{array}{rl} & \underline{1210} \\ x^0 = & 1010 \quad 2^0 \\ x^1 = & 0100 \quad 2^1 \end{array}$$

A construção A, se aplicada a códigos binários que consistem somente de palavras de peso par, pode gerar reticulados D_n após a multiplicação dos centros das esferas por $1/\sqrt{2}$.

Definição 2.2.11- Construção B [12]: Seja $C(q;n,M,d)$ um código binário com a propriedade de que o peso de cada palavra-código é par. No empacotamento esférico em $\mathbb{R}^n$ desta construção, o elemento $x = (x_1,\ldots x_n)$, é o centro de uma esfera, se e somente se x for congruente (módulo 2) a uma palavra-código de C e $\sum_{i=1}^{n}(x_i)$ for divisível por 4.

Esta construção pode também ser aplicada em uma classe mais ampla de códigos, porém a versão dada pela definição 2.2.11 cobre os casos mais importantes.

Exemplo de construção B.

O exemplo a seguir ilustra uma construção do tipo B. Neste exemplo, a construção B foi obtida utilizando-se um arranjo de coordenadas onde a primeira linha é representada por uma palavra do código de repetição (8,1), a segunda linha é uma palavra do código de Hamming estendido (8,4) e a terceira linha é uma palavra não codificada.

02046602

x^0=00000000
x^1=01001101
x^2=00111100

Neste capítulo foram apresentados os conceitos fundamentais de reticulados e suas construções. Estes conceitos serão de grande valia para o entendimento de sistemas reticulados e seus parâmetros, utilizados como ferramenta de avaliação dos particionamentos de códigos feitos no capítulo 3. Os reticulados apresentam ser de grande utilidade para a análise de sistemas de modulação codificada que utilizam códigos binários ou multiníveis. Isto será constatado nos capítulos 3 e 6 deste trabalho.

CAPÍTULO 3

CONCATENAÇÃO GENERALIZADA

3.1 INTRODUÇÃO.

A técnica de concatenação de códigos foi originalmente proposta por Forney [18] como uma possível alternativa de codificação que pudesse satisfazer os requisitos do teorema de codificação de canal. Assim, desejava-se mostrar a existência de códigos longos, tais que, para uma taxa R constante e menor do que a capacidade do canal C, a probabilidade média de erro do conjunto alcançasse valores tão pequenos quanto desejados. Tinha-se, também, por objetivo, que estes códigos devessem apresentar uma simplicidade quanto ao processo de decodificação quando comparado com a complexidade inerente ao processo de decodificação de códigos de bloco longos sem concatenação.

Em sua forma mais simples, a concatenação é formada por dois códigos: um, geralmente binário e chamado de código interno, e outro, geralmente multinível e chamado de código externo. São usados comumente como códigos internos códigos ortogonais, códigos de bloco de pequeno comprimento e códigos convolucionais.

Este trabalho tem como objetivo o estudo de concatenação generalizada onde códigos de blocos são utilizados tanto no codificador externo quanto no interno.

Antes porém de apresentar-se uma descrição formal da concatenação generalizada, será descrito um sistema de concatenação convencional.

A figura 3.1.1 ilustra um modelo esquemático de um sistema

de concatenação que será denominado convencional.

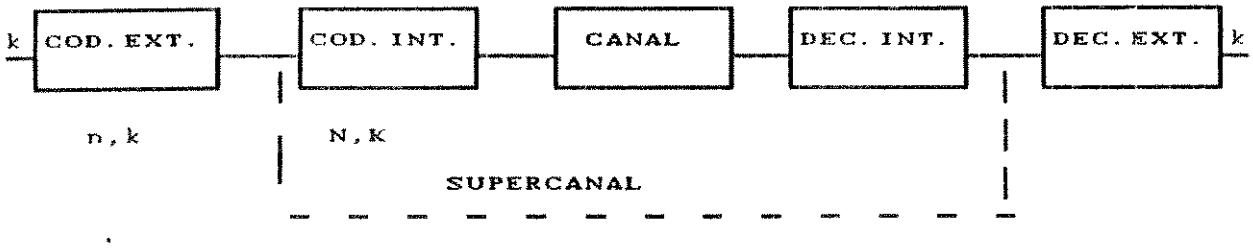


Fig.3.1.1 Concatenação de códigos

Assim um **código concatenado** (ou aninhado ou em cascata) é o código formado pelo "enlaçamento" de dois ou mais códigos. Da figura 3.1.1 pode ser notado o "enlaçamento" entre dois códigos onde o codificador externo (COD. EXT.), formado por um código geralmente longo, não binário e com parâmetros n,k,d, alimenta o codificador interno (COD. INT.) formado geralmente por um código curto, binário e por parâmetros representados por N,K,D. Como resultado tem-se o código concatenado oriundo deste enlaçamento, caracterizado pelos parâmetros $N^{*}= n.N$, $K^{*}= k.K$, D^{*}.

Como, tradicionalmente, o código interno serve para proteger os símbolos do código externo, o conjunto código interno-canal-decodificador interno pode ser visto como um supercanal.

O que realmente acontece, é que o código interno corrige a maior parte dos erros, alterando a distribuição dos mesmos, de tal maneira que o código externo possa facilmente detetar e corrigir (se for o caso) os erros restantes.

Opcionalmente ao sistema visto na figura 3.1.1, pode ser acrescentada uma estrutura de misturadores/desmisturadores ("interleaver/deinterleaver") [8]. Esta estrutura é acrescentada

externamente ao sistema e é utilizada quando os erros ocorrem em surtos ("bursts"). O objetivo é fazer com que a distribuição dos erros seja aleatória para o bom desempenho de certos tipos de decodificadores.

Antes de se iniciar a caracterização do processo de concatenação generalizada, serão apresentadas as definições dos tipos de códigos que serão utilizados. Em primeiro lugar, é apresentada a definição de códigos de bloco.

Definição 3.1.1: Um **código de bloco linear** C(q;n,M,d) é um subconjunto do espaço vetorial $GF(q)^n$ sobre GF(q) de ordem q, comprimento n, cardinalidade M e distância mínima de Hamming d.

No projeto e análise de algoritmos que utilizam técnicas de decisão suave, bem como em sistemas de concatenação de códigos, é importante a representação e mapeamento dos códigos no espaço Euclidiano. Basicamente, um código no espaço Euclidiano $\mathbb{R}^n$ é um conjunto de M pontos $s_1,\ldots,s_M$ em um espaço Euclidiano n-dimensional real $\mathbb{R}^n$ [9].

Para representar um código no espaço Euclidiano, será usada a mesma notação da definição 3.1.1, com excessão da distância d que, no espaço Euclidiano, representa a distância mínima Euclidiana. Por exemplo um conjunto de sinais $s_i \in \mathbb{R}^2$, i=1,...M, ou seja um conjunto de M pontos, define um código C(2;n,M,d) em um espaço Euclidiano, onde a distância mínima d é definida por:

$$d = \min_{\substack{s_i,s_j \in C \\ i \neq j}} \{ dist_E(s_i,s_j) \} \qquad (3.1.1)$$

As propriedades desejáveis de um código no espaço Euclidiano são [9]:

1- O número M de palavras-código deve ser bastante grande;

2- A energia total $\Sigma \| x_i \|^2$ deve ser pequena;

3- A distância mínima entre os vetores-código x_i deve ser grande;

4- Possuir facilidade para encontrar o k-ésimo vetor-código x_k para um dado valor de k;

5- Possuir facilidade em se encontrar o índice k para um dado vetor-código x_k.

6- Possuir facilidade para se encontrar um vetor-código x_k o mais próximo possível de um ponto $z \in \mathbb{R}^n$ dado de maneira arbitrária.

3.2 CONCATENAÇÃO GENERALIZADA.

A concatenação generalizada de códigos (CGC) ([4], [56], [57] e [58]), tem por objetivo básico abranger todas as categorias possíveis de concatenação. Como exemplo, pode-se citar a concatenação código-código, a concatenação código-modulação e a concatenação código-modulação-modulação. Neste aspecto, a forma combinada de modulação codificada (MC) passa a ser um caso particular de concatenação generalizada. O objetivo básico da concatenação generalizada reside no particionamento de conjuntos, podendo estes ser códigos, modulações, reticulados, etc. Assim sendo, a concatenação generalizada pode ser dividida em dois grandes grupos, a saber :

1- Concatenação código/modulação que envolve as teorias e técnicas de codificação e modulação atuando juntas ou separadamente;

2- Concatenação de códigos, onde somente são analisados os códigos concatenados sob a óptica das propriedades dos códigos [5]. .

Em ambos os grupos, a teoria de reticulados é uma ferramenta indispensável. Isto ficará mais claro à medida que os conceitos forem sendo introduzidos.

Como foi mencionado, a concatenação generalizada está fundamentada no particionamento de conjuntos. Para o caso específico destes conjuntos serem representados por códigos, faz-se necessária a introdução de alguns conceitos fundamentais para a compreensão dos capítulos seguintes.

Particionamento de códigos: Seja um código $B^{(1)}$ representado por um conjunto de M pontos de um espaço n-dimensional. A distância mínima entre os vetores $b \in B^{(1)}$ é $d^{(1)}$. O conjunto $B^{(1)}$ pode ser dividido em s subconjuntos disjuntos $B_i^{(2)}$, tais que:

$$B^{(1)} = \bigcup_{i=1}^{s} B_i^{(2)} \qquad (3.2.1)$$

A **distância mínima** de cada subconjunto $B_i^{(2)}$ é $d_i^{(2)} \geq d^{(1)}$, $i=1,2,\ldots,s$. O código $B^{(1)}$ é dito particionado em s subconjuntos $B_i^{(2)}$ e a distância mínima dos subconjuntos é definida como:

$$d^{(2)} = \min_{i=1,\ldots,s} \{ d_i^{(2)} \} \qquad (3.2.2)$$

É claro que este particionamento pode ser feito de diferentes maneiras. Entretanto, o interesse maior está relacionado com o caso específico de particionamento que conduz a subcódigos onde a distância entre as palavras pertencentes a cada subcódigo é a maior possível.

Assim,

Definição 3.2.1: Um particionamento é dito ótimo para um número fixo de subcódigos, se a distância mínima dos subcódigos $d^{(2)}$ é a máxima quando comparada com todos os possíveis particionamentos do código $B^{(1)}$.

Para códigos de baixa cardinalidade, um método para a obtenção de particionamento de um código $B^{(1)}(q;n,M^{(1)},d^{(1)}) \leq GF(q)^n$ é dividir as palavras-código de $B^{(1)}$em sub-códigos, procurando fazê-la com o particionamento ótimo. Com o uso de técnicas e algoritmos mais sofisticados, o particionamento ótimo pode ser também obtido para códigos de maior cardinalidade.

Um dos métodos existentes de particionamento e que será visto aqui é o seguinte [5]:

Seja um código $B^{(1)}(q; n, M^{(1)}, d^{(1)}) \leq GF(q)^n$.

A partição de $B^{(1)}$ em s subcódigos $B_i^{(2)}(q; n, M_i^{(2)}, d_i^{(2)})$, i = 1, 2, ...s é obtida através de um procedimento de divisão tal que :

$$\text{Se} \quad B^{(1)} = \bigcup_{i=1}^{s} B_i^{(2)} \quad , \quad \text{então } M^{(1)} = \sum_{i=1}^{s} M_i^{(2)} \qquad (3.2.3)$$

e a distância mínima de Hamming $d^{(2)}$ é definida por:

$$d^{(2)} = \min_{i=1,\ldots s} \{ d_i^{(2)} \}.$$

Dada uma palavra-código $b \in B^{(1)}$, o subcódigo $B_i^{(2)}$, ao qual b pertence, é unicamente determinado.

Note-se que, até agora, as distâncias envolvidas nos níveis de partições foram as **distâncias de Hamming**. Isto significa que foi utilizada a **concatenação do tipo código-código**. Como exemplo trivial deste particionamento, seja C o seguinte código mostrado na fig.3.2.1.

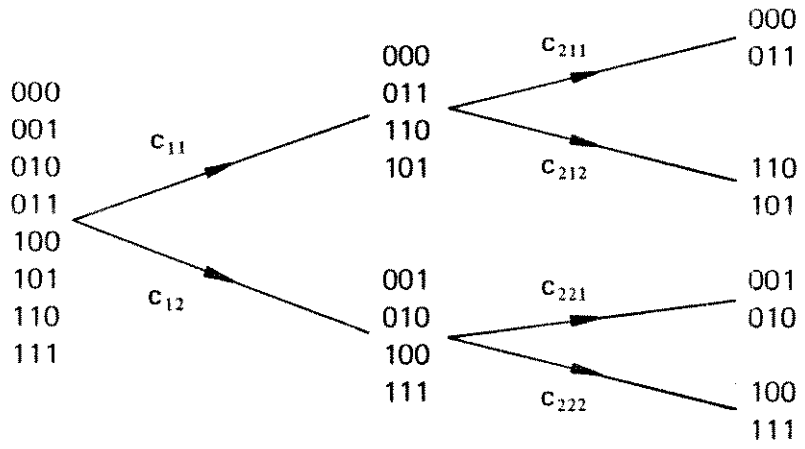


Fig. 3.2.1 Particionamento do código C.

A primeira partição é realizada mediante a escolha de $d_{11} = d_{12} = 2 > d = 1$, o que resulta nos subconjuntos C_{11} e C_{12}. A segunda partição dos C_{1i} se verifica através de $d_{11j}=d_{12j}=2=d_{11}=d_{12}$, o que leva aos subconjuntos C_{2ij}, como mostra a figura 3.2.1.

A seguir, visando uma conceituação gradual de concatenação

generalizada, será inicialmente abordado um sistema de concatenação de códigos convencional. Como mencionado anteriormente, um sistema de códigos concatenados é constituído por dois códigos: um interno, geralmente binário, e um externo geralmente multinível. Como código externo, quase sempre é utilizado o código de Reed-Solomon (RS), pelas suas reconhecidas propriedades de corretor de surtos de erros, distância máxima separável, etc. Como exemplo desta descrição, utiliza-se um código de Reed-Solomon como código externo e um código de um único dígito de verificação de paridade como código interno.

Exemplo:

Seja um código de RS C(4;3,16,2) sobre GF(4) num espaço tridimensional com distância de Hamming mínima igual a 2 e especificado pelos elementos de GF(4), dados por ({0,1,α, $\beta = \alpha^2$}, $\alpha^2+\alpha+1=0$, $g(x)=x-\alpha$). As 16 palavras-código são:

000	1α0	β0α	$\beta\alpha$1
01α	$\alpha\beta$0	10β	111
0$\alpha\beta$	β10	1$\beta\alpha$	$\alpha\alpha\alpha$
0β1	α01	α1β	$\beta\beta\beta$

Para facilidade notacional e de entendimento, será empregada a seguinte transformação de GF(4) para os elementos de $GF(2)^2$.

$0 \rightarrow 00$; $1 \rightarrow 10$; $\alpha \rightarrow 01$; $\alpha^2 \rightarrow 11$; Assim de acordo com a representação utilizada em [31], as palavras-código serão convertidas para:

000	010	101	110	001	110	001	000
000	100	100	101	010	010	101	111
011	100	011	111	010	100	101	111
001	110	110	000	011	001	011	111

Para a efetivação da concatenação, será utilizado um código de um único dígito de verificação de paridade em cada palavra do código de RS. Acrescentar um dígito de verificação de paridade é equivalente a acrescentar uma linha do arranjo acima, um 1 na coluna de paridade ímpar e "0" nas demais posições. Note que i é a numeração das linhas com i=0,1,2,...,m. Como exemplo, seja:

010

100

a palavra-código do código de RS. A verificação de paridade será, então, dada por:

650

010

100

110

Para facilitar o processo de decodificação, será adicionada também verificação de paridades de linhas.
Assim, do exemplo anterior tem-se:

		n'	
seqüência-código		650	3
k	{	010	1
	{	100	1
		110	0

É fácil ver que a soma de toda seqüência-código possui valor par e, com exceção das seqüências de repetição (333,555,666,000), todas as demais seqüências possuem soma igual a 14 como no exemplo dado, onde a seqüência-código é 6503. Portanto, ao adicionar um bit de verificação de paridade em cada coluna do código original (externo), obtém-se um código binário C[n'(k+1),kk', 2d'], [31,pag.590]. Esta é a forma mais simples de concatenação de códigos. A distância mínima de C é pelo menos 2d', porque existe pelo menos d' colunas diferentes de zero, cada uma de peso ≥ 2.

Kasahara e outros em [31] observaram que é quase sempre possível incrementar em 1 a dimensão de C. O novo código C' é constituído por palavras-código de C juntamente com as palavras-código formadas pela linha de verificação de paridade. Desta forma, tem-se que este novo código C' possui os seguintes parâmetros:

$$[n,k,d] = [\, n'(k+1),\ kk'+1,\ 2d' \,], \text{ dado que } n' \geq 2d' \qquad (3.2.4)$$

O código obtido, neste exemplo, através da concatenação de um código RS como código externo e um código binário de um dígito de verificação de paridade de colunas como código interno, pode ser visto como um reticulado do tipo D_4 gerado pelo código binário de um dígito de verificação de paridade $C(2;4,2^3,2)$. Consequentemente, este reticulado é gerado pelo agrupamento das

palavras do código C' , formando subconjuntos com propriedades específicas, algumas das quais serão mencionadas adiante.

O reticulado do tipo D_n (vide cap. 2) é obtido pela construção A, sendo que os pontos deste reticulado são vértices alternados do reticulado cúbico Z^n. Como existem n (n-1)/2 palavras-código de peso 2, tem-se:

$$\rho=1/\sqrt{2},\ \delta=2^{-(n+2)/2} \text{ e } \tau = 2n(n-1) \qquad (3.2.5)$$

Para a construção de reticulados do tipo $E_8=\Lambda_8$ pode-se utilizar um subconjunto do código C. Por exemplo, pode-se fazer uso somente das palavras de peso mínimo do código. Usando como base o código de Hamming C(2;7,16,3), tem-se:

1365240	7
1101000	1
0110100	1
0011010	1
1000110	1

Fig. 3.2.2 Geração de reticulados.

Ao se acrescentar uma coluna unitária, obtém-se o código de Hamming estendido e um reticulado do tipo E_8. Este reticulado obtido por esta construção caracteriza-se por possuir a seqüência de 0 a 7 sem repetição (com exceção das seqüências de símbolos iguais), de acordo com a fig. 3.2.2. O reticulado obtido com o código C(2;8,16,4) é o empacotamento mais denso em R^8 [12] com

$$\delta = 2^{-4} \text{ e } \tau = 2.8+14.16=240 \qquad (3.2.6)$$

Note que, no processo de concatenação apresentado, o interesse está voltado, basicamente, para a obtenção de matrizes que venham a facilitar o processo de decodificação. Estas matrizes serão vistas mais adiante (equação 3.2.7). Isto é equivalente a obter códigos **binários** e **multiníveis** com propriedades específicas. Um exemplo de um código deste tipo é o C(2;5,8,2) definido pela seguinte matriz geradora:

$$G = \begin{array}{c|c|c} \begin{matrix} 1 & 2 & 4 & 5 & 6 \end{matrix} & \multicolumn{1}{|c}{} & 18 \\ \hline \begin{matrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{matrix} & & \begin{matrix} 18 \\ 9 \\ 7 \end{matrix} \end{array}$$

Pode-se verificar facilmente que todas as palavras deste código são múltiplos de 7 ou de 9. Esta propriedade, como será visto mais adiante na concatenação generalizada, facilita a decodificação das palavras recebidas.

A partir deste ponto tem-se o interesse em estabelecer o conceito de Concatenação Generalizada sem que haja uma conotação puramente abstrata. Isto será feito através de um exemplo. Para tanto, são utilizados os seguintes códigos externos

$A^{(1)}$ $(2^2; {}^{n}_{8}, {}^{m}_{4}, {}^{d}_{8})$-código de repetição em $GF(2^2)$-RS;

$A^{(2)}$ $(2;8,2^4,4)$ -código de Hamming estendido em GF(2)

Como código interno é utilizado o código de um digito de verificação de paridade C(2;4,8,2). Desse modo, o código

concatenado resultante é:

$$C_c(2;32,2^6,16),$$

que é um código de Reed-Muller.

A fig. 3.2.3 ilustra este sistema de concatenação generalizada. Basicamente o processo de concatenação para este exemplo consiste em se particionar o código interno em quatro subcódigos da seguinte maneira:

	0	1
0	0000	1111
1	0011	1100
2	0101	1010
3	0110	1001

É importante observar que, com este particionamento os subcódigos obtidos possuem o dobro da distância mínima de Hamming do código original. As palavras-código de cada subcódigo podem ser facilmente identificadas através do sistema de indexação mostrado. Desta maneira a palavra-código 1111 do primeiro subcódigo pode ser identificada pelos índices (0,1). O papel de índices será feito pelos códigos externos.

Assim, o primeiro índice será caracterizado pelo código de repetição, cabendo ao código de Hamming estendido apontar o segundo índice. De acordo com esta indexação, cada palavra-código obtida por esta concatenação, fará uso somente de combinações de um subcódigo de cada vez.

Desta forma, será criada uma matriz formada pelos códigos externos que mapearão o código interno de acordo com a tabela anterior. A matriz que fará o mapeamento será chamada de matriz

mapeadora e é dada por:

$$\begin{vmatrix} a_1^{(1)} a_1^{(2)} \\ a_1^{(1)} a_2^{(2)} \\ \cdot \ \cdot \\ \cdot \ \cdot \\ a_1^{(1)} a_8^{(2)} \end{vmatrix} \qquad (3.2.7)$$

onde $a_i^{(1)}$ e $a_j^{(2)}$ são palavras dos códigos externos de RS e de Hamming, respectivamente. A matriz-código obtida, ou seja a palavra-código em forma de matriz, será chamada de matriz mapeada.

A fig. 3.2.3 apresenta o esquema de codificação realizada na concatenação generalizada.

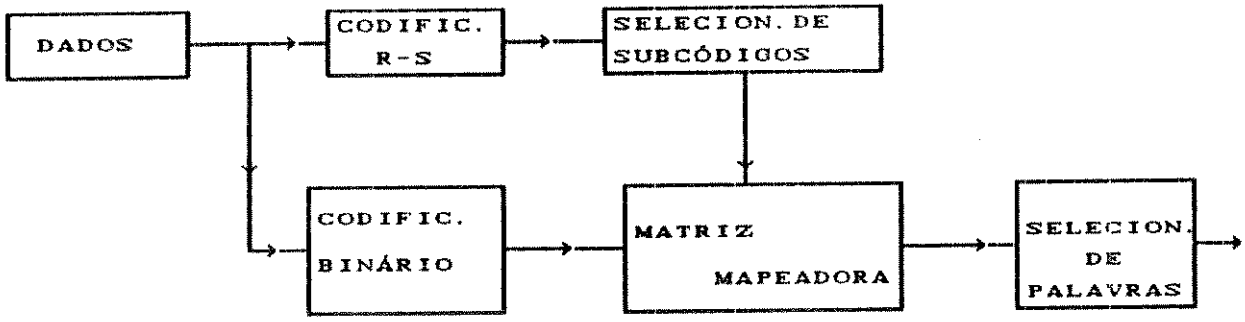


Fig. 3.2.3 - Codificador de concatenação generalizada.

Basicamente, o ganho de um código concatenado pode ser analisado como o ganho de dois reticulados: um para o código interno e outro para os códigos externos envolvidos. A maneira como estes ganhos estão associados será vista mais adiante.

Aplicando a construção A para os códigos externos $A^{(1)}$ e $A^{(2)}$, tem-se um reticulado associado do tipo $H_8 = E_8$ com os

seguintes parâmetros:

$$\delta = 1/16 = 0{,}0625,$$
$$\tau = 2.8 + 14.16 = 240 \qquad (3.2.8)$$
$$\text{e} \quad \rho = 0{,}5 \min \{ 2, \sqrt{d} \} = 1.$$

O reticulado $H_8 = E_8$ obtido pela construção A é o mais denso em R^8. Por outro lado, se for utilizada a construção do tipo B para os códigos externos, $A^{(1)}$ e $A^{(2)}$, tem-se os seguintes parâmetros associados:

$$\rho = 0{,}5 \min\{\sqrt{d}, \sqrt{8}\} = 1$$
$$\delta = M.\rho.2^{-\tau/4-1} = 0{,}03125\ , \qquad (3.2.9)$$
$$\tau = 240.$$

Como pode ser observado através destes parâmetros, o particionamento obtido é ótimo, porém não possui propriedades que facilitem a utilização dos algoritmos utilizados em [9] e [53], ou seja a palavra-código, obtida em forma de matriz não possui a primeira coluna da esquerda ou da direita como sendo uma palavra-código de um código de repetição binário. Se tal ocorresse, poder-se-ia, através da rotação da matriz-código, obter uma matriz análoga à obtida por Williams e Farrell [53].

A seguir, é analisado um esquema de particionamento sub-ótimo, onde o ganho do código externo é maior em detrimento do ganho do código interno. Este particionamento proporciona um ganho menor do que o obtido pelo particionamento ótimo, trazendo porém, como fator positivo as propriedades conseguidas em [9] e [53] para uma maior facilidade de decodificação.

	0	1
0	0000	0101
1	0011	0110
2	1111	1010
3	1100	1001

Este particionamento fornece agora um código de repetição C(2;8,2,8) para a primeira e a terceira colunas da matriz mapeada. Assim sendo, pode-se afirmar que se tem um código de repetição para duas quaisquer colunas da referida matriz mapeada, por exemplo para a primeira e a segunda colunas.

Aplicando a construção do tipo B no código de repetição, tem-se:

$$\rho = 0,5\min(\sqrt{d},\sqrt{8}) = \sqrt{2}$$
$$\delta = M.\rho^{n}.2^{-n-1} = 0,0625 \qquad (3.2.10)$$
$$\tau = 240$$

Desta forma, neste caso, o particionamento sub-ótimo apresenta maiores facilidades para a decodificação por decisão suave por possuir um raio de empacotamento maior para a mesma densidade dos particionamentos.

Ao passar-se do primeiro tipo de particionamento (ótimo) para o segundo, altera-se completamente o código. O novo código concatenado possui agora um código interno de distância mínima de Hamming igual a 2. Devido aos códigos externos, duas das colunas da matriz mapeada possuem peso igual a 8 (código de repetição) e duas colunas da mesma matriz possuem peso mínima igual a 4

(código de Hamming estendido). Sendo a distância mínima de Hamming do código concatenado igual a:

$$d_{min} \geq \min_{i=1,\ldots,s} (d_b^{(i)} . \ d_a^{(i)}), \ i=1,2,\ldots,s \qquad (3.2.11)$$

onde d_b é a distância mínima de Hamming do código interno e d_a é a distância mínima de Hamming para o código externo e s o número de particionamentos, tem-se:

a) primeiro tipo de particionamento:

$$d_1 \geq \min (2.8, 4.4) = 16 \qquad (3.2.12)$$

b) segundo tipo de particionamento:

$$d_2 \geq \min (2.8, \ 2.4) = 8 \qquad (3.2.13)$$

Dando prosseguimento à análise do segundo tipo de particionamento, serão estudadas as propriedades da matriz-código (matriz mapeada).

Esta matriz apresenta as seguintes propriedades:

1- A soma de todas as linhas é um número divisível por 8;

2- Os números representativos das linhas são ou todos pares ou todos ímpares em cada matriz.

Como resultado destas propriedades para a decisão suave, os algoritmos apresentados em [9], [12] e [53] podem ser utilizados. Para decisão abrupta, se 5 bits da primeira coluna da matriz mapeada, ou seja, da palavra-código obtida pela concatenação, estiverem errados, então o algoritmo utilizado no particionamento sub-ótimo cometerá um erro.

Para a decodificação é usada a seguinte estrutura para o

mapeamento da matriz recebida:

	0	1	*
0	a	a'	
1	b	b'	
2	c	c'	
3	d	d'	

Esta estrutura é semelhante à estrutura utilizada para o particionamento do código interno. O sinal * representa palavras recebidas que não pertencem a nenhum dos 4 subcódigos.

Tem-se, então, os seguintes passos:

- É feita uma contagem das palavras dos vários subcódigos.

Por exemplo:

		*
4	3	

Este exemplo indica que foram recebidas 4 palavras c, 3 c' e uma que não pertence a nenhum subcódigo. Neste passo, pode ser detetado se a matriz recebida está isenta de erros. Deteta-se, também, qual o subcódigo utilizado na formação da matriz.

Se houver uma palavra *, ela será decodificada como uma das faltantes de um subcódigo. Se houver duas palavras *, elas serão decodificadas como sendo palavras do subcódigo em minoria. Em caso de empate, decodificam-se as duas palavras * juntas para qualquer uma das palavras do subcódigo. As demais composições da matriz recebida serão analisadas e decodificadas de maneira que,

se necessário, possam ser decodificadas com o auxílio do código C(2;8,16,4).

Pode ser visto que, apesar das distâncias dos subcódigos serem otimizadas, este algoritmo não é eficiente.

Desta forma o segundo tipo de particionamento contém maiores facilidades para a decodificação com decisão suave por possuir propriedades análogas às utilizadas pelo algoritmo de decodificação proposto por Conway e Sloane [9] para reticulados. Para decisão abrupta, para o caso em que por exemplo, 7 bits da primeira coluna estejam errados o algoritmo utilizado com o segundo particionamento fará um erro.

Uma maneira de contornar este problema é a utilização de particionamentos alternados. Este particionamento envolve a codificação de duas palavras, sendo que, nas matrizes mapeadas, a primeira coluna é alternadamente toda zero ou toda um.

part. A		part. B	
0000	0101	1111	1100
0011	0110	1001	1010
0000	0011	1111	1010
0101	0110	1100	1001

Neste caso, as palavras-código devem ser decodificadas duas a duas, pois, somente desta forma, tem-se o código interno completo. A localização do subcódigo, também, depende das duas palavras adjacentes. Assim, se, seqüencialmente, em duas palavras-código não ocorrerem mais de 4 erros na primeira coluna, então, todos os padrões de 8 erros serão decodificados. Diante do

exposto, no ítem 3.7 será apresentado um algoritmo de decodificação suave parcial para utilização em sistemas de concatenação generalizada. Este algoritmo utiliza as propriedades dos códigos interno e externo e, portanto, fica condicionado à escolha e à análise da estrutura dos mesmos.

3.3 CONCATENAÇÃO CONVENCIONAL (Código-código).

A diversidade de esquemas de concatenação é bastante grande [6] assim como o enfoque teórico dado ao assunto [4],[20], [30], [38] e [39]. Apesar disto, existem determinados esquemas mais usuais, que serão resumidamente vistos a seguir.

3.3.1 Reed Solomon/Código ortogonal.

Neste esquema, é usado o código RS como código externo e, como código interno, os códigos ortogonais $C(2;2^k,M,d)$ ou os códigos biortogonais $C(2;2^{k-1},M,d)$. Estes códigos internos são caracterizados por possuírem baixas taxas e por isso são usados em sistemas de faixa larga como, por exemplo, em comunicações espaciais.

3.3.2 Reed Solomon/Códigos de bloco binários.

Os códigos de bloco curtos são usados como códigos internos, quando se deseja taxas R^* entre 1/4 e 3/4. O desempenho do código interno influi fortemente no desempenho do código concatenado. Assim para o código interno são utilizados os melhores algoritmos de decodificação. Em geral, estes algoritmos fazem uso de técnicas de decisão suave. Quando possível, os códigos internos são decodificados através de um algoritmo de máxima verossimilhança, conduzindo a resultados excelentes. Em geral,

são utilizados códigos com $4 \leq n \leq 14$ e $3 \leq k \leq 5$. Para códigos muito curtos, $k=3;4$, a decodificação de máxima verossimilhança (MLD) é obtida pela correlação total das palavras do código. Para comprimentos maiores, algoritmos de máxima verossimilhança ou assintóticos ao MLD são requisitados.

Como exemplo de concatenação convencional muito utilizado em sistemas de comunicações, tem-se como código externo o código de Reed-Solomon (15,11) com símbolos pertencentes a $GF(2^4)$ e o código binário interno de Hamming (7,4). Cada símbolo do código externo representado por 4 bits do alfabeto do código interno. Como resultado, tem-se o código concatenado binário (105,44) com

$$N^* = n.N = 15.7 = 105 \text{ bits},$$
$$K^* = k.K = 11.4 = 44 \text{ e}$$
$$R^* = rR = kK/nN = 44/105 \text{ e}$$
$$d_{min} = d_1.d_2 = 3.5 = 15 \qquad (3.3.1)$$

3.3.3 Códigos concatenados que usam matrizes especiais.

Na categoria destes códigos estão incluídos os códigos de matrizes do tipo *B* [51], isto é, códigos fundamentados em matrizes que usam dupla verificação de paridade: em linhas ou colunas e em diagonal.

Neste tipo de concatenação, cada bit de informação é dupla ou triplamente verificado pelos bits de paridade. Estes códigos são usados em canais com memória (ou seja, onde os erros ocorrem em "bursts", "clusters") ou em canais onde os erros são aleatórios.

3.4 CONCATENAÇÃO CÓDIGO/MODULAÇÃO.

3.4.1 Códigos de Sayegh.

Sayegh [44] propôs a construção de arranjos constituídos por códigos de bloco com o objetivo de obter resultados análogos àqueles obtidos por Ungerboeck [50] para códigos convolucionais.

No esquema apresentado por Sayegh utiliza-se uma matriz construída a partir de vários códigos binários.

Será considerado em seguida um exemplo para um sistema modulado em 8-PSK, onde é proposta a montagem de um arranjo constituído por 3 códigos binários independentes especificados por (n,k_i,d_i), para $i = 1,2,3$. Neste exemplo o arranjo será composto da seguinte maneira:

00000000	C_1 (2;8,2,8)
01001101	C_2 (2;8,16,4)
00011100	C_3 (2;8,256,1)

Todas as "matrizes-código" serão formadas pelas combinações das palavras dos códigos mencionados acima. Cada coluna do arranjo corresponde a um ponto no espaço de sinais. O arranjo (matriz-código) será transmitido de uma coluna por vez.

Para este exemplo, o esquema proposto é uma construção típica de reticulados usando a construção B (vide cap. 2).

De acordo com construção proposta por Sayegh tem-se por objetivo, neste exemplo, escolher k_1, k_2 e k_3 de maneira a maximizar a distância Euclidiana entre as palavras (matrizes-código) do código formado.

Para o cálculo da distância Euclidiana mínima entre dois pontos deste reticulado obtido pelo mapeamento destas "matrizes-código" será utilizada a fig. 3.4.1.

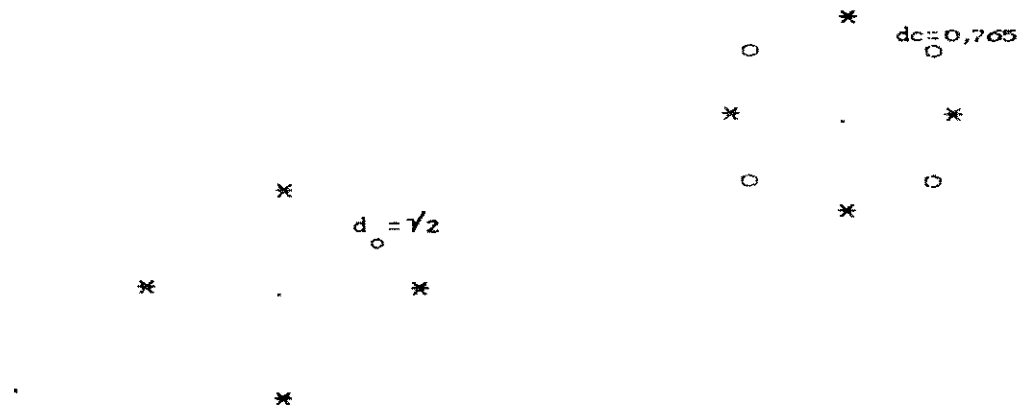


Fig. 3.4.1 Constelações 4PSK e 8PSK.

A distância Euclidiana D entre duas "matrizes-código" que difiram na primeira linha, deverá ser, no mínimo,

$$D^2 => (0,765)^2 . d_1 = 0,586\ d_1 \qquad (3.4.1)$$

pois d_1 é a distância mínima de Hamming do código C_1 que forma a primeira linha e os símbolos do reticulado tem distância igual a 0,765.

Caso as matrizes coincidam na primeira linha e divirjam a partir da segunda, tem-se que:

$$D^2 => 2\ d_2 , \qquad (3.4.2)$$

uma vez que o segundo código C_2 possui distância mínima d_2 e os símbolos transmitidos têm distância Euclidiana mínima igual a $\sqrt{2}$, equivalente ao primeiro nível da partição. Note-se que, como houve coincidência na primeira linha, todos os símbolos pertencem a um mesmo subconjunto do primeiro nível da partição.

Dando prosseguimento a esta análise conclui-se que:

$$D^2min = min \langle 0,586\ d_1, 2d_2, 4d_3 \rangle \qquad (3.4.3)$$

Para o exemplo dado tem-se então:

D^2=min [0,586.8, 2.4, 4.3] = 4,688

O ganho assintótico de codificação para este exemplo é calculado da seguinte maneira:

$$GANHO = 10 \log [\ d_c^2/P_c \times P_o / d_o^2\ ] \qquad (3.4.4)$$

onde, de acordo com a figura 3.4.1, d_c e P_c são as distância e potência média respectivamente da constelação codificada (8PSK) e d_o e P_o são os mesmos parâmetros para a constelação sem codificação (4PSK) . Assim tem-se:

Ganho= 10 log (4,688/1. 1/ 2)= 3,7dB

3.4.2 Sistema de modulação codificada de Forney.

Forney [19] propôs um esquema de modulação codificada bastante eficiente, utilizando códigos de bloco . O modelo deste esquema é mostrado na fig.3.4.2.

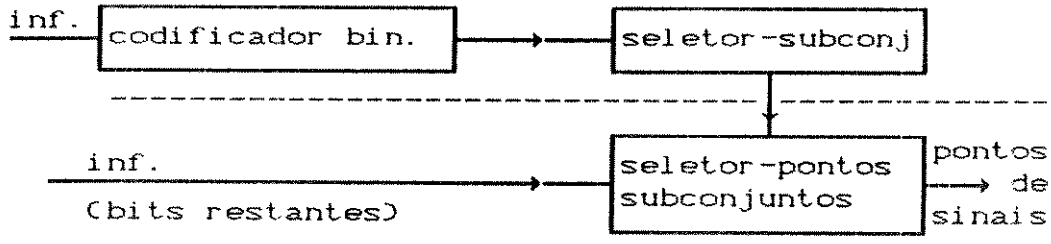


Fig.3.4.2 Sistema de modulação codificada

Neste esquema um bit de informação é utilizado para particionar o reticulado Z^2 em duas classes laterais contendo a mesma constelação (equivalente ao reticulado D_n). Este processo é típico de construção de reticulados simples, em geral com pequena densidade de empacotamento. A proposta apresentada em [19] para a construção de reticulados densos se faz via emprego de códigos de bloco específicos de acordo com a dimensão desejada do reticulado. Por exemplo, o reticulado que apresenta o empacotamento mais denso em 8 dimensões é $\Lambda_8 = E_8$. A construção deste reticulado se processa com o produto cartesiano do reticulado Z^2 quatro vezes, de tal forma que cada componente do vetor seja especificado como os dígitos da palavra-código de um código de um único dígito de paridade C(2;4,8,2).

Deste modo, o processo de codificação combinada com modulação é essencialmente descrito como segue:

Para enviar, por exemplo, m.N = 16 bits (m bits/dimensão em N dimensões), 1 bit é utilizado para particionar o reticulado Z^2 em duas classes laterais, em que cada elemento destas classes consiste em uma constelação de 32 pontos; 3 bits são codificados através do código C(2;4,8,2) e a palavra-código resultante especificará os sub-índices do particionamento do conjunto original para cada uma das quatro componentes do vetor de 8 dimensões; e 4 conjuntos de 3 bits cada são utilizados para identificar cada ponto de sinal pertencente ao subconjunto representativo. Assim, a taxa efetiva do processo de codificação é 16/17. O ganho assintótico alcançado por este esquema é de 3 dB.

Das figuras 3.2.3 e 3.4.2 pode-se notar a diferença básica entre o codificador da concatenação generalizada e o codificador da modulação codificada proposto por Forney. Esta diferença

reside no bloco "codificador binário ", onde os bits são codificados na concatenação generalizada, ao passo que os mesmos não são codificados na modulação codificada.

3.4.3 Esquema de decodificação de Imai-Hirakawa.

Em [29] Imai e Hirakawa propuseram um método de codificação e decodificação para códigos multiníveis. Na codificação, o código multinível é construído com o auxílio de vários códigos binários e não apresenta nenhuma inovação em relação ao método de passos múltiplos. A figura 3.4.3 ilustra o diagrama em blocos do codificador proposto em [29].

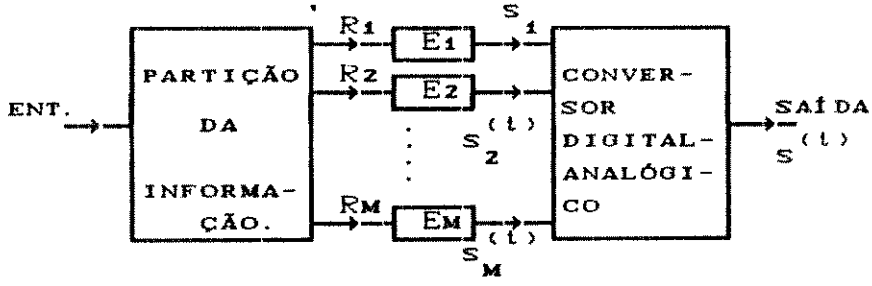


Fig.3.4.3 Diagrama em blocos do sistema de codificação de Imai-Hirakawa.

O circuito codificador consiste de M codificadores $E_1,\ldots,E_M$, que fazem uso de M códigos binários $C_1,\ldots C_M$ com taxas $R_1,\ldots R_M$, respectivamente.

Primeiramente a seqüência de informação é particionada em M blocos de informação de acordo com os códigos binários utilizados. Após a codificação as mensagens $s_1^{(t)},\ldots,s_M^{(t)}$ são combinadas pelo conversor digital-analógico que calcula a seqüência de saída $s^{(t)}$ do codificador como :

$$s^{(t)} = \sum_{i=1}^{M} s_i^{(t)} \cdot 2^{i-1}, \qquad (3.4.5)$$

sendo, desta maneira $s^{(t)}$ um símbolo M-ário.

A originalidade do esquema de Imai-Hirakawa está na decodificação. Para facilidade de exposição, é apresentado na figura 3.4.4 um diagrama de blocos simplificado do circuito de decodificação para um código multinível representado por 4 códigos binários (M = 4).

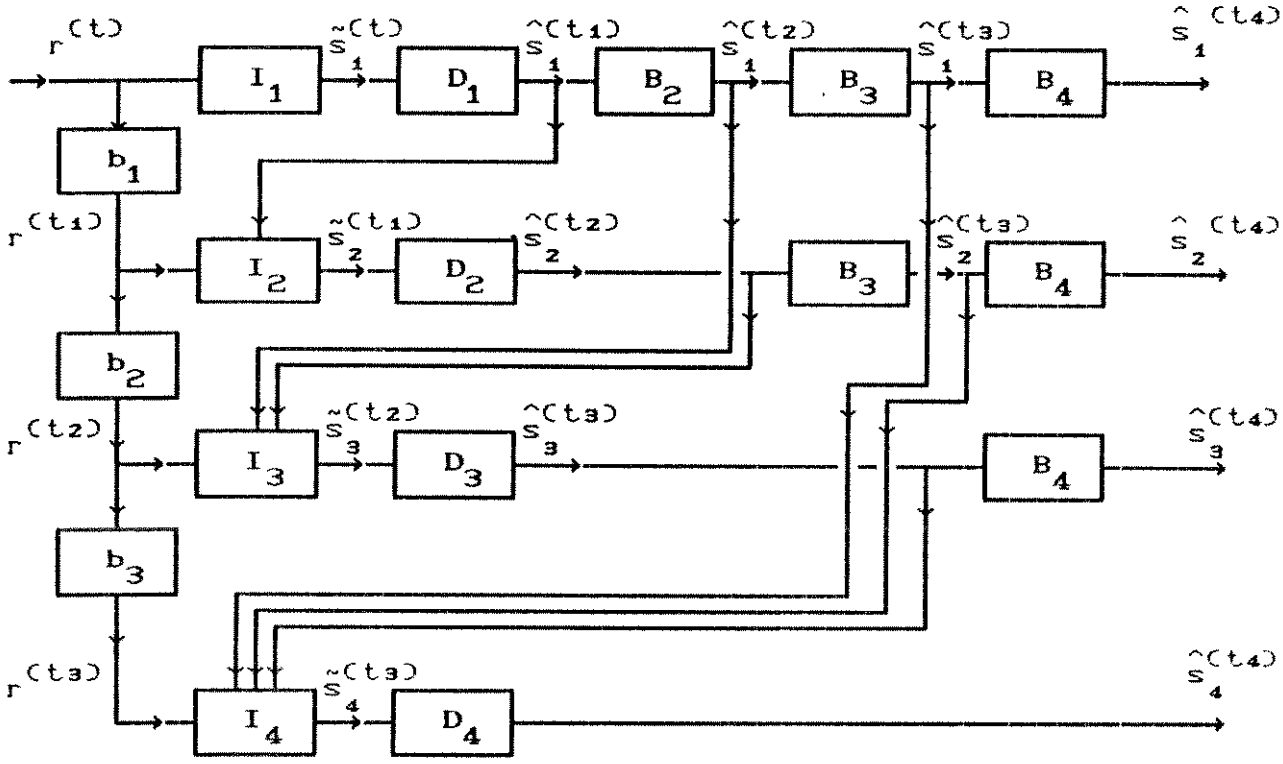


Figura 3.4.4 Decodificador de Imai-Hirakawa para M = 4

Nesta figura é utilizada a seguinte notação:

d_i - atraso de decodificação para c_i.

I_i - estimativa intermediária do circuito para $\tilde{s}_i$.

D_i - decodificador de c_i.

B_i - d_i-unidade de tempo do buffer.

b_i - d_i-unidade de tempo do buffer.

O algoritmo de decodificação pode ser resumido da seguinte maneira:

1-A estimativa intermediária de s_1, $\tilde{s}_1$, é calculada com a utilização das probabilidades a posteriori $P_1(0/r)$ e $P_2(1/r)$. Em seguida, a estimativa intermediária é processada no decodificador D_1 do código C_1, o qual fornece a estimativa final da seqüência $(\hat{s}_1)$, através da correção dos erros da estimativa intermediária.

2-No i-ésimo passo (i=2,...,M): $\tilde{s}_i$ é calculada através das probabilidades $P_i(0/\hat{s}_1 \ldots \hat{s}_{i-1} r)$ e $P_i(1/\hat{s}_1 \ldots \hat{s}_{i-1} r)$. Em seguida, $\tilde{s}_i$ entra no decodificador D_i do código C_i o qual fornece a seqüência $(\hat{s}_i)$ através da correção dos erros em $(\tilde{s}_i)$.

De acordo com o sistema proposto por Imai e Hirakawa, pode ser observado que, se o primeiro decodificador obtiver sucesso na decodificação então o segundo decodificador terá maior probabilidade de obter sucesso, visto que a estimativa intermediária de s_1, $\tilde{s}_1$, está correta. Da mesma forma o sucesso da decodificação do terceiro decodificador depende da decodificação do segundo decodificador e assim por diante.

Desta exposição torna-se evidente que o esquema proposto por Imai-Hirakawa possui melhor desempenho quando os códigos binários utilizados obedecem ao princípio de Ungerboeck da mesma maneira que os códigos de Sayegh.

3.5 CONCATENAÇÃO CÓDIGO-MODULAÇÃO-MODULAÇÃO.

Este tipo de concatenação envolve as técnicas de modulação conjugadas com o particionamento de conjuntos. Um exemplo típico deste tipo de concatenação é o esquema de modulação proposto por Padovani e Wolf [37]. Em [37], Padovani e Wolf tiveram como objetivo apresentar uma generalização dos códigos de Ungerboeck

utilizando modulação em freqüência e fase.

Em [37] a codificação é feita usando códigos convolucionais e a decodificação proposta utiliza o algoritmo de Viterbi.

A regra de mapeamento deste esquema é bem ilustrada em [37] através de um exemplo e que, para facilidade de entendimento, será reproduzido em seguida. O código convolucional possui taxa R=2/3, e a constelação de referência é 4PSK, sendo o conjunto de sinais expandido representado por uma constelação 8PSK.

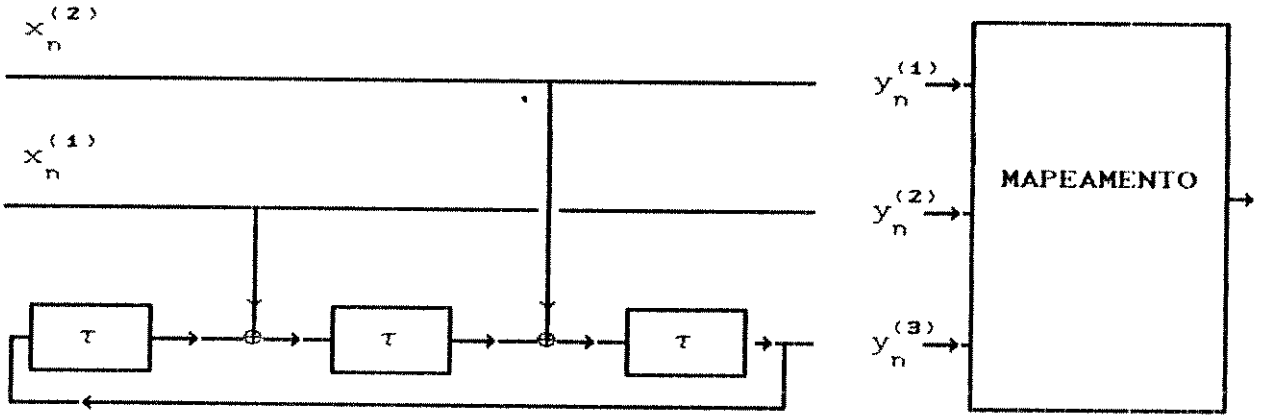


Figura 3.5.1 Esquema e Padovani e Wolf. Codificador convolucional sistemático com realimentação e mapeamento

De acordo com a regra de mapeamento, um dos três bits na saída do codificador convolucional é usado para selecionar uma das duas freqüências possíveis. Com isto é obtida a modulação FSK. Os dois bits restantes são usados para selecionar uma das 4 fases da freqüência correspondente. Obtém-se, assim, o esquema de modulação codificada 2-FSK/4PSK.

$y_n^{(2)}$	$y_n^{(1)}$	$y_n^{(0)}$	sinal	fase	freqüência
0	0	0	s_0	0	$f_c + f_d$
0	0	1	s_1	$\pi/4$	$f_c - f_d$
0	1	0	s_2	$\pi/2$	$f_c + f_d$
0	1	1	s_3	$3/4\pi$	$f_c - f_d$
1	0	0	s_4	π	$f_c + f_d$
1	0	1	s_5	$5/4\pi$	$f_c - f_d$
1	1	0	s_6	$3/2\pi$	$f_c + f_d$
1	1	1	s_7	$7/4\pi$	$f_c - f_d$

Tabela 3.5.1 Regras de mapeamento para modulação 2 FSK/4PSK

3.6 PROTEÇÃO DESIGUAL DE ERROS.

Os códigos com proteção desigual de erros garantem uma proteção desigual para diferentes subconjuntos de informação; isto é, os subconjuntos de informação são codificados com códigos de diferentes distâncias mínimas.

Definição: [5] Sejam d_a^i e d_b^i as distâncias mínimas de Hamming para os códigos externos e internos respectivamente. Um código concatenado generalizado possui proteção desigual de erros quando $d_a^i . d_b^i$ decresce com i. A informação codificada pelo código externo $A^{(1)}$ é, então, protegida com a distância mínima $d_a^i . d_b^i$.

Pode-se notar que a proteção desigual de erros é bastante

evidente nos sistemas concatenados generalizados pela utilização de códigos de diferentes distâncias mínimas no processo de seleção e particionamento de conjuntos. Em geral o código externo oferece maior proteção que o interno.

3.7 ALGORITMO DE DECODIFICAÇÃO PROPOSTO.

Conforme visto nos ítens anteriores, foi notada a importância de se encontrar códigos e reticulados que apresentassem propriedades que facilitassem a decodificação dos mesmos. Com este objetivo foram analisados vários códigos e reticulados, modos para construí-los e suas propriedades específicas.

Como exemplo de aplicação das propriedades de reticulados, é apresentado um algoritmo para a decodificação do código C(2;32,64,16). Este algoritmo proposto faz uso de uma decisão suave parcial [25].

Inicialmente, a matriz geradora do código de Hamming estendido C(2;8,16,4), será colocada em uma forma de modo que as palavras do mesmo, se divididas ao meio, sejam palavras do código interno C(2;4,8,2). Assim, tem-se subcódigos nas linhas e colunas da matriz mapeada.

1-Através de qualquer algoritmo para decodificação de códigos de bloco com decisão suave, identificar os subcódigos de linhas e colunas da matriz recebida;

2-Fazer decisão abrupta na matriz obtida no passo anterior; desta maneira, o conjunto de matrizes fica restrito a quatro possíveis palavras-código com as seguintes propriedades: estas matrizes são constituídas por duas submatrizes (superior e inferior) de mesma dimensão com quatro combinações possíveis,

isto é, iguais duas a duas ou distintas duas a duas;

3- Somar, em módulo 2, a matriz obtida no passo anterior com uma das quatro matrizes-código possíveis. Devido à simetria destas matrizes, tem-se as seguintes possibilidades:

a) O número de 0's nas submatrizes superior e inferior é maior que o número de 1's; neste caso, esta matriz-código escolhida é a matriz decodificada, FIM;

b) O número de 0's da submatriz superior é maior que o número de 1's, ocorrendo o contrário com a submatriz inferior; neste caso, a submatriz inferior deverá ser substituída pela outra variante da mesma, FIM;

c) O número de 1's da submatriz superior é maior que o número de 0's, este é o caso análogo ao anterior, FIM;

d) O número de 1's tanto da submatriz inferior como da submatriz superior é maior que o número de 0's; neste caso, deve-se escolher a segunda variante para ambas as matrizes, FIM.

Como pode ser observado da Fig. 3.7.1, o desempenho deste algoritmo é 0,7 dB inferior ao desempenho apresentado pelo algoritmo MLD com decisão suave. Todavia, a complexidade do primeiro é menor que a do segundo, uma vez que o mesmo emprega correlação parcial.

3.8 CONCLUSÕES.

Neste capítulo, foram estabelecidos conceitos essenciais da concatenação generalizada de códigos de bloco. Através de exemplos clássicos de modulação codificada, ficou evidenciada esta generalização. Recursos de estruturas reticulares foram empregados na análise qualitativa desta generalização. Como aplicação desta técnica, foi também apresentado um algoritmo simples que usa decodificação suave parcial para a decodificação de códigos concatenados.

Como uma possível extensão deste trabalho podem ser sugeridos os seguintes ítens:

-particionamento variante no tempo;

-concatenação adaptativa (usando somente o código interno ou o código externo).

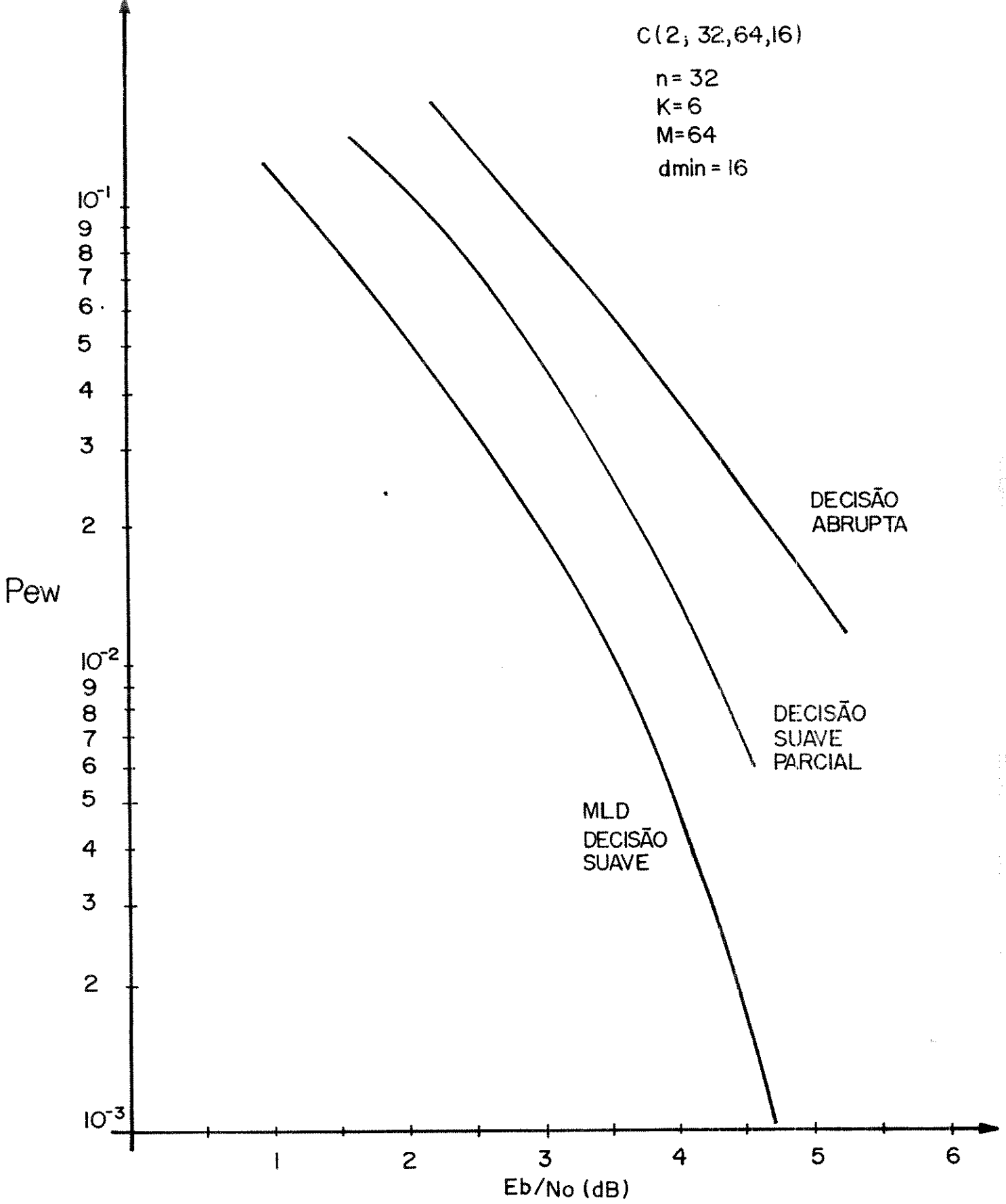


Fig. 3.7.1 Desempenho do algoritmo proposto com decisão suave parcial para o código C(2;32,64,16).

CAPÍTULO 4

DECODIFICAÇÃO DE CÓDIGOS DE BLOCO COM DECISÃO SUAVE ATRAVÉS DA GENERALIZAÇÃO DO ALGORÍTMO "VIZINHOS DE ZERO"

4.1 INTRODUÇÃO.

Neste capítulo é apresentada uma contribuição original aos algoritmos para decodificação de códigos de bloco com decisão suave. Preliminarmente é apresentado o algoritmo Vizinhos de Zero ("Zero-Neighbours") para decisão abrupta [30], após o que se propõe uma generalização deste algoritmo para decisão suave.

O problema da decodificação eficiente de códigos de bloco com decisão suave é, ainda, uma questão em aberto. O problema fundamental é a obtenção de algoritmos rápidos, eficientes, de fácil implementação e aplicáveis a uma grande classe de códigos.

Hartmann e Rudolph [28] e Gerard Battail et all:[2] propuseram um algoritmo ótimo para a decodificação de códigos de bloco com decisão suave, visando a minimização da probabilidade de erro de bit. Esse algoritmo, embora ótimo, é de fácil implementação apenas para códigos de taxas elevadas ou códigos de taxas baixas. A simplificação proposta por Greenberger [27], embora aparentemente viabilize a implementação desse algoritmo para códigos de taxas intermediárias, traz como resultado uma degradação do desempenho e uma complexidade elevada do circuito decodificador.

Wolf [54] propôs um algoritmo ótimo para a decodificação de códigos de bloco lineares sobre GF(q) com decisão suave. O esquema proposto utiliza o algoritmo de Viterbi aplicado a uma treliça com um número de estados não superior a $q^{(n-k)}$. A complexidade

desse algoritmo é baixa somente quando $q^{(n-k)}$ assume valores pequenos.

Chase [7] e outros [32,34,52] desenvolveram algoritmos subótimos que, para certos casos, são relativamente simples quanto à implementação, porém, com desempenho inferior ao do decodificador ideal de máxima verossimilhança (MLD). Entre os algoritmos mencionados, o que apresenta melhores resultados parece ser o algoritmo de Chase que, entretanto, possui o inconveniente da complexidade depender fundamentalmente da distância mínima de Hamming do código e de necessitar de um decodificador com decisão abrupta.

4.2 GENERALIZAÇÃO DO ALGORITMO VIZINHOS DE ZERO.

Levitin e Hartmann [30] propuseram um algoritmo ótimo com decisão abrupta, aplicável a códigos de bloco lineares. Para a decodificação da palavra recebida, esse algoritmo faz combinações lineares com as palavras-código de uma lista que contém, pelo menos, todas as palavras de peso mínimo e que, por hipótese, contém uma base do código. Uma palavra desta lista, organizada de maneira aleatória, é somada módulo 2 com a palavra recebida. Se o peso de Hamming do resultado desta soma é menor do que o peso de Hamming da palavra recebida, então esta palavra é armazenada. O processo se repete tomando-se agora como nova palavra recebida o resultado da soma efetuada. O algoritmo termina quando, após uma busca exaustiva na lista, for impossível a redução de peso mencionada. A palavra decodificada é então obtida pela soma módulo 2 das palavras armazenadas durante o processo. Pode ser demonstrado que, quando o algoritmo termina, traz como resultado final o padrão de erro da palavra recebida. Assim, quando a decodificação tiver sido correta, as posições em que as

componentes da palavra final for diferente de zero corresponderão aos bits errados da palavra recebida. É interessante observar que como a lista é utilizada de uma maneira aleatória, existem vários "caminhos de decodificação", uma vez que a palavra decodificada pode ser gerada através de diversos modos de combinações lineares. Levitin e Hartmann chamaram este algoritmo de "Zero-Neighbours Algorithm" (algoritmo de vizinhos de zero).

O algoritmo aqui proposto é uma generalização do Algoritmo Zero-Neighbours (ZNA) para decisão abrupta [30].

Para a generalização do ZNA para decisão suave, depara-se com as seguintes questões:

-definir uma operação análoga à operação módulo 2 para a redução do peso analógico;

-definir o conjunto das palavras vizinhas à palavra-código zero;

-obter uma regra de parada eficiente, ou seja, abortar a decodificação quando não for possível ou não for de interesse continuar o processo de redução de peso, evitando uma busca exaustiva na lista do ZNA;

O algoritmo aqui apresentado tem por objetivo conseguir um desempenho muito próximo ao do decodificador ótimo (MLD) com decisão suave. Será visto que sua complexidade é função da relação sinal/ruído do canal, do peso da palavra transmitida e das palavras vizinhas da palavra-código zero pertencentes à lista do ZNA.

Com o objetivo de aumentar a eficiência do algoritmo aqui proposto, serão realizadas algumas alterações, como, por exemplo, a incorporação do algoritmo de conjuntos de informação. Estas alterações, entretanto, só serão apresentadas no capítulo 5.

Com a finalidade de facilitar a implementação do

decodificador, são sugeridas e analisadas algumas simplificações de maneira a obter um algoritmo sub-ótimo com desempenho assintoticamente ótimo.

4.2.1. Considerações Iniciais.

A modulação utilizada será a antipodal (unidimensional) na qual se inclui a BPSK. As palavras-código serão supostas equiprováveis e o canal linear, sem memória, com ruído Gaussiano, branco, com média zero e variância σ^2.

Considere um código de bloco linear $\bar{C}(n,k,d)$ sobre GF(2).

Seja $\underset{\sim}{\bar{c}} = (\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_i, \ldots, \bar{c}_n) \in \bar{C}$ uma palavra-código e seja $\underset{\sim}{c} = (c_1, c_2, \ldots, c_i, \ldots c_n) \in C$ a sua representação equivalente no código C espaço Euclidiano $\mathbb{R}^n$, onde

$$c_i = 1 \text{ se } \bar{c}_i = 1 \quad \text{e} \quad c_i = -1 \text{ se } \bar{c}_i = 0 \qquad (4.2.1)$$

Sejam $\underset{\sim}{y}$ o vetor recebido em $\mathbb{R}^n$, $d_H(.,.)$ a distância de Hamming em $\bar{C}$ e $W_H(\underset{\sim}{\bar{c}})$ o peso de Hamming de $\underset{\sim}{\bar{c}} \in \bar{C}$. O peso analógico de $\underset{\sim}{y}$ será representado por $W_E(\underset{\sim}{y}) = \|\underset{\sim}{y} - \underset{\sim}{c}_0\|$, onde $\underset{\sim}{c}_0$ é a palavra-código zero.

4.2.2 Definições

Definição 4.2.1: [30] O domínio $D(\underset{\sim}{\bar{c}})$ da palavra-código $\underset{\sim}{\bar{c}} \in \bar{C}$ é o conjunto de todos os $\underset{\sim}{\bar{x}} \in Z$, onde Z é o conjunto de todas as n-uplas (vértices do hipercubo $\{0,1\}^n$), tal que $d_H(\underset{\sim}{\bar{x}}, \underset{\sim}{\bar{c}}) \leq d_H(\underset{\sim}{\bar{x}}, \underset{\sim}{\bar{c}}')$ para todo $\underset{\sim}{\bar{c}} \in C$, $\underset{\sim}{\bar{c}}' \neq \underset{\sim}{\bar{c}}$. O domínio da palavra-código zero, será representado por $D(\underset{\sim}{\bar{c}}_0)$.

Definição 4.2.2 :[30] A vizinhança $B(\bar{x})$ de $\bar{x} \in Z$ é o conjunto de todo $\bar{y} \in Z$, tal que $d_H(\bar{x}, \bar{y})=1$. A moldura $G(\bar{c})$ do domínio $D(\bar{c})$ de uma palavra-código $\bar{c} \in \bar{C}$ é o conjunto

$$G(\bar{c}) = \bigcup_{x \in D(\bar{c})} \left(B(\bar{x}) - D(\bar{c}) \right) \qquad (4.2.2)$$

Definição 4.2.3: [30] O conjunto de vizinhos de zero (Zero Neighbours) é o conjunto $N_o \subseteq \bar{C}$ de palavras-código tais que

$$G(\bar{c}_o) \subset \bigcup_{\bar{c} \in N_o} D(\bar{c}) \qquad (4.2.3)$$

$$e \quad |N_o| = \min_{N} \left\{ |N| : N \subseteq \bar{C}, \right.$$

$$\left. G(\bar{c}_o) \subset \bigcup_{\bar{c} \in N} D(\bar{c}) \right\} \qquad (4.2.4)$$

O conjunto de vizinhos de zero forma a cobertura mínima da moldura do domínio da palavra-código zero. Os vizinhos de zero são palavras-código, cujos domínios são adjacentes ao domínio de zero.

Definição 4.2.4: [10] A região de Voronoi $V(\bar{c}_i)$ de um vetor $\bar{c}_i$ é o conjunto fechado $V(\bar{c}_i) = \left\{ \bar{x} \in \mathbb{R}^n : \|\bar{x}-\bar{c}_i\| \leq \|\bar{x}-\bar{c}_j\| \text{ para todo } j \neq i \right\}$.

4.2.3 Soma módulo dois estendida ($\boxplus$) [21].

Como foi mencionado anteriormente, um dos problemas, a ser resolvido para a generalização do algoritmo Vizinhos de Zero

(ZNA), é a definição de uma operação análoga à operação módulo 2 para a redução do peso analógico.

Uma maneira usada em [15] e [16] para se solucionar este problema é utilizar como métrica a distância Euclidiana que seja igual ao módulo da diferença para um sistema quantizado.

Um exemplo ilustrará melhor esta abordagem. Seja C(2;7,16,3) o código de Hamming e Q=4 o número de quantizações do sinal analógico. Assim:

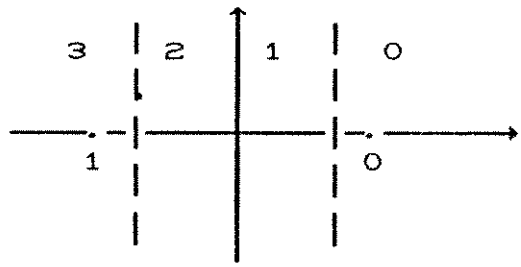


Fig. 4.2.1 Níveis de quantização para Q=4.

$$d_E(.,.) = (Q-1).d_H(.,.) = (4-1).3 = 9$$

Chamando de t_E o número de padrões de erro corrigidos com decisão suave, tem-se que:

$$t_E = \lfloor (d_E(.,.)-1)/2 \rfloor = 4$$

Seja $\underline{c}_t$ = (1 1 0 1 0 0 0) o vetor binário transmitido.

Seja $\underline{c}_r$ = (3 3 0 2 0 0 1) o vetor recebido quantizado em quatro níveis. A operação soma - resultado da correlação do vetor recebido com a palavra-código transmitida - é efetuada da seguinte maneira:

$$\begin{array}{r} 3\ 3\ 0\ 2\ 0\ 0\ 1 \\ +3\ 3\ 0\ 3\ 0\ 0\ 0 \\ \hline 0\ 0\ 0\ 1\ 0\ 0\ 1 \end{array}$$

Uma outra solução para o mesmo problema é a soma módulo dois estendida ($\boxplus$), [21], que será vista a seguir.

Definição 4.2.5: Seja um vetor $\underset{\sim}{\chi} \in \mathbb{R}^n$ de comprimento n representado por dois vetores de comprimento n :

$$\underset{\sim}{\chi} = (\underset{\sim}{\chi}_a, \underset{\sim}{\chi}_r) \qquad (4.2.5)$$

onde o vetor $\underset{\sim}{\chi}_a$ é obtido por decisão abrupta de todos os componentes do vetor $\underset{\sim}{\chi}$ e o vetor $\underset{\sim}{\chi}_r$ é o vetor cujos componentes são as confiabilidades (valores absolutos) dos componentes de $\underset{\sim}{\chi}$.

Assim,

$$y_{ai} = (1 + \text{sinal}(y_i))/2 \ ; \ y_{ai} \in GF(2)$$
$$y_{ri} = |\ y_i\ | \quad ; \quad y_{ri} \in \mathbb{R} \qquad (4.2.6)$$

Definição 4.2.6: [21] Será definida como operador $\boxplus$ a soma entre um vetor analógico $\underset{\sim}{\chi} \in \mathbb{R}^n$ e um vetor binário $\underset{\sim}{\alpha} \in GF(2)^n$:

$$\underset{\sim}{\chi} \boxplus \underset{\sim}{\alpha} = (\underset{\sim}{\chi}_a \oplus \underset{\sim}{\alpha},\ \underset{\sim}{\chi}_r) \qquad (4.2.7)$$

onde $\oplus$ simboliza a adição módulo 2 e $\underset{\sim}{\chi} = (\underset{\sim}{\chi}_a, \underset{\sim}{\chi}_r)$.

Como exemplo ilustrativo, seja o vetor recebido $\underset{\sim}{y} = (\ 1,\ -0,1,\ -0,5\ )$, $\underset{\sim}{y} \in \mathbb{R}^3$ e seja $\underset{\sim}{\alpha}=(\ 1,\ 1,\ 0\ )$ uma 3-upla qualquer. Então $\underset{\sim}{y} \boxplus \underset{\sim}{\alpha}$ será:

$$\begin{array}{rrr} & 1 & -0,1 & -0,5 \\ \boxplus & 1 & 1 & 0 \\ \hline & -1 & 0,1 & -0,5 \end{array}$$

Note-se que o operador $\boxplus$ preserva a confiabilidade de $\underset{\sim}{y}$, alterando somente o sinal dos componentes de $\underset{\sim}{y}$ nas posições em que os componentes de $\underset{\sim}{\alpha}$ forem iguais a 1. Note-se também que o vetor $\underset{\sim}{x}$ dado pela definição 4.2.5 pertence a $\mathbb{R}^{2n}$, porém, esta é só uma maneira conveniente de representar o vetor $\underset{\sim}{y}$, o qual de fato, pertence a $\mathbb{R}^n$.

Em seguida será vista uma propriedade importante da definição da soma $\boxplus$. Esta propriedade será útil para a demonstração do teorema fundamental do algoritmo proposto que generaliza o ZNA para decisão suave.

Propriedade da soma $\boxplus$: Seja um vetor analógico $\underset{\sim}{y} \in \mathbb{R}^n$ e um vetor-código $\underset{\sim}{c}_i \in \{1,-1\}^n$. Então, como decorrência da definição 4.2.6, a soma $\boxplus$ entre estes dois vetores será:

$$\underset{\sim}{y} \boxplus \underset{\sim}{c}_i = (-y_1 \cdot c_{i1},\ -y_2 \cdot c_{i2},\ \ldots,-y_n \cdot c_{in}) \qquad (4.2.8)$$

A soma $\boxplus$, em relação à soma quantizada proposta por Farrell [15], [16], possui a vantagem de ser mais fácil de ser implementada com qualquer número de níveis de quantização e, finalmente, como será visto adiante, é adequada para a implementação de limiares de parada do algoritmo a ser proposto.

Para facilidade das definições e demonstrações, assume-se que o vetor recebido $\underset{\sim}{y}$ é normalizado em relação ao valor absoluto de seu mais confiável componente, de modo a ficar em uma das faces do hipercubo $\langle 1,-1\rangle^n$.

4.3 REVISÃO DO ALGORITMO VIZINHOS DE ZERO (ZNA).

Em [30] Levitin e Hartmann descreveram dois distintos algoritmos de distância mínima que usam o conceito de vizinhos de zero. Um é baseado na propriedade fundamental dos vizinhos de zero e o outro é formulado em termos de síndrome da palavra recebida. Será abordado somente o primeiro algoritmo, pela sua simplicidade em relação ao segundo.

A propriedade fundamental do ZNA é a seguinte [30]:

Se $\underset{\sim}{\bar{x}} \notin D(\bar{\xi}_0)$, então existe $\xi \in No$ tal que

$$W_H(\underset{\sim}{\bar{x}} \oplus \xi) < W_H(\underset{\sim}{\bar{x}}) \qquad (4.3.1)$$

4.3.1 Algoritmo de Levitin e Hartmann [30].

Seja $\chi = \underset{\sim}{\bar{x}} \in Z$ a palavra recebida a ser decodificada.

No i-ésimo passo do algoritmo calcula-se

$$W_E(\chi_{i-1} \oplus \underset{\sim}{\bar{c}}) \text{ para todo } \underset{\sim}{\bar{c}} \in No.$$

Se existe $\underset{\sim}{\bar{c}}_i \in No$ tal que $W(\chi_{i-1} \oplus \underset{\sim}{\bar{c}}_i) < W(\chi_{i-1})$,

faz-se $\chi_i = \chi_{i-1} \oplus \underset{\sim}{\bar{c}}_i$

e vai-se ao passo seguinte. Caso contrário, o algoritmo termina.

Se o algoritmo termina no (m+1)-ésimo passo, então

$$\chi_m = \chi \oplus \sum \underset{\sim}{\bar{c}}_i \in D(\bar{\xi}_0)$$

pode ser tomado como o padrão de erro de peso mínimo, enquanto $\bar{c} = \sum \bar{c}_i \in \bar{C}$ é a palavra-código mais próxima de χ.

Para este algoritmo, são guardadas na memória somente as palavras vizinhas de zero, que podem ser calculadas antecipadamente ao processamento da palavra recebida . O cálculo de $|$ No $|$ é detalhado em [30]. Para os códigos perfeitos, as palavras vizinhas do zero constituem o conjunto de palavras de peso mínimo.

Para reduzir o número de passos do algoritmo, pode-se adicionar à lista No, palavra(s)-código adicionais (como, por exemplo, o vetor de peso n se ele fizer parte do código). Assim, o algoritmo é inicializado consultando-se previamente estas palavras. Com este procedimento, em média, reduz-se o número de passos pela metade [30]. Como será visto mais adiante, este resultado só é válido para decodificação com decisão abrupta.

A seguir, apresenta-se um teorema no qual está fundamentada a generalização do algoritmo de Hartmann e Levitin proposta neste trabalho.

4.4 TEOREMA 1. $\chi \in V(c_i)$ se e somente se $(\chi \boxplus c_i) \in V(c_o)$

Prova.

Seja χ um vetor analógico recebido, normalizado no hipercubo $\langle 1,-1\rangle^n$ e $c_i \in \langle +1,-1\rangle^n$, respectivamente dados por:

$$\chi = (y_1, y_2, \ldots, y_n)$$

e

$$c_i = (c_{i1}, c_{i2}, \ldots, c_{in}).$$

Seja c_o a palavra-código zero, dada por $c_o = (-1,-1,\ldots,-1)$.

Da propriedade da soma $\boxplus$ (eq. 4.2.8) tem-se que:

$$\chi \boxplus \underset{\sim}{c}_i = (-y_1 . c_{i1}, -y_2 . c_{i2}, \ldots, -y_n . c_{in}),$$

$$(\chi \boxplus \underset{\sim}{c}_i - \underset{\sim}{c}_o) = (-y_1 . c_{i1} + 1, -y_2 . c_{i2} + 1, \ldots, -y_n . c_{in} + 1),$$

$$\chi - \underset{\sim}{c}_i = (y_1 - c_{i1}, y_2 - c_{i2}, \ldots, y_n - c_{in}).$$

Porém, como $c_{ij}^2 = 1$,

$$(y_j - c_{ij}) = -c_{ij}(-y_j c_{ij} + 1)$$

e portanto,

$$\| \chi - \underset{\sim}{c}_i \| = \| \chi \boxplus \underset{\sim}{c}_i - \underset{\sim}{c}_o \| \qquad (4.4.1)$$

Conseqüentemente, tem-se que, $\chi \in V(\underset{\sim}{c}_i)$ se e somente se $(\chi \boxplus \underset{\sim}{c}_i) \in V(\underset{\sim}{c}_o)$, **c.q.d.**

Como conseqüência direta desse teorema tem-se que

$$(\chi - \underset{\sim}{c}_i) = -\underset{\sim}{c}_i \otimes (\chi \boxplus \underset{\sim}{c}_i - \underset{\sim}{c}_o) \qquad (4.4.2)$$

onde $\otimes$ designa o produto:

$$a \otimes b = (a_1 b_1, a_2 b_2, \ldots a_n b_n).$$

Daí, a soma $\boxplus$ pode ser definida como:

$$(\chi \boxplus \underset{\sim}{c}_i) = -(\chi \otimes \underset{\sim}{c}_i). \qquad (4.4.3)$$

4.4.1 Observações.

1-Uma interpretação do Teorema 1 pode ser obtida ao se expressar a distância Euclidiana entre dois vetores em termos de produto interno:

$$d^2(\chi_i, \underset{\sim}{c}_o) = \| \chi_i \|^2 - 2(\chi_i . \underset{\sim}{c}_o) + \| \underset{\sim}{c}_o \|^2 \qquad (4.4.4)$$

Para um dado χ, minimizar $d^2(\chi_i, \underset{\sim}{c}_o)$ em i é equivalente a maximizar o produto interno $(\chi_i . \underset{\sim}{c}_o)$ sobre i uma vez que $\|\chi_i\|^2$ é independente de i e $\| \underset{\sim}{c}_o \|^2$ é igual a n (comprimento do código).

O processo de decodificação, consiste em encontrar um vetor-código $\underset{\sim}{c}_i$ que possua a mínima distância Euclidiana possível de χ. Assim, de acordo com o Teorema 1 e equação (4.4.4), este processo consiste em maximizar o produto interno entre $(\chi_i \boxplus \underset{\sim}{c}_i)$ e $\underset{\sim}{c}_o$, uma vez que :

$$d^2(\chi_i \boxplus \underset{\sim}{c}_i, \underset{\sim}{c}_o) = \| \chi_i \boxplus \underset{\sim}{c}_i \|^2 - 2(\chi_i \boxplus \underset{\sim}{c}_i).\underset{\sim}{c}_o + \| \underset{\sim}{c}_o \|^2 \qquad (4.4.5)$$

Maximizar o produto interno é mais fácil, em termos computacionais, do que minimizar a distância Euclidiana. Este resultado será utilizado futuramente no capítulo 5.

2-A seguir é feita outra observação importante com relação ao teorema 1 e a generalização do algoritmo de Vizinhos de Zero para a decisão suave.

O problema principal que dificulta a generalização do algoritmo Vizinhos de Zero para decisão suave, é o fato de que qualquer componente pode assumir o valor zero, impossibilitando a diminuição do peso analógico. Isto poderá ser melhor observado através do seguinte:

Seja $\underset{\sim}{U}_i$ o vetor transmitido (+1,-1), de peso diferente de n e G o conjunto formado pelas componentes de $\underset{\sim}{U}_i$ iguais a um. Desta maneira $G = \{i \mid U_{ti}=1\}$.

Seja o caso particular, onde o vetor recebido χ_i é constituido somente de componentes zeros e -1, tal que nas posições onde $U_{ti}=1$, tem-se $y_i=0$ e onde $U_{ti}=-1$, tem-se $y_i=-1$. Como pode ser notado, é impossível reduzir o peso analógico deste vetor com qualquer palavra do código. Neste caso o algoritmo de Levitin e Hartmann assume que o mesmo pertence à $V(\underset{\sim}{c}_o)$.

Agora, para o mesmo vetor transmitido $\underset{\sim}{U}_i$, será suposto que o

vetor recebido χ_i consiste de uma pequena alteração do mesmo vetor χ_i visto anteriormente. Esta alteração é tal que um dos bits pertencentes à G seja infimamente superior à zero sendo que $\chi_i \notin V(\underset{\sim}{c}_o)$. Desta maneira, o peso deste vetor pode ser reduzido, mas, como pode ser visto, somente pela palavra-código a cuja região de Voronoi χ_i pertence. Qualquer outra palavra-código traria como consequência o aumento de peso nas posições não pertencentes à G.

Exemplo: Seja o código C(2;7,16,3) . Sejam ainda a palavra-código transmitida (-1,1,1,1,-1,-1,1) e o vetor recebido (-1,+δ,0,0,-1,-1,0), onde +δ é um infinitésimo positivo. Assim, a única palavra-código que reduz o peso é a própria palavra-código transmitida.

De acordo com o exposto, para que haja uma decodificação MLD exclusivamente por redução de peso, como é o caso do ZNA, a lista de correlação deve conter quase todas as palavras do código, o que não é uma boa solução.

Por outro lado, pode-se utilizar o mesmo conjunto de palavras do ZNA e, quando for impossível reduzir o peso de χ_i, usa-se uma palavra pertencente à lista a qual aumenta o mínimo possível o seu peso, e depois segue-se com o procedimento usual de redução de peso . Este processo é repetido até verificar-se que não é mais possível a sua redução. Então, considera-se que $\chi_i \in V(\underset{\sim}{c}_o)$ e o algoritmo termina.

4.5 ALGORITMO DE VIZINHOS DE ZERO PARA DECISÃO SUAVE .

Seja χ um vetor analógico recebido, normalizado no hipercubo $\langle 1,-1\rangle^n$.

Sejam ainda $\underset{\sim}{m}=\underset{\sim}{m}'=\underset{\sim}{c}_o$, $r=-\langle \chi . \underset{\sim}{c}_o \rangle$ e um contador I com $I_o=0$.

1) No i-ésimo passo busca-se $\underset{\sim}{c}_i \in No$ tal que:

$$W_E(\chi_i) > W_E(\chi_i \boxplus \underset{\sim}{c}_i) \qquad (C1)$$

Se a condição (C1) for satisfeita, ir ao passo 2.

Caso contrário, ir ao passo 3.

2) $\underset{\sim}{m} = \underset{\sim}{c}_i \boxplus \underset{\sim}{m}$; $\chi_i = \chi_i \boxplus \underset{\sim}{c}_i$

Se $\underset{\sim}{m}=\underset{\sim}{m}'$ ou $I=5$

então a palavra decodificada está armazenada em $\underset{\sim}{m}'$. FIM.

Vá para o passo 1.

3) $r^*=-\langle \chi_i . \underset{\sim}{c}_o \rangle$

Se $r^* < r$, então $\underset{\sim}{m}'=\underset{\sim}{m}$ e $r=r^*$

$I=I+1$

Encontrar $\underset{\sim}{c}_i^* \in N_o$, tal que:

$W_E(\chi_i) < W_E(\chi_i \boxplus \underset{\sim}{c}_i^*)$

e simultaneamente

$$d_E(\chi_i, \underset{\sim}{c}_i^*) = \min_{\underset{\sim}{c}_j \in N_o} \langle \chi_i, \underset{\sim}{c}_j \rangle$$

Faça $\underset{\sim}{m} = \underset{\sim}{c}_i^* \boxplus \underset{\sim}{m}$

4) Colocar a palavra encontrada $\underset{\sim}{c}_i$ no final da lista. Ir ao passo número 1.

4.5.1 Comentários.

1-Regra de parada do algoritmo.

De acordo com a idéia básica do algoritmo, o processamento termina quando não for possível uma redução de peso.

Conforme o passo de número 3 do algoritmo, a palavra decodificada em $\underset{\sim}{m}'$ será aquela cujo líder de classe lateral (coset leader) estiver mais próximo de $\underset{\sim}{c}_o$. Isto não garante, porém, que haverá uma repetição de $\underset{\sim}{m}'$, ou seja, pode acontecer que o líder de classe lateral atinja o seu valor somente uma vez, conforme a figura 4.5.1. Esta situação será chamada de "laços de redução".

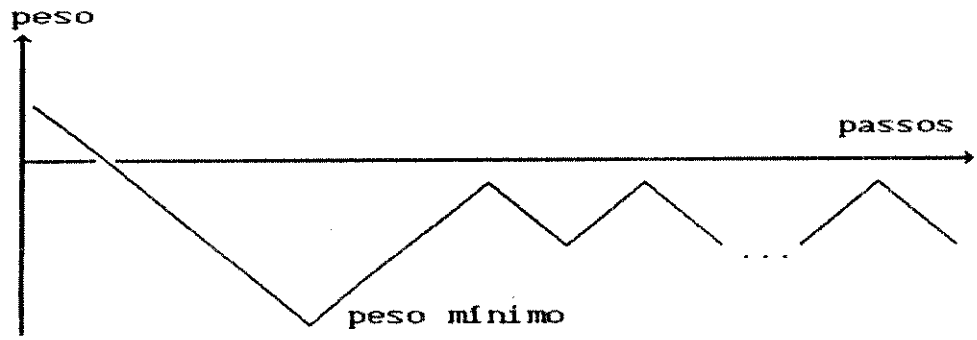


Fig. 4.5.1 "Laços de redução"

Uma regra de parada em software deve prever este caso que, como será visto na prática ,é muito raro. Isto é feito no passo 2, onde através do contador I limita-se o número de aumentos de peso igual a 5. Este número é função da relação sinal/ruído, influenciando muito pouco a velocidade e desempenho do algoritmo, uma vez que, como foi dito, serve quase que somente para paradas por "laços", que são raras. Na maioria das vezes a parada ocorre quando $\underset{\sim}{m}=\underset{\sim}{m}'$.

Uma parada por abortagem do tempo de decodificação poderá ser utilizada em conjunto com a regra de parada descrita.

2-Lista de correlação No.

A regra de parada vista anteriormente não garante uma decodificação MLD para |No|= número de palavras de peso mínimo. Visto com rigor, é necessário aumentar o peso várias vezes com diferentes palavras da lista até ser verificado que o processo de redução de peso se estabilizou, o que nos garante que o líder de classe lateral atingiu a região da palavra-código zero. Outra opção é de manter-se a regra de parada anterior e acrecentar-se mais palavras-código à lista No. Para códigos de taxa =1/2, onde este algoritmo tem o seu melhor desempenho, |No| é aproximadamente igual ao número de palavras de peso mínimo e o desempenho é assintoticamente igual ao do MLD. Como exemplo, tem-se que, para o código de Hamming (15,11), caso se quiser manter a regra de parada anterior, a lista de correlação deverá ser composta de todas as palavras de peso 3,4 e 5. O número de palavras da lista |No| foi encontrado através de simulações em computador, sendo notado que é função da relação sinal/ruído.

Exemplo de decodificação com o algoritmo proposto.

Seja um código C(2;7,16,3).

Seja χ_t uma palavra deste código a ser transmitida onde:

$$\chi_t = (-1\ 1\ 1\ 1\ -1\ -1\ 1)$$

Seja χ_i o vetor recebido normalizado entre -1 e 1 tal que:

$$\chi_i = (-1\ -0,2\ 0\ 0\ -1\ -1\ 1)$$

Com estes dados, um possível caminho de decodificação utilizando o algoritmo proposto é o seguinte:

-1	-0,2	0	0	-1	-1	1	Σ =	-2,2
0		1	0	0	0	1	1	
-1	0,2	0	0	-1	1	-1	Σ =	-1,8
0		0	1	1	0	1	0	
-1	0,2	0	0	-1	-1	-1	Σ =	-3,8
1		1	0	1	0	0	0	
1	-0,2	0	0	-1	-1	-1	Σ =	-2,2
1		1	0	1	0	0	0	
-1	0,2	0	0	-1	-1	-1	Σ =	-3,8

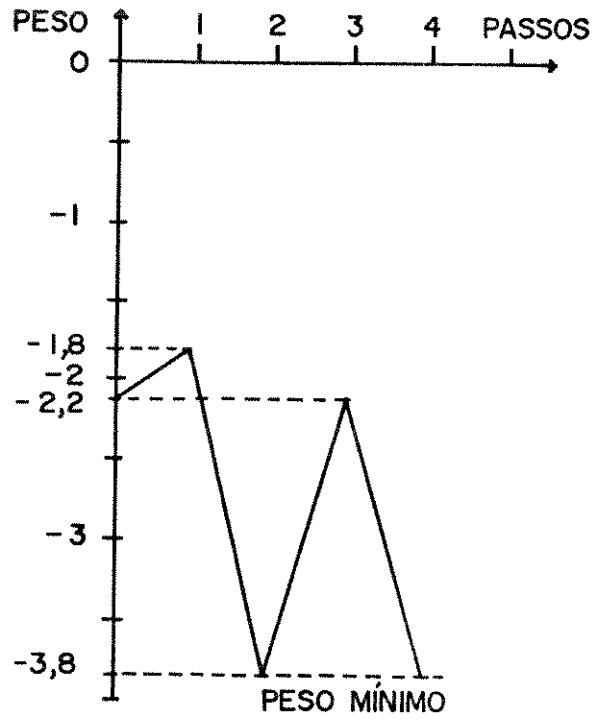


Fig. 4.5.2 Caminho de decodificação.

A palavra decodificada é obtida pela combinação linear (soma módulo 2) das palavras utilizadas até a estabilização do processo de redução de peso, ou seja :

$$\begin{array}{rl} & 0\ 1\ 0\ 0\ 0\ 1\ 1 \\ \oplus & 0\ 0\ 1\ 1\ 0\ 1\ 0 \\ \hline & 0\ 1\ 1\ 1\ 0\ 0\ 1 \end{array} \longrightarrow \text{palavra decodificada}$$

Outra maneira de decodificar a palavra recebida através deste algoritmo é fazer decisão abrupta no vetor recebido e trocar o sinal do bit em cuja posição o padrão de erros (líder de classe lateral) for positivo.

A tabela 4.5.1 e a figura 4.5.3 mostram o desempenho do algoritmo proposto para o código de Golay (23,12). Como pode ser observado, o desempenho do mesmo é praticamente igual ao do decodificador de máxima verossimilhança. Através de uma análise do algoritmo verifica-se, porém, que o número médio de correlações para decodificar cada palavra é muito elevado, variando entre 1200 a 1000 para relações sinal/ruído de 1 a 4 dB. Isto torna o algoritmo pouco eficiente. Uma maneira de aumentar a eficiência do algoritmo proposto neste capítulo, é a utilização de limiares para decodificação.

PALAVRAS TRANSM.	8192	8192	12288	16384
Eb/No (dB)	1	2	3	4
PARADA POR LAÇOS	4	4	0	0
TAXA DE ERRO	0,142	0,055	0,014	0,0020

Tab. 4.5.1 Estatística do algoritmo proposto para o código C(2;23,4096,7)

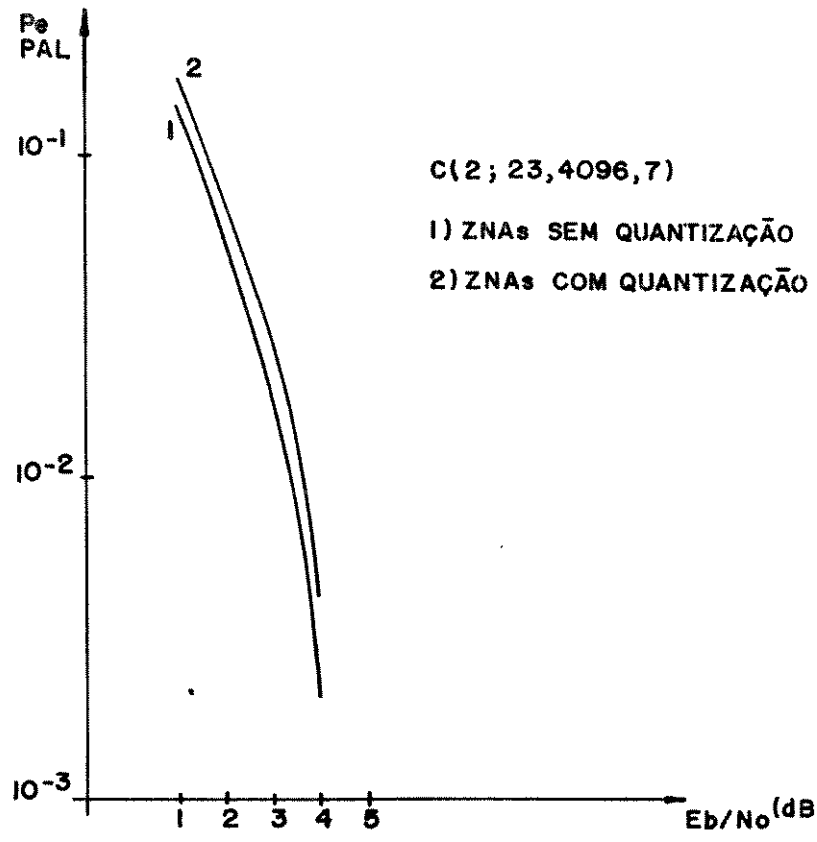


Fig. 4.5.3 Desempenho do algoritmo proposto para o código de Golay C(2;23,4096,7).

4.6 LIMIARES PARA DECODIFICAÇÃO.

Como foi colocado no início deste trabalho, é importante encontrar uma regra eficiente de parada do algoritmo visando à otimização da decodificação.

Um teste que, em certos casos, pode servir como regra de parada, é o limiar do "Generalized Minimum Distance" (GMD) [17]. Este limiar pode ser formulado pelo seguinte **teorema** [17]:

Teorema [17]: Em um código binário de bloco de comprimento n e de distância mínima de Hamming d_{Hmin}, existe quando muito uma palavra-código $\underset{\sim}{c}_i$ tal que:

$$\chi.\underset{\sim}{c}_i > n - d_{Hmin} \qquad (4.6.1)$$

onde χ é o vetor recebido normalizado entre [-1,1], $\underset{\sim}{c}_i$ é uma palavra-código pertencente a $(-1,1)^n$ e $\chi.\underset{\sim}{c}$ representa o produto interno entre os vetores χ e $\underset{\sim}{c}$.

De acordo com [17], se a condição (4.6.1) é satisfeita, então $\underset{\sim}{c}_i$ é a palavra-código mais próxima de χ.

Uma aplicação deste limiar no algoritmo do ZNA para decisão suave é vista em [21]. No algoritmo proposto em [21], são feitas n-d decisões abruptas nos bits mais confiáveis. O algoritmo pára por busca exaustiva ou quando o limiar GMD é atingido.

O limiar GMD, porém, não é ótimo, ou seja, ao aplicar-se o mesmo ao ZNA, somente um percentual de palavras decodificadas é "parado" por este limiar. Este percentual é função da relação sinal/ruído do sistema.

Uma otimização na aplicação deste limiar surge através da demonstração do teorema do GMD [3], [17]. Assim, no i-ésimo passo do algoritmo proposto (pág.61), pode-se buscar uma palavra-código $\underset{\sim}{c}_i'$ pertencente a No, tal que difira de $\underset{\sim}{c}_o$ nas d posições mais positivas de $\underset{\sim}{c}_o$. Como conseqüência, tem-se a seguinte regra de parada:

Se no i-ésimo passo de decodificação for possível escolher os d bits cuja soma seja positiva, continua-se a decodificação, caso contrário o algoritmo termina. Este resultado pode ser tido como óbvio, se obtido através do teorema do ZNA, em que, com uma palavra-código de peso mínimo tenta-se reduzir o peso de χ.

Um outro limiar muito mais estreito e com reduzida complexidade adicional é o limiar dado pelo ângulo δ. Este limiar é ilustrado pela representação do código C(2;3,4,2) no espaço

Euclidiano, como mostra a figura 4.6.1.

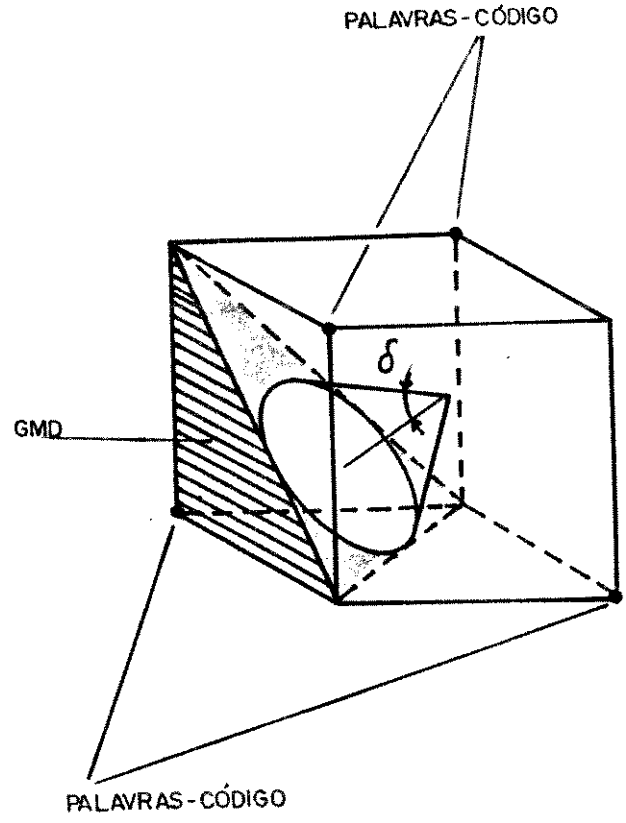


Fig. 4.6.1 Limiares para decodificação

Conforme a figura 4.6.1, tem-se que o ângulo δ de máxima abertura pode ser calculado da seguinte maneira:

$$\cos(\underset{\sim}{y},0,\underset{\sim}{c}) = \frac{(\underset{\sim}{y}\,.\,\underset{\sim}{c})}{\|\underset{\sim}{y}\|\cdot\|\underset{\sim}{c}\|} > \cos(\delta), \qquad (4.6.3)$$

onde $(\underset{\sim}{a}.\underset{\sim}{b})$ é o produto escalar de dois vetores $\underset{\sim}{a}$ e $\underset{\sim}{b}$.

Para um código binário linear com distância mínima d_{Hmin}, pode-se demonstrar [17] que :

$$\cos(\delta) = \sqrt{1-d_{Hmin}/n} \qquad (4.6.4)$$

Assim, se:

$$(\underset{\sim}{\chi} \cdot \underset{\sim}{c}) > \| \underset{\sim}{\chi} \| \sqrt{n - d_{Hmin}} \qquad (4.6.5)$$

então $\underset{\sim}{c}$ é a palavra-código mais próxima de $\underset{\sim}{\chi}$. Em outras palavras, $\underset{\sim}{\chi}$ está na região de Voronoi de $\underset{\sim}{c}$. Para o caso particular de $\underset{\sim}{c} = \underset{\sim}{c}_o$, a equação (4.6.5) fica simplificada:

$$\sum_{i=1}^{n} y_i < - \| \underset{\sim}{\chi} \| \sqrt{n - d_{Hmin}} \qquad (4.6.6)$$

O limiar GMD (4.6.1) em conjunto com a equação (4.6.5) será chamado de limiar GMD estendido.

Da fig. 4.6.1 fica claro que, para o código C(2;3,4,2), o limiar GMD definido pela equação 4.6.1 é mais apertado que o limiar definido pela equação 4.6.6, uma vez que o vetor recebido $\underset{\sim}{\chi}$ se encontra normalizado em uma das faces do cubo. Entretanto, para códigos mais longos acontece o contrário, ficando o limiar definido pela equação 4.6.6 mais eficiente se comparado com o limiar da equação 4.6.1.

Exemplo:

Para o código de Golay tem-se:

$\cos \delta = (1-7/23)^{1/2} = 0,834 \longrightarrow \delta = 33,482^{\circ}$

Esta regra de parada pode ser aplicada em sistemas ARQ com decodificação simplificada, onde somente seriam requeridas repetições caso o vetor $\underset{\sim}{\chi}_i$ se encontrasse fora do cone definido pela equação (4.6.5) ou (4.6.6).

Evidentemente, este limiar, além de não ser muito justo, se for utilizado com freqüência, poderia deixar o algoritmo mais lento, devido às operações efetuadas.

Após a aplicação do limiar GMD estendido, o algoritmo proposto fica estruturado da seguinte maneira:

$\underset{\sim}{\chi}$ = vetor recebido normalizado.

$\overline{\underset{\sim}{\chi}}$ = vetor recebido com decisão abrupta.

$\underset{\sim}{m} = \underset{\sim}{m}' = \underset{\sim}{c}_o$ (palavra-código pertencente a C)

$r' = -\underset{\sim}{\chi} \cdot \underset{\sim}{c}_o$ = negação do produto interno de $\underset{\sim}{\chi}$ e $\underset{\sim}{c}_o$

$I = 0, \quad i = 1.$

Se $S = \overline{\underset{\sim}{\chi}}\ H^T = \underset{\sim}{0}$, então decodifique $\underset{\sim}{\chi}$ como sendo a palavra-código $\underset{\sim}{m} = \overline{\underset{\sim}{\chi}}$.

Caso contrário, faça $\underset{\sim}{z} = \underset{\sim}{\chi}$ e continue.

1- Encontre $\underset{\sim}{\alpha} \in No$ tal que

$$W(\underset{\sim}{z}) > W(\underset{\sim}{z} \boxplus \underset{\sim}{\alpha}_i)$$

Se tal $\underset{\sim}{\alpha}_i$ não puder ser encontrado vá ao passo 3.

2- Faça:

$\underset{\sim}{m} = \underset{\sim}{m} \oplus \underset{\sim}{\alpha}_i$

$\underset{\sim}{z} = \underset{\sim}{z} \boxplus \underset{\sim}{\alpha}_i$

$i = i + 1$

Se $\underset{\sim}{m} = \underset{\sim}{m}'$ ou I=5, então $\underset{\sim}{\chi}$ é decodificado como sendo $\underset{\sim}{m}$ e a decodificação está terminada.

Caso contrário, vá para 1.

3- Faça $r = -\underset{\sim}{z} \cdot \underset{\sim}{c}_o$

Se $r < r'$ faça $\underset{\sim}{m}' = \underset{\sim}{m}$ e $r' = r$

Encontre $\underset{\sim}{\alpha}_i^* \in No$ tal que:

$$WC(\underset{\sim}{z}) < WC(\underset{\sim}{z} \boxplus \underset{\sim}{\alpha}_i^*) \text{ e } d_E(\underset{\sim}{z}, \underset{\sim}{\alpha}_i^*) = \underset{\underset{\sim}{\alpha}_j \in No}{MIN} d_E(\underset{\sim}{z}, \underset{\sim}{\alpha}_j)$$

Faça:

$$\underset{\sim}{m} = \underset{\sim}{m} \oplus \underset{\sim}{\alpha}_i^*$$

$$\underset{\sim}{z} = \underset{\sim}{z} \boxplus \underset{\sim}{\alpha}_i^*$$

Se $\underset{\sim}{z}$ e $\underset{\sim}{c}_o$ satisfazem o limiar GMD estendido, então $\underset{\sim}{z}$ está na região de Voronoi de $\underset{\sim}{c}_o$ e $\underset{\sim}{\chi}$ está na região de Voronoi de $\underset{\sim}{m}$. Decodifique $\underset{\sim}{\chi}$ como sendo $\underset{\sim}{m}$ e o algoritmo termina.

Caso contrário, continue.

4- Coloque $\underset{\sim}{\alpha}_i^*$ no fim da lista No.

$I = I + 1$

$i = i + 1$

Vá para o passo 1.

Os resultados através de simulações com computador para o código de Golay mostram que a complexidade do algoritmo é substancialmente menor se comparada com o decodificador de correlação completa precedido pelo teste da síndrome zero, especialmente para $Eb/No \geq 5$ dB.

Como exemplo, para o código de Golay e para $Eb/No = 5$ dB, o número médio do total de palavras consultadas da lista No encontra-se próximo de 400. Obviamente, o presente algoritmo requer algum esforço computacional adicional para a implementação dos testes de síndrome e dos limiares, porém, a

maior complexidade está realmente no número médio de palavras consultadas da lista No. Em face do exposto, analisa-se em seguida os parâmetros de complexidade e a eficiência do algoritmo Vizinhos de Zero, bem como são feitos comentários para a complexidade da generalização deste algoritmo para decisão suave.

4.7 COMPLEXIDADE E EFICIÊNCIA.

A idéia básica para medir a eficiência de um algoritmo é determinar a razão de crescimento dos tempos de execução do algoritmo conforme aumenta o tamanho da entrada. Em um caso como o visto, por exemplo, em que se tem uma busca em uma lista desordenada, a média de tempo consumida é da ordem de n/2, sendo n o número de vetores da lista. Se a lista estivesse ordenada de acordo com um determinado critério, o tempo seria reduzido a $\log_2(n)$.

Os algoritmos, de uma maneira geral, podem ser classificados em dois grandes grupos: polinomiais e exponenciais.

Os algoritmos eficientes são vinculados a alguma função polinomial e representados por $O(p(n))$. Os algoritmos do tipo exponencial serão representados por $O(c^n)$, onde c é uma constante.

O tempo requerido por um algoritmo, expresso como uma função do tamanho da entrada é denominado de complexidade no tempo do algoritmo. O comportamento limite da complexidade em função do crescimento da entrada, é denominado de complexidade de tempo assintótica. De maneira análoga, pode-se definir a complexidade em espaço e a complexidade assintótica em espaço, isto é, a quantidade de memória e a razão de crescimento da mesma que um algoritmo necessita para resolver um problema.

Apesar da classificação de um algoritmo em exponencial e polinomial, existe uma classe de problemas que estão em uma situação intermediária: as melhores soluções que se conhecem requerem tempos exponenciais; porém, não foi demonstrado ainda que eles não possuem soluções em tempos polinomiais. Estes problemas são chamados de não deterministicamente polinomiais (NP) [43]. Um subconjunto de NP são os chamados NP-completos. Uma propriedade notável destes problemas é que, se é encontrada uma solução eficiente para alguns deles, então pode-se obter soluções eficientes para todos os demais.

O decodificador de mínima distância (DMD) para um código corretor de erro linear é um problema computacional bastante difícil, o qual foi recentemente demonstrado como sendo um problema NP-completo [30].

4.7.1 Complexidade em tempo e espaço do ZNA.

Uma vez que para implementar o algoritmo ZNA é necessário arquivar todas as palavras pertencentes a No e usá-las em cada passo do algoritmo, a complexidade de espaço do algoritmo é determinada pelo número de palavras-código vizinhas de zero.

Visto que os cálculos de $\chi_{i-1} \oplus c_i$, $W(\chi_{i-1} \oplus c_i)$, $(c_i \in No)$ e as comparações de peso $W(\chi_{i-1} \oplus c_i)$ e $W(\chi_{i-1})$ podem ser feitas em paralelo, a complexidade em tempo do ZNA é determinada pelo número de passos m, onde, de acordo com [30], tem-se:

$m \leq \lfloor n/2 \rfloor$ para códigos com peso par e ímpar;

$m \leq \lfloor n/4 \rfloor$ para códigos com peso par somente;

onde $\lfloor a \rfloor$ é a parte inteira de a.

Assim, a complexidade em tempo do algoritmo cresce linearmente com o comprimento do código. Deste modo, o número de

palavras vizinhas de zero | No | é o fator crucial que determina a complexidade do algoritmo [30].

De acordo com [30], a complexidade assintótica pode ser caracterizada como:

$$F(R) = \lim_{n \to \infty} 1/n \log_2 M \ , \qquad (4.7.2)$$

onde R=k/n é a taxa do código e M é a complexidade espacial ou temporal.

Em [30] é mostrado que para R ≃ 0,5, onde o DMD é difícil de ser implementado, o ZNA fornece o máximo ganho em complexidade.

O número de palavras a ser guardado e o número de operações elementares (cálculo da distância de Hamming e comparações), cresce aproximadamente com a raiz quadrada do número total de palavras-código. Assim, a complexidade temporal do ZNA cresce linearmente com n.

Dado que o problema do DMD é um NP-completo, não se deve esperar encontrar um algoritmo cuja complexidade crescerá polinomialmente com o crescimento de n (este resultado trará como conseqüência que NP=P).

Com o uso do ZNA pode-se dar um desempenho linear ao tempo do DMD; porém, a complexidade em hardware proveniente do número de operações elementares cresce exponencialmente com n.

A complexidade do ZNA para a decisão suave pode ser caracterizada como sendo função da taxa do código e da relação sinal/ruído do canal. Assim, a complexidade do algoritmo que está sendo proposto, terá como limitante inferior (em função da relação sinal/ruído) a complexidade do ZNA. Esta complexidade pode ser diminuída se for tirado proveito da informação que se

tem sobre a confiabilidade dos bits para indexar a lista No, ou com o auxílio do algoritmo de conjunto de informação.

4.8 ORGANIZAÇÃO E PARTICIONAMENTO DA LISTA No.

Como foi visto, a complexidade do algoritmo proposto depende fundamentalmente de |No|. Em [15] é proposto um algoritmo para o cálculo de um subconjunto de No em função dos padrões de erros a serem corrigidos. Em outras palavras, este subconjunto depende da relação sinal/ruído do canal. Este conjunto é constante para todas as palavras recebidas, porém, não garante uma decodificação MLD.

Uma alternativa para a decodificação com decisão suave seria a de encontrar este subconjunto em função da confiabilidade da palavra recebida.

Uma das maneiras de se encontrar este subconjunto é utilizar um campo adicional de cada palavra da lista para a indexação das mesmas. Esta indexação poderá ser feita do seguinte modo:

1-Tome-se como objetivo tornar negativos (mais próximo da palavra zero) os k bits mais confiáveis;

2-As palavras da lista que tornarem negativos os k bits da palavra recebida, obterão um peso em função dos bits reduzidos que será gravado no campo de indexação;

3-As palavras da lista que possuem zero na posição mais confiável, receberão pesos inferiores ao dos anteriores, salvo se as mesmas se encontrarem no primeiro conjunto ;

4-As palavras que possuem zero na posição dos dois bits mais confiáveis receberão pesos inferiores ao dos demais, salvo se já se encontrarem nos conjuntos anteriores, e assim por diante.

Assim, se a lista for consultada na ordem dos pesos, a probabilidade de encontrar a palavra que reduza o peso analógico nas primeiras consultas é maior do que se a lista estivesse desordenada. Dependendo do comprimento dos códigos e da lista No, este processamento adicional pode ser muito vantajoso.

Outra maneira de indexar a lista é a seguinte:

1-Tomar o bit mais confiável da palavra recebida;

2-Se o mesmo for positivo, dividir a lista em dois grupos:

a) Um grupo contendo palavras com bit 1 (um) na posição mais confiável;

b) Outro grupo que contenha zero nesta posição;

3-Procurar no primeiro grupo a palavra que reduza ao máximo o peso da palavra recebida;

4-Aplicar o ZNAs ao segundo grupo, deixando de lado o primeiro grupo até o final, da decodificação, caso não ocorra aumento de peso. Caso contrário, voltar ao passo inicial;

5-Se o bit mais confiável da palavra recebida for negativo, não se passa pelo terceiro passo.

A indexação da lista deixa o algoritmo mais veloz, porém, não garante que o mesmo seja MLD. Isto, porque pode ocorrer que algumas palavras necessárias à redução de peso se encontrem na parte da lista que já foi consultada. Assim, tem-se uma otimização da lista em função da RSR do canal. É evidente que esta otimização é maior do que a proposta em [15], uma vez que leva em conta a confiabilidade de cada símbolo, exigindo, porém, um processamento adicional da palavra recebida e da lista do ZNA.

4.8.1 Resultados obtidos.

Os resultados obtidos por simulação em computador para o código de Golay mostram que o método de ordenação da lista, que inclui o particionamento da mesma, é mais eficiente que o método que prevê somente a ordenação dos k bits mais confiáveis. Por exemplo, para relações sinal/ruído entre 1 e 4 db o comprimento médio da lista caiu de 253 para aproximadamente 210. Obteve-se uma diminuição do tempo de processamento em torno de 20%.

O desempenho do algoritmo ZNAs para ambos os casos de indexação da lista é mostrado na fig.4.8.1.

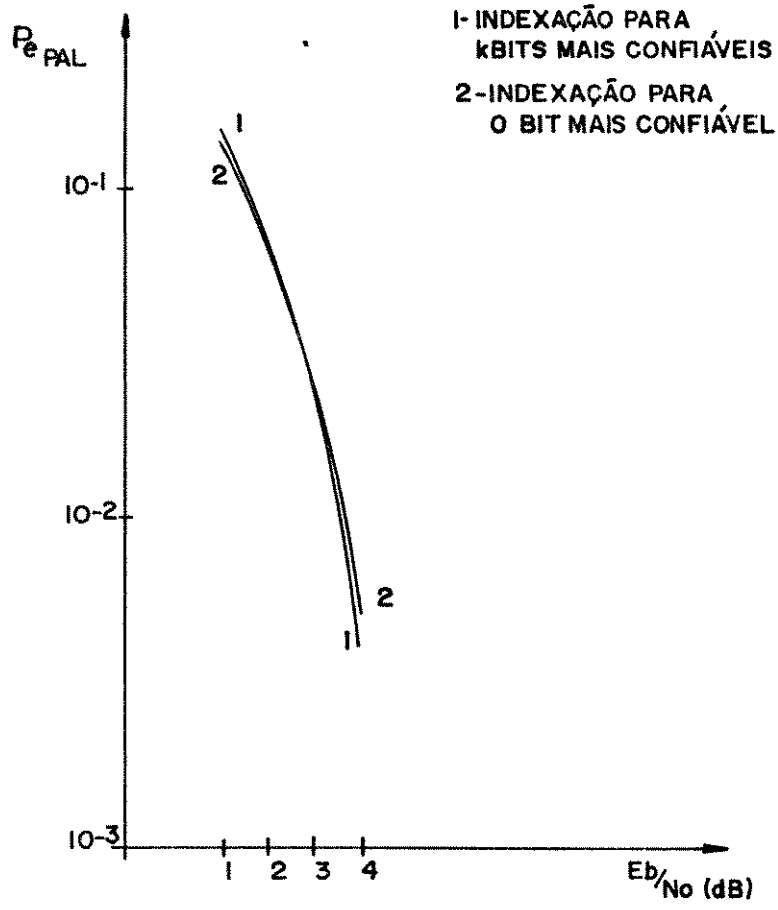


Fig.4.8.1 Desempenho do algoritmo proposto para C(2;23,4096,7)
1-Lista indexada em função dos k bits mais confiáveis.
2-Particionamento e indexação da lista em função do bit mais confiável.

CAPÍTULO 5

UM NOVO ENFOQUE PARA A DECODIFICAÇÃO POR CONJUNTOS DE INFORMAÇÃO COM DECISÃO SUAVE.

5.1 INTRODUÇÃO.

Neste capítulo, dando continuidade ao exposto no capítulo anterior, é proposta uma modificação na implementação do algoritmo tradicional de "Conjuntos de Informação", originando o que será chamado de "Algoritmo de Conjuntos de Informação Modificado-ACIM". É proposto também um algoritmo de decodificação que combina o ACIM com o algoritmo desenvolvido no capítulo anterior.

Definição 5.1.1 [8]: Em um código linear de bloco C(q;n,M,d), qualquer conjunto de k posições de uma palavra, que possa ser especificado independentemente, constitui um conjunto de informação. As (n-k) posições restantes são chamadas de conjunto de paridade.

Como conseqüência desta definição, os símbolos de um conjunto de informação definem uma palavra código. Se não houver erros nas posições do conjunto de informação, então é possível reconstituir a palavra transmitida. Existem vários tipos de algoritmos de conjunto de informação [8]. Basicamente, estes podem ser divididos em três categorias [8]:

-Algoritmos simples de conjuntos de informação ;

-Algoritmos do tipo Omura [36];

-Algoritmos que utilizam a técnica de síndrome parcial.

A grande vantagem destes algoritmos é que, praticamente, todos eles possibilitam a generalização para decisão suave. Com o objetivo de um casamento melhor entre os algoritmos de conjunto de informação e o algoritmo de vizinhos de zero, generalizado para decisão suave, é que é proposto o Algoritmo de Conjuntos de Informação Modificado. Este algoritmo enquadra-se na primeira categoria da classificação feita.

Para facilitar a exposição do algoritmo proposto, serão definidos alguns conceitos fundamentais.

Definição 5.1.2: Chama-se máscara de um conjunto de informação a um vetor binário que possui k bits unitários nas posições do conjunto de informação e (n-k) bits zeros nas posições do conjunto de paridade.

Definição 5.1.3: Denomina-se de conjunto de cobertura a uma coleção de máscaras de um mesmo código, cuja propriedade básica é diferirem entre si no maior número possível de posições.

Um exemplo simples pode ilustrar melhor estas definições. Considere-se o código de Golay C(2;23,4096,7), onde I_1 = (12,13,14,15,16,17,18,19,20,21,22,23) é um conjunto de informação. A máscara correspondente a este conjunto é dada pelo vetor

M = 00000000000111111111111

sendo que a matriz geradora (de cobertura) deste conjunto é:

$$G = \begin{vmatrix} 1\,0\,1\,0\,1\,1\,1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,1\,1\,1\,0\,0\,1\,0\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,0\,1\,0\,0\,1\,0\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 1\,0\,1\,1\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,1\,1\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0 \\ 1\,0\,1\,1\,1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,1\,0\,1\,1\,1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{vmatrix}$$

Obviamente o conjunto de informação dado, cobre, entre outros, os seguintes padrões de erro:

$$P_1 = 1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0$$
$$P_2 = 0\,0\,1\,0\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0$$

porém não cobre os padrões:

$$P_3 = 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0$$
$$P_4 = 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1$$

O algoritmo clássico de conjuntos de informação se resume no seguinte:

1- Selecionar diferentes conjuntos de informação para o código utilizado, de acordo com uma regra estabelecida. Em geral, procura-se conjuntos de informação que cubram a maioria dos padrões de erros que o código permite corrigir.

2- A partir do vetor recebido, construir palavras-código para cada conjunto de informação.

3- Escolher como palavra-código decodificada, a palavra do item anterior que estiver mais próxima da palavra recebida.

Usualmente, o algoritmo de conjuntos de informação visto acima possui a mesma complexidade para qualquer relação sinal/ruído. Neste trabalho, é proposta a incorporação do algoritmo de conjuntos de informação ao ZNA para decisão suave. Como resultado obteve-se um algoritmo adaptativo ao canal, ou seja,. a sua complexidade é função da relação sinal/ruído. Assim, através de um limiar eficiente para a "parada" do algoritmo, é possível otimizá-lo para o seu funcionamento em taxas de transmissão mais elevadas. A análise dos limiares será vista mais adiante.

Além da adaptação de algoritmo de conjuntos de informação, tem-se como objetivo utilizá-lo como uma sub-rotina do ZNA para decisão suave (ZNAs), resolvendo desta maneira o problema de velocidade de decodificação, principalmente para palavras de peso elevado.

O agrupamento dos algoritmos de conjuntos de informação e do ZNA oferece uma opção para o compromisso entre a complexidade e o desempenho do decodificador. A associação do ZNA ao algoritmo de conjuntos de informação traz como conseqüência uma melhoria no desempenho do decodificador porém, na prática, diminui a velocidade de operação do mesmo. A opção de associar-se ou não o ZNA ao algoritmo de conjuntos de informação depende da aplicação específica do decodificador.

Assim, a opção de agrupamento dos dois algoritmos pode ser obtida ao incorporar-se a seguinte rotina ao ZNAs:

Seja um vetor analógico $\underset{\sim}{y}$ recebido normalizado entre -1 e 1.

Seja um vetor $\underset{\sim}{m}$ inicialmente igual á 0.

1- Seja I_i um conjunto de informação do código C(q;n,M,d);

2- Se a soma do tipo $\boxplus$ da palavra-código gerada pela matriz do referido conjunto com o vetor χ possuir um peso analógico menor do que o do vetor χ, então somar esta palavra com o conteúdo do vetor $\underset{\sim}{m}$.

Caso contrário, ir ao passo número 3.

3- Escolher em uma ordenação qualquer outro conjunto de informação. Se todos os conjuntos de informação foram "varridos" uma única vez, ir ao passo número 4; caso contrário, ir ao passo número 1.

4- A palavra decodificada está no conteúdo de $\underset{\sim}{m}$. Fim.

Esta rotina será chamada de "Algoritmo de Conjuntos de Informação Modificado" (ACIM).

A seguir, é apresentado um teorema sobre a comparação de desempenhos entre o algoritmo tradicional de conjuntos de informação e o algoritmo de conjuntos de informação modificado (ACIM).

Teorema: Seja χ um vetor analógico recebido, normalizado no hipercubo $\langle -1,1\rangle^n$. Seja uma coleção I_I de conjuntos de informação e C_I um conjunto de palavras-código gerado pelo vetor χ e pela coleção I_I.
Então existe uma palavra-código $\underset{\sim}{c}_m \in C_I$ tal que

$$\min_{\xi \in C_I} d_E(\chi, \xi) = d_E(\chi, \xi_m)$$

onde $\xi_m = \xi_i \oplus \xi_{i+1} \oplus \ldots \oplus \xi_j$

e

$W(\chi_i) > W(\chi_{i+1})$,

onde $\chi_{i+1} = \chi_i \boxplus \xi_i$, $0 \leq i \leq j$

Prova:

A prova deste teorema consiste em demonstrar que os resultados da decodificação através do algoritmo de conjuntos de informação (ACI) e do algoritmo de conjunto de informação modificado (ACIM) são idênticos.

Seja o conjunto C_I formado pelas palavras-código

$$\langle \xi_0, \xi_1, \xi_2, \xi_3, \ldots, \xi_j \rangle$$

Decodificando-se inicialmente pelo ACI tem-se, sem perda de generalidade, que:

$$\| \chi - \xi_0 \| = W(\chi)$$

$$\| \chi - \xi_1 \| > \| \chi - \xi_0 \|$$

$$\| \chi - \xi_2 \| < \| \chi - \xi_0 \|$$

$$\| \chi - \xi_3 \| > \| \chi - \xi_2 \|$$

$$\| \chi - \xi_4 \| < \| \chi - \xi_2 \|$$

.....

.....

$$\| \chi - \xi_4 \| < \| \chi - \xi_j \|$$

Assim a palavra-código decodificada é ξ_4.

Para o ACIM tem-se a seguinte decodificação:

$$\| \chi - \xi_0 \| = W(\chi)$$

$WC(\chi \boxplus \xi_1) = \| \chi \boxplus \xi_1 - \xi_o \|$

$= \| \chi - \xi_1 \|$

$WC(\chi \boxplus \xi_1) > WC(\chi)$

$WC(\chi \boxplus \xi_2) < WC(\chi)$

$\chi_1 = \chi \boxplus \xi_2$

Será chamado de ξ_3' a palavra-código gerada por χ_1 e pelo conjunto de informação I_3. Uma vez que ξ_3 e ξ_3' são geradas pelo mesmo conjunto de informação, porém com a atuação de vetores distintos, ou seja, por χ e $\chi \boxplus \xi_2$ respectivamente, tem-se que:

$\xi_3' \boxplus \chi \boxplus \xi_2 = \xi_3 \boxplus \chi$

$\xi_3' \boxplus \xi_2 = \xi_3$ ou $\xi_3' \oplus \xi_2 = \xi_3$

$\xi_3' = \xi_2 \oplus \xi_3$

Desta maneira, tem-se que:

$WC(\chi \boxplus \xi_3') = WC(\chi \boxplus \xi_2 \boxplus \xi_3 \boxplus \xi_2)$

$= WC(\chi \boxplus \xi_3)$

$= \| \chi - \xi_3 \| > WC(\chi_1)$

$WC(\chi_1 \boxplus \xi_4') < WC(\chi_1)$

Como

$\xi_4' \boxplus \xi_2 = (\xi_4 \boxplus \xi_2) \boxplus \xi_2 = \xi_4$

então

$WC(\chi \boxplus \xi_4) < WC(\chi \boxplus \xi_2)$

.....

.....

$WC(\chi \boxplus \xi_4) < WC(\chi \boxplus \xi_j)$

Assim a palavra-código decodificada é também ξ_4, **c.q.d.**.

Para implementar a rotina do ACIM, é imprescindível responder as seguintes perguntas:

- Quantos conjuntos de informação serão necessários ?
- Que critério será adotado para encontrar tais conjuntos de

informação ?

Estas perguntas serão respondidas a seguir.

5.2 LIMITANTE INFERIOR PARA O CONJUNTO DE COBERTURA.

O número de conjuntos de informação, em geral, depende do número de padrões de erro que se deseja corrigir. Suponha-se que é necessário corrigir t erros. Em todas as n posições, o número total de distintos padrões é $\binom{n}{t}$. Se os conjuntos de informação forem associados com a matriz de paridade, a dimensão do conjunto de cobertura é n-k e o número máximo de padrões distintos que podem cobrir um determinado conjunto é $\binom{n-k}{t}$.

Assim, o menor número de conjuntos de informação que cobrem os padrões de t erros [48] é :

$$N_{cov} \geq \lceil n / n-k \lceil n-1 / n-k-1 \lceil \ldots \lceil n-t+1 / n-k-t-1 \rceil \ldots \rceil \rceil \rceil \qquad (5.2.1)$$

onde o símbolo $y = \lceil x \rceil$ é o menor inteiro y tal que $y \geq t$. Assim, para o código de Golay, para t=3 tem-se $N_{cov} \geq 15$.

Como, para decisão suave, tem-se por objetivo uma cobertura de um número maior de padrões de erro que o obtido para a decisão abrupta, o número de conjuntos de informação deverá também ser pelo menos maior que o calculado para decisão abrupta.

5.3 ALGORÍTMO DE DECODIFICAÇÃO PROPOSTO.

A seguir, é proposto um algoritmo de decodificação fundamentado nos algoritmos de Conjuntos de Informação Modificado (ACIM) e na generalização dos algoritmos de Vizinhos de Zero na sua forma mais simples, ou seja, sem a rotina de aumentos de peso. Para aumentar a eficiência do algoritmo, são usados os limiares de síndrome e GMD estendido, conforme a equação (4.6.6) .

O algoritmo será apresentado na forma de rotina, da seguinte maneira:

$\underset{\sim}{\chi}$ = vetor recebido normalizado;

$\overline{\underset{\sim}{\chi}}$ = vetor recebido com decisão abrupta;

$\underset{\sim}{c}_o$ = palavra-código zero em C;

$\underset{\sim}{\beta} = \underset{\sim}{c}_o$;

I= número máximo de conjuntos de informação a serem utilizados;

i=1.

Se $S = \underset{\sim}{\chi} \cdot H^T = 0$, então decodifique $\underset{\sim}{\chi}$ como sendo a palavra-código $\underset{\sim}{\chi}$. Caso contrário, faça $\underset{\sim}{z} = \underset{\sim}{\chi}$ e continue.

1- Tome o i-ésimo conjunto de informação da coleção N_I e forme uma palavra-código $\underset{\sim}{\alpha}$ a partir de $\underset{\sim}{\chi}$. Se

$$W(\underset{\sim}{z}) > W(\underset{\sim}{z} \boxplus \underset{\sim}{\alpha}) \qquad (5.3.1)$$

vá para o passo 2.

Faça i = i + 1. Se i > I, vá para o passo 3. Caso contrário, retorne para o passo 1.

2- Faça

$$\underset{\sim}{z} = \underset{\sim}{z} \boxplus \underset{\sim}{\alpha} \qquad (5.3.2)$$

$$\underset{\sim}{\beta} = \underset{\sim}{\beta} \oplus \underset{\sim}{\alpha}$$

Se $\underset{\sim}{z}$ satisfaz o limiar GMD estendido com $c = \underset{\sim}{c}_o$, vá para o passo 5. Caso contrário, faça i = i + 1.

Se i > I, vá para o passo 3. Vá para o passo 1.

3- Encontre $\underset{\sim}{\alpha} \in N_o$ tal que :

$$W(\underset{\sim}{z}) > W(\underset{\sim}{z} \boxplus \underset{\sim}{\alpha}) \qquad (5.3.3)$$

Se tal α não pode ser encontrado, vá ao passo 5.

4- Faça

$$\underset{\sim}{z} = \underset{\sim}{z} \boxplus \underset{\sim}{\alpha} \qquad (5.3.4)$$

$$\underset{\sim}{\beta} = \underset{\sim}{\beta} \oplus \underset{\sim}{\alpha}$$

Se $\underset{\sim}{z}$ satisfaz o limiar GMD estendido (4.14) com $\varsigma=\varsigma_o$, vá para o passo 3.

.5- Decodifique $\underset{\sim}{\chi}$ como sendo a palavra-código $\underset{\sim}{\beta}$.

Note que o algoritmo tenta sempre diminuir o peso do vetor recebido $\underset{\sim}{\chi}$ e de seus sucessores. Isto é feito primeiramente com a coleção de conjuntos de informação e depois com o auxílio das palavras da lista fixa N_o proveniente do algoritmo de Vizinhos de Zero [30]. Note, também, que o limiar GMD estendido é usado somente quando uma redução de peso acontece. Na prática verificou-se que o uso do limiar GMD estendido atua principalmente no ACIM, sendo que no algoritmo de vizinhos de zero sua atuação é quase nula. Isto deve-se ao fato de que o algoritmo ZNA, quando conjugado com o algoritmo de conjuntos de informação, decodifica, principalmente, palavras com alto ruído e, portanto, próximas às fronteiras de decisão em que os limiares possuem quase nenhuma atuação.

5.4 SISTEMA DE DECODIFICAÇÃO PROPOSTO.

A seguir é proposto um decodificador que utiliza o algoritmo visto de acordo com a figura 5.4.1.É feita agora uma descrição sucinta de cada bloco do decodificador.

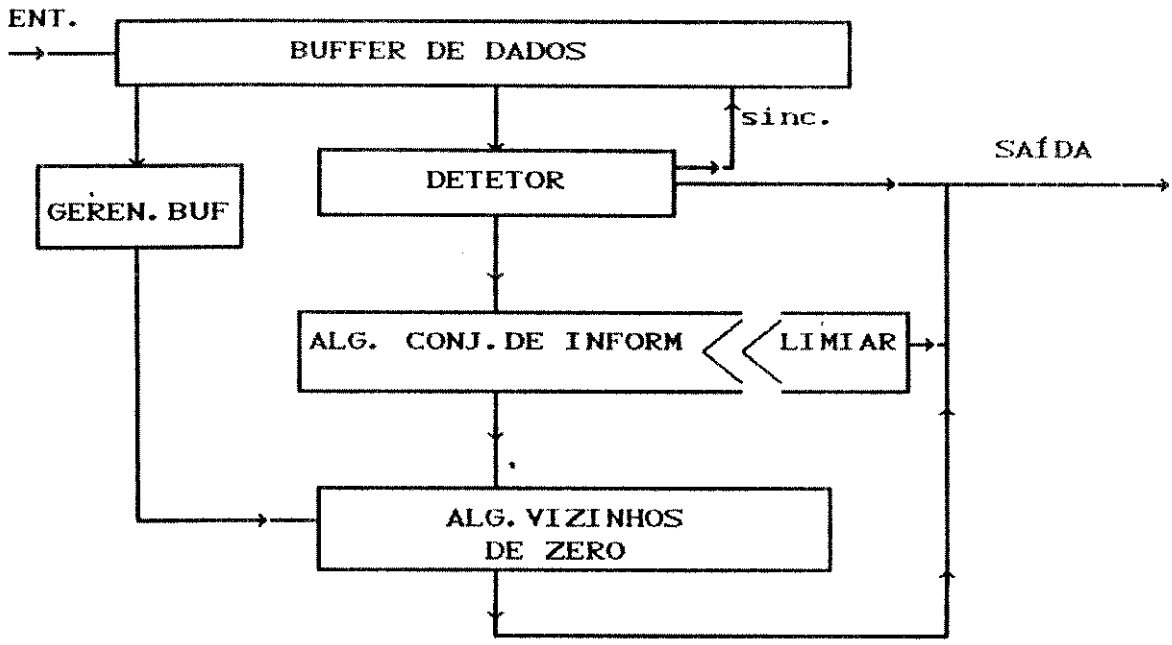


Fig. 5.4.1 Sistema de decodificação.

5.4.1 Buffer de dados.

Este bloco tem por finalidade armazenar os dados de entrada, enquanto uma palavra estiver sendo processada. O dimensionamento deste buffer é função da taxa de informação e da RSR do canal.

5.4.2 Gerenciador de buffer.

Tem por objetivo abortar o processamento quando o buffer não comportar mais dados.

5.4.3 Detetor.

Este bloco possui duas aplicações: evitar processamento desnecessário quando a palavra recebida estiver correta e fornecer sinal de sincronismo de bloco para o buffer de dados, caso haja uma falha ou alteração no sistema. Este sinal é fornecido por um monitor/comparador de erros acoplado ao detetor.

5.4.4 Algoritmo de Conjuntos de Informação Modificado (ACIM).

É utilizado o Algoritmo de Conjuntos de Informação Modificado em conjunto com a equação (4.6.6) do limiar do GMD estendido visto anteriormente. Este limiar é testado após cada redução de peso efetuada. Quando o mesmo não for atendido após passar por todos os conjuntos de informação, a palavra é submetida ao processamento adicional do algoritmo de Vizinhos de Zero (ZNA).

Verificou-se que, quando da utilização do detetor, o teste de parada do algoritmo obtido da equação (4.6.1) do limiar do GMD estendido, embora fácil de ser calculado, não possui praticamente utilidade. Isto porque este limiar só atua para altas RSR, onde o detetor opera com mais freqüência.

5.4.5. Generalização do algoritmo Vizinhos de Zero (ZNA-G).

É utilizado o algoritmo descrito inicialmente.

Não foi considerado o limiar do GMD estendido pelos motivos mencionados anteriormente.

5.5. RESULTADOS OBTIDOS E CONCLUSÕES.

Dos resultados da simulação obtidos com o sistema de decodificação proposto para o código de Golay C(2;23,4096,7), conclui-se o seguinte:

5.5.1 Desempenho do sistema de decodificação com o ZNAs.

O desempenho do decodificador com a primeira variante da generalização do Algoritmo de Vizinhos de Zero, isto é, incluindo os aumentos de peso, foi analisado no capítulo anterior. Este algoritmo, embora possua um desempenho próximo ao do MLD, é muito lento. Para melhorá-lo, sugere-se a sua implementação em circuito microprocessado com dois microprocessadores: um, para a redução de peso, e outro, para encontrar a palavra-código adicional às da lista.

Outra maneira para melhorar o desempenho deste algoritmo é, evidentemente, a incorporação do ACIM, de acordo com a fig.5.4.1. Os resultados obtidos através de simulações mostraram, porém, que é mais vantajoso, em termos de complexidade, aumentar-se o número de conjuntos de informação e simplificar o ZNAs, eliminando os aumentos de peso. Desta forma, o ZNAs transformou-se no ZNA-G do item 5.5.3.

5.5.2 Desempenho Isolado do Algoritmo de Conjuntos de Informação Modificado (ACIM).

Quando utilizado isoladamente, o ACIM pode atingir praticamente o desempenho do decodificador de máxima verossimilhança, sendo este desempenho função da relação sinal/ ruído e do número de conjuntos de informação.

A figura 5.5.1 mostra os resultados da simulação para diferentes números de conjuntos de informação. O desempenho depende, também, de como são escolhidos os conjuntos de informação, ou seja, da quantidade de padrões de erro que este conjunto abrange. Para isto, sob a coordenação e orientação do autor deste trabalho, estão sendo desenvolvidos, pelos alunos do **CEFET-PR**, programas para a análise e síntese de conjuntos de informação. Para a obtenção dos resultados da figura 5.5.1 , os conjuntos de informação foram gerados de maneira aleatória com a única condição de que somente fossem diferentes entre si.

Devem ser notadas **duas características básicas do ACIM**:

a) Se comparado com o algoritmo tradicional, descrito no item 5.1, tem-se que o ACIM é mais rápido na sua execução para circuitos microprocessados. Isto deve-se ao fato de que, após as primeiras reduções de peso, a maior parte dos bits do vetor resultante χ_i terão sinais negativos. Assim, quando da decisão abrupta nos k bits subsequentes, o número de combinações lineares para cada matriz geradora será bastante reduzido.

b) O ACIM propicia um "casamento" perfeito com o algoritmo Vizinhos de Zero, facilitando a implementação do sistema de decodificação proposto. Ambos os algoritmos (o ACIM e o ZNA) usam o princípio de redução de peso. A diferença consiste em que, para a redução de peso, o ACIM usa um conjunto variável de palavras que depende do vetor recebido, enquanto o ZNA usa um conjunto fixo de palavras-código.

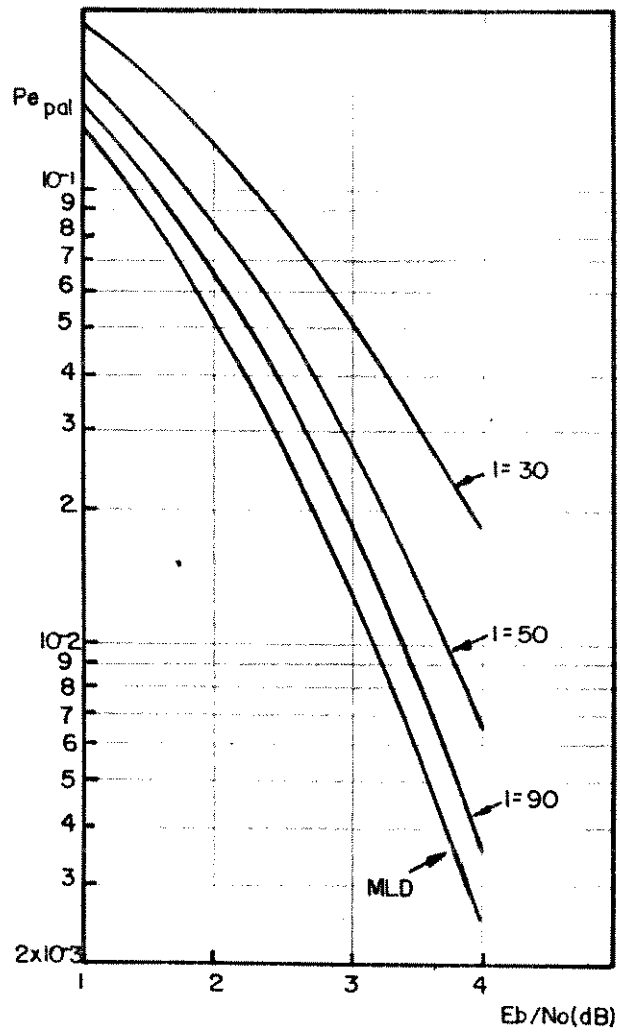


Fig. 5.5.1 Probabilidade de erro de palavra em função de Eb/No para o Algoritmo de Conjuntos de Informação Modificado, utilizado isoladamente.

I=30, 50 e 90 é o número de conjuntos utilizados.

5.5.3. Algoritmo Vizinhos de Zero Generalizado

O desempenho do decodificador aumenta consideravelmente, quando a versão do Algoritmo de Vizinhos de Zero Generalizado com decisão suave é introduzida juntamente com o ACIM. A figura 5.5.2 ilustra o desempenho do sistema completo para diferentes números de conjuntos de informação, escolhidos de maneira aleatória.

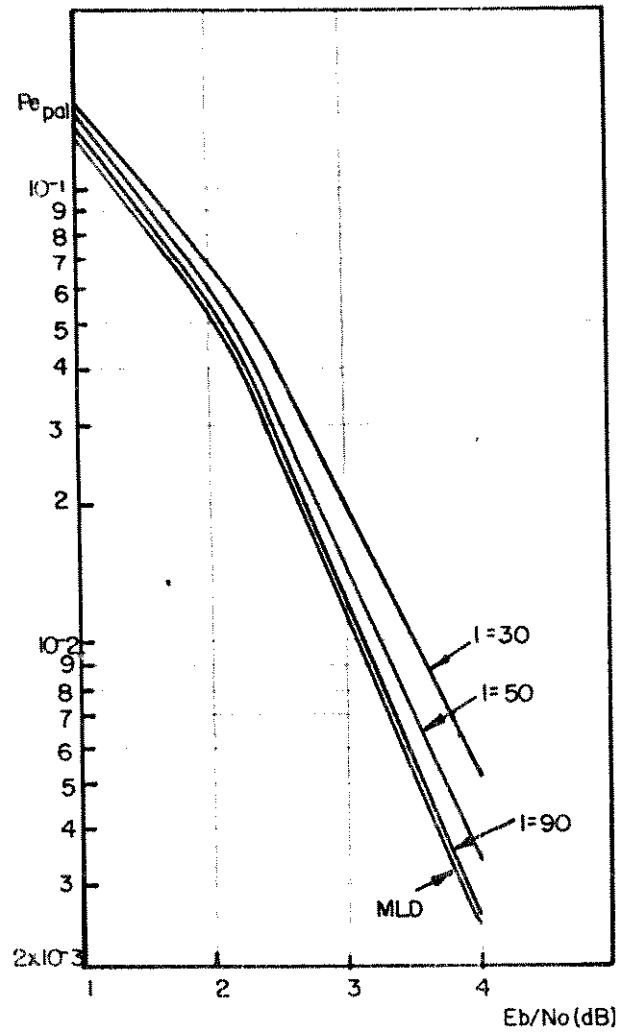


Fig. 5.5.2 Desempenho do sistema de decodificação proposto.

5.5.4. Adaptatibilidade do Decodificador.

O sistema é altamente adaptativo ao canal, ou seja, as palavras mais distorcidas pelo ruído necessitam de um processamento mais demorado para serem decodificadas em relação às palavras menos influenciadas pelo ruído. A tabela 5.5.1 ilustra a dependência do número de correlações de cada parte do algoritmo proposto em função da relação sinal ruído, o que apresenta também uma medida da complexidade do algoritmo em função do canal.

E_b/N_o (dB)	1	2	3	4
N. DE PALAVRAS TRANSMITIDAS	8192	8192	12288	16384
N. DE ACESSOS AO A C I M	621226	523834	576717	448385
N. DE ACESSOS AO A C I M POR PALAVRA	75,83	63,94	46,93	27,37
N. DE ACESSOS AO ZNA-G	1774440	1477340	1601692	1206702
MÉDIA DE ACESSOS AO ZNA-G POR PALAVRA	216,61	180,34	130,34	73,65
MÉDIA TOTAL DE ACESSOS POR PALAVRA	292,44	244,28	176,18	101,02
TAXA DE ERRO DO A C I M (P_{PAL})	0,1580	0,0658	0,0181	0,0036
TAXA DE ERRO TOTAL (P_{PAL})	0,1434	0,0567	0,01375	0,002563

Tab. 5.5.1 Desempenho do algoritmo proposto para 90 conjuntos de informação.

5.5.5. Otimização do algoritmo proposto.

Existem basicamente duas formas de otimização do algoritmo proposto:

-Aumentar a eficiência da regra de parada, encontrando limiares mais justos;

-Selecionar conjuntos de matrizes de modo que cubram o maior número de padrões de erros possíveis.

O primeiro item que é uma tarefa de considerável complexidade poderá ser tratada em uma futura extensão deste trabalho.

Com relação ao segundo item tem-se que, apesar da dificuldade de desenvolvimento de algoritmos genéricos e eficientes para selecionar conjuntos de cobertura otimizados, é possível conseguir alguns resultados parciais. A tabela 5.5.2 ilustra o ganho obtido na escolha parcialmente otimizada de 100 conjuntos de informação para o código de Golay.

ERROS	PADRÕES DE ERROS	PADRÕES DE ERROS NÃO COBERTOS	
		CONJUNTO ALEATÓRIO	CONJUNTO OTIMIZADO
1	23	NENHUM	NENHUM
2	253	NENHUM	NENHUM
3	1771	NENHUM	NENHUM
4	8855	191	11
5	33649	8241	5611

Tab. 5.5.2 Ganho obtido pelo selecionamento de 100 conjuntos de cobertura para o código de Golay C(2;23,4096,7).

E_b/N_o (dB)	1	2	3	4
N. DE PALAVRAS TRANSMITIDAS	8192	8192	12288	16384
N. DE ACESSOS AO A C I M	690443	581997	641483	495938
N. DE ACESSOS AO A C I M POR PALAVRA	84,28	71,04	52,20	30,27
N. DE ACESSOS AO ZNA-G	1774075	1476134	1601220	1207295
MÉDIA DE ACESSOS AO ZNA-G POR PALAVRA	216,56	180,19	130,30	73,68
MÉDIA TOTAL DE ACESSOS POR PALAVRA	300,84	251,23	182,51	103,96
TAXA DE ERRO DO A C I M (P_{PAL})	0,1583	0,0681	0,0180	0,004638
TAXA DE ERRO TOTAL (P_{PAL})	0,1431	0,0600	0,0140	0,002868

Tab. 5.5.3 Desempenho do algoritmo proposto para 100 conjuntos de informação (ALEATÓRIOS).

E_b/N_o (dB)	1	2	3	4
N. DE PALAVRAS TRANSMITIDAS	8192	8192	12288	16384
N. DE ACESSOS AO A C I M	689731	580861	638612	490970
N. DE ACESSOS AO A C I M POR PALAVRA	84,19	70,90	51,97	29,96
N. DE ACESSOS AO ZNA-G	1764592	1472538	1599552	1205850
MÉDIA DE ACESSOS AO ZNA-G POR PALAVRA	215,40	179,75	130,17	73,59
MÉDIA TOTAL DE ACESSOS POR PALAVRA	299,60	250,65	182,14	103,55
TAXA DE ERRO DO A C I M (P_{PAL})	0,1517	0,0620	0,01652	0,003234
TAXA DE ERRO TOTAL (P_{PAL})	0,1425	0,0576	0,01367	0,002441

Tab. 5.5.4 Desempenho do algoritmo proposto para 100 conjuntos de informação (SELECIONADOS).

O gráfico da figura 5.5.3 ilustra o desempenho do algoritmo proposto para os dois tipos de conjuntos de informação vistos.

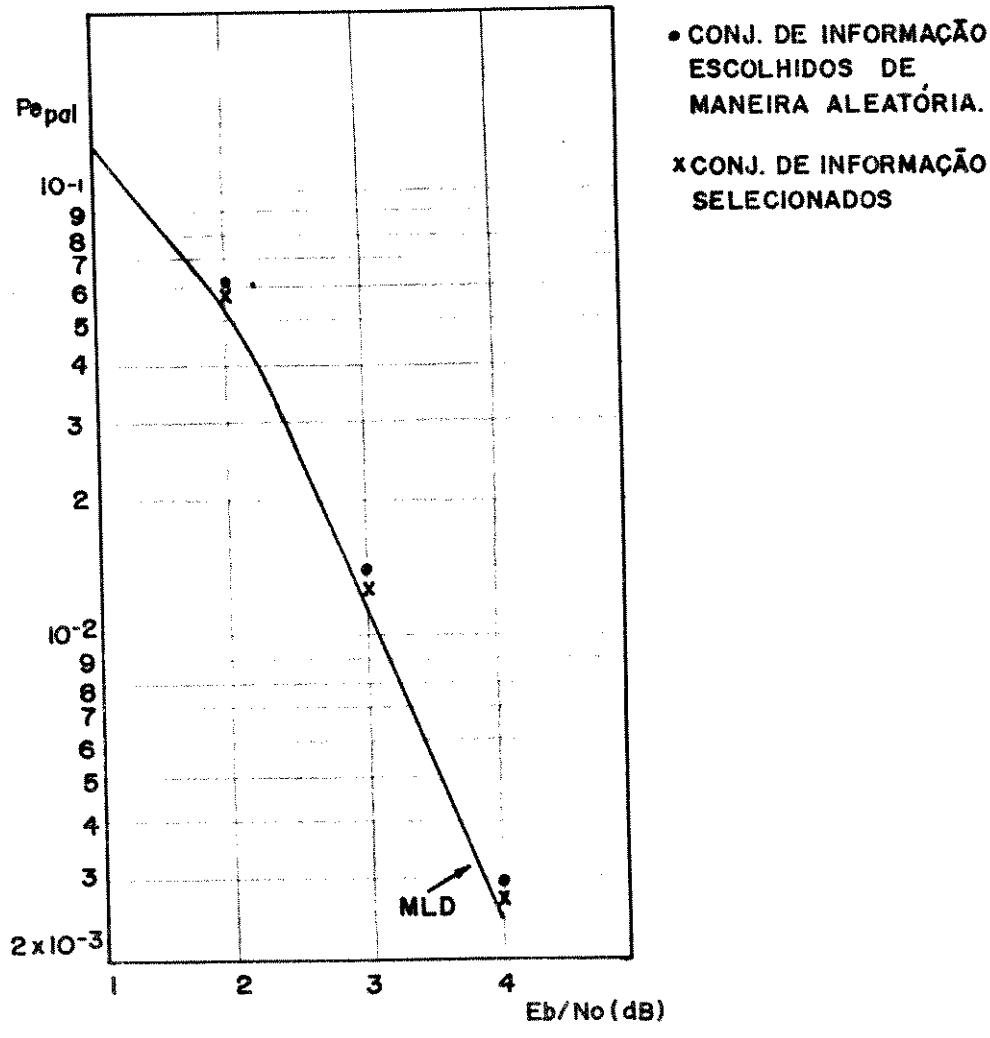


Fig. 5.5.3 Desempenho do algoritmo proposto para os conjuntos de informação da tab. 5.5.2.

5.5.6 Desempenho do algoritmo proposto para códigos longos.

Em seguida é analisado o desempenho do algoritmo proposto para códigos mais longos. Com o objetivo de se efetuar tal análise, foi escolhido o código de resíduos quadráticos $C(2;48,2^{24},12)$, uma vez que até a presente data a forma mais prática de se decodificar este código é com uma das variantes do algoritmo de conjuntos de informação [8]. Inicialmente foram utilizados 500 conjuntos de informação obtidos de forma aleatória. Os resultados da simulação em computador estão na tabela 5.5.5 e na figura 5.5.4. Foi utilizado o Algoritmo de Conjuntos de Informação modificado de forma isolada uma vez que não havia sentido de se utilizar o algoritmo ZNA-G visto o grande número de palavras de peso mínimo: 17.296. Para a elaboração da tabela 5.5.5 foi simulada uma fonte de dados com distribuição uniforme das palavras do código. A geração das palavras foi feita de maneira aleatória. Paralelamente foram feitas simulações em computador, onde se utilizou uma palavra-código de peso 24. A escolha de uma palavra de peso 24 se deve ao fato de existirem muitas palavras-código deste peso (768.1680) em relação ao total de palavras do código. Os resultados obtidos com ambas as simulações foram bastante próximos. Os resultados tabelados se referem a uma fonte aleatória de palavras-código.

E_b/N_o (dB)	2	3	4
N. DE PALAVRAS TRANSMITIDAS NA SIMULAÇÃO	300	1000	3040
N. DE ACESSOS AO A.C.I.M	147022	448337	922689
MÉDIA DE ACESSOS AO A.C.I.M. POR PALAVRA	490,07	448,34	303,52
TAXA DE ERRO DE BIT	$2,9.10^{-2}$	$4,9.10^{-3}$	$5,2.10^{-4}$

Tab. 5.5.5 Desempenho do A C I M utilizado isoladamente para 500 conjuntos de informação, para o código (48,24).

Em seguida, através de simulação em computador, foi estimado o desempenho do ACIM para o código (48,24) utilizando 800 conjuntos de informação obtidos de forma aleatória. Os resultados desta simulação estão contidos na tabela 5.5.6 e na figura 5.5.4.

E_b/N_o (dB)	2	3	4
N. DE PALAVRAS TRANSMITIDAS NA SIMULAÇÃO	300	1000	3000
N. DE ACESSOS AO A C I M	235223	717137	1657091
MÉDIA DE ACESSOS AO A C I M POR PALAVRA	784,07	717,14	553,02
TAXA DE ERRO DE BIT	$2,2.10^{-2}$	$3,8.10^{-3}$	$3,3.10^{-4}$

Tab. 5.5.6 Desempenho do Algoritmo de Conjuntos de Informação Modificado utilizado isoladamente para 800 conjuntos de informação, para o código (48,24).

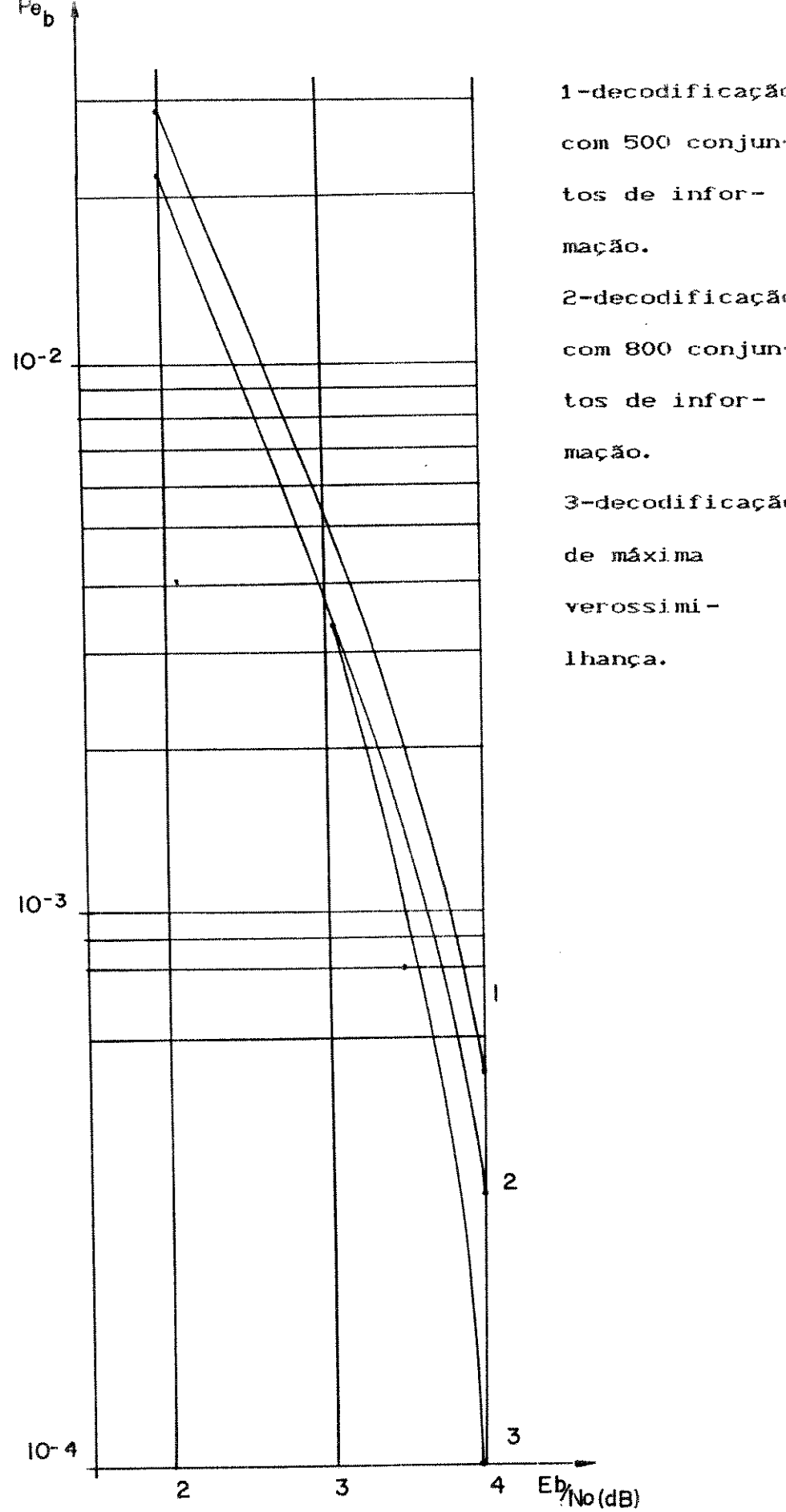


Fig. 5.5.4 Desempenho do ACIM : tabelas 5.5.5 e 5.5.6.

5.5.7 Conclusões.

Em face da análise dos resultados obtidos, conclui-se que o algoritmo proposto apresenta um desempenho quase ótimo com baixa complexidade. Aparentemente, o algoritmo parece mais adaptado para implementação em software. Foi implementado um decodificador que utiliza este algoritmo em um circuito microprocessado para baixas taxas de transmissão (2 a 5 kbps) com o código de Golay C(2;23,4096,7). De acordo com simulações feitas em tempo real com o protótipo do decodificador, foi comprovada a adaptabilidade do mesmo. Assim, na faixa de E_b/N_o = 5 a 10 dB, o decodificador pode operar em taxas de transmissão entre 2 e 5 kbps.

CAPÍTULO 6

DECODIFICAÇÃO EM SISTEMAS COM MODULAÇÃO CODIFICADA E CÓDIGOS MULTINÍVEIS .

6.1 INTRODUÇÃO.

Neste capítulo é apresentada uma revisão de uma classe de códigos multiníveis, denominados pseudo-cíclicos em [1], bem como são propostos algoritmos para a sua decodificação. Neste trabalho, estes códigos serão chamados de plésio-cíclicos a fim de evitar dubiedades com os conhecidos códigos binários pseudo-cíclicos.

Os algoritmos de decodificação propostos, apresentados para modulação MPSK codificada, se baseiam em uma generalização dos algoritmos já desenvolvidos para o caso binário, propostos nos capítulos 4 e 5.

É proposta, ainda, uma variante do algoritmo desenvolvido, em que se empregam perturbações em alguns símbolos com baixa confiabilidade.

A análise de desempenho dos algoritmos é realizada através de resultados de simulação em computador.

Utilizando os conceitos desenvolvidos nos capítulos 2 e 3, é analisado um método de representação para essa classe de códigos definidos sobre anéis de inteiros Z_q .

6.2 CONCEITOS BÁSICOS.

Em [1] é apresentado um processo de modulação codificada que usa códigos de bloco multiníveis definidos sobre o anel de

inteiros módulo q. O processo descrito em [1] consiste basicamente de duas etapas:

1- Mapeamento dos bits de informação no conjunto expandido de sinais ;

2- Codificação dos símbolos do espaço de sinais visando a maximização da distância Euclidiana mínima entre os blocos de símbolos (palavras-código).

Serão apresentadas, inicialmente, as definições e alguns parâmetros para a análise desta classe de códigos multiníveis.

Definição 6.2.1 [1]: Os códigos plésio-cíclicos multiníveis são códigos de bloco lineares C(q;n,M,d) definidos por uma matriz geradora G de dimensão k X n, com elementos pertencentes ao anel de inteiros Z_q, dada por:

$$G = \begin{vmatrix} g_{11} g_{12} \cdots\cdots\cdots g_{1r} \cdots\cdots \\ 0 \quad g_{11} g_{12} \cdots\cdots\cdots g_{1r} \cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ 0 \cdots\cdots\cdots\cdots g_{11} \; g_{12} \cdots g_{1r} \end{vmatrix}$$

onde $r = n-k+1$ e $g_{ij} \in Z_q$.

O processo de codificação então se resume a:

$$V = \underset{\sim}{a} \cdot G \qquad (6.2.1)$$

onde o vetor de saída V (palavra-código) e o vetor de entrada $\underset{\sim}{a}$ (informação) possuem as dimensões n e k, respectivamente.

Estrutura do codificador. O processo de codificação proposto em [1] possui a seguinte estrutura:

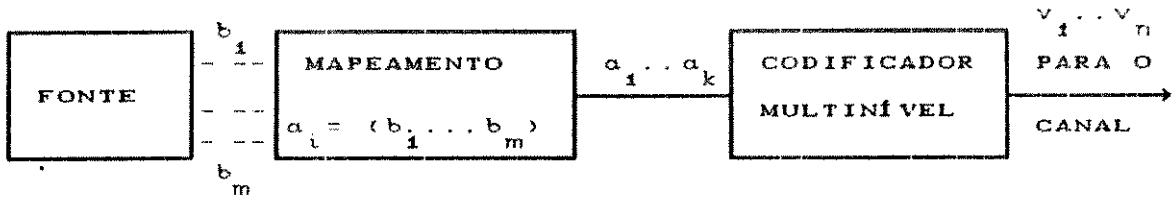


Fig. 6.2.1 Estrutura do codificador.

O ganho assintótico de codificação g_∞ para sistemas codificados em relação aos não codificados, é dado pela seguinte expressão [1]:

$$g_\infty = g\left(E_b/N_0 \longrightarrow \infty\right) = 10 \log_{10}\left[\frac{\log Mc}{\log Mu} \cdot R_c \cdot \frac{D_{Ec}^2}{D_{Eu}^2}\right] dB \qquad (6.2.2)$$

onde :

M_c e M_u são as cardinalidades dos alfabetos de modulação codificada e não codificada, respectivamente;

D_{Ec} e D_{Eu} são as distâncias Euclidianas mínimas resultantes da modulação codificada e da não codificada, respectivamente.

$R_c = k/n$ é a taxa de codificação da modulação codificada.

6.3 ALGORITMO DE DECODIFICAÇÃO PROPOSTO PARA UM SISTEMA COM MODULAÇÃO MPSK CODIFICADA COM CÓDIGO MULTINÍVEL.

6.3.1 Considerações gerais.

O algoritmo a ser analisado é uma extensão para duas dimensões do algoritmo unidimensional apresentado nos capítulos 4 e 5., Para tanto, faz-se necessário estender para duas dimensões os conceitos de soma ⊞, cálculo de pesos e limiar GMD.

6.3.2 Generalização da soma ⊞ para a modulação MPSK.

Neste parágrafo é feita uma extensão para duas dimensões, da soma ⊞ definida no capítulo 4 para modulações unidimensionais. Em particular, será abordada a modulação M-PSK.

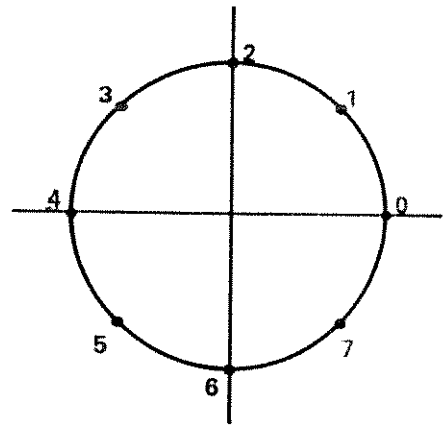


Fig. 6.3.1 Mapeamento dos inteiros módulo M, pertencentes a Z_n, para modulação M-PSK.

Definição 6.3.1: Será definida como operador ⊞ para modulação MPSK à soma entre um vetor complexo z e um símbolo q-ário $p \in Z_q$:

$$z \boxplus p = |z| \, e^{j(\beta+\varphi\gamma)} \qquad (6.3.1)$$

onde

$p = e^{j\varphi\gamma}$ e $z = |z|\ e^{j\beta}$

sendo $\varphi = 2\pi/M$ e $\gamma = 0,1,2,\ldots,M-1$

Da definição 6.3.1 pode ser visto que a operação $\boxplus$ provoca uma rotação do vetor z em torno da origem das coordenadas.

É conveniente definir, para a modulação MPSK, transformações diretas sobre as coordenadas da mesma maneira que foram feitas para BPSK, ou seja usando a simetria do sistema de coordenadas. É interessante notar que uma modulação MPSK pode ser vista como uma combinação de duas modulações (M-1)PSK defasadas de um ângulo φ. Assim, um modo de se realizar as transformações impostas pela operação $\boxplus$ consiste em rotacionar os eixos das coordenadas de maneira que possa ser utilizada a simetria do novo sistema de coordenadas. Uma rotação dos eixos no sentido anti-horário em um ângulo φ resulta em novas coordenadas x' e y' facilmente calculadas através das seguintes equações:

$$x' = x \cos \varphi + y \operatorname{sen} \varphi$$

$$y' = -x \operatorname{sen} \varphi + y \cos \varphi$$

Através destas equações pode ser calculada uma tabela de conversão da soma $\boxplus$ para uma modulação MPSK.

Particularmente, para um sistema 4PSK, esta soma dá origem às seguintes transformações:

INTEIROS DE Z_4		COORDENADAS DO PONTO RECEBIDO		RESULTADO DA TRANSFORMAÇÃO
0	⊞	(X,Y)	=	(X,Y)
1	⊞	(X,Y)	=	(-Y,X)
2	⊞	(X,Y)	=	(-X,-Y)
3	⊞	(X,Y)	=	(Y,-X)

Tab. 6.3.1 Soma ⊞ para modulação 4PSK.

Para a modulação 8PSK, tem-se que $\varphi = 45^{\circ}$ e a tabela de conversão de coordenadas é mostrada na tabela 6.3.2.

INTEIROS DE Z_8		COORDENADAS DO PONTO RECEBIDO		RESULTADO DA TRANSFORMAÇÃO
0	⊞	(x,y)	=	(x,y)
1	⊞	(x,y)	=	((√2/2)(x-y) , (√2/2)(x+y))
2	⊞	(x,y)	=	(-y,x)
3	⊞	(x,y)	=	((√2/2)(-x-y) , (√2/2)(x-y))
4	⊞	(x,y)	=	(-x,-y)
5	⊞	(x,y)	=	((√2/2)(-x+y),(√2/2)(-x -y))
6	⊞	(x,y)	=	(y,-x)
7	⊞	(x,y)	=	((√2/2)(x+y) , (√2/2)(-x+y))

Tab. 6.3.2 Soma ⊞ para modulação 8PSK.

Pode-se notar que a aplicação da transformação "soma $\boxplus$ " dá origem à um grupo de reflexão em relação a determinados planos que passam pelo centro das coordenadas. Este é um caso particular de uma reflexão no espaço Euclidiano R^n. Uma reflexão no espaço Euclidiano R^n é uma transformação linear R, que leva cada vetor em sua imagem refletida em relação a um hiperplano fixo P [14].

6.3.3 Cálculo de pesos e confiabilidade do sinal.

Outro item importante no desenvolvimento do algoritmo é o cálculo do peso da seqüência recebida. Este peso será definido da mesma maneira que para BPSK, ou seja, é a distância Euclidiana entre a seqüência recebida e a palavra-código zero.

Nesta seção será introduzida ainda a definição de confiabilidade de sinal.

Definição 6.3.2: Confiabilidade relativa. No espaço de sinais o ponto r_1 é mais confiável que r_2 se

$$\max_i p(r_1 / s_i) > \max_i p(r_2 / s_i) \qquad (6.3.2)$$

onde $p(r/s)$ é a probabilidade condicionada de r dado s, e s_i é um símbolo no espaço de sinais .

Para ruído aditivo com distribuição monotonicamente decrescente e simétrica, tem-se:

$$p(r_1/s_i) > p(r_2/s_i) \Rightarrow \| r_1 - s_i \| < \| r_2 - s_i \|$$

Portanto a equação 6.3.2 é equivalente a

$$\min_i \| r_1 - s_i \| < \min_i \| r_2 - s_i \| \qquad (6.3.3)$$

Assim, neste caso, r_1 é mais confiável que r_2.

A figura 6.3.2 ilustra um exemplo de confiabilidade relativa para 4PSK. A área hachurada representa a região de decisão do símbolo s_o.

Outra maneira de tratar a confiabilidade relativa de um elemento em um espaço com modulação MPSK foi proposta por Williams e Farrell [53], em que assume-se que o demodulador, através do controle automático de ganho (CAG), entrega ao decodificador elementos com módulo unitário. Isto será ilustrado através de um exemplo, conforme a figura 6.3.3.

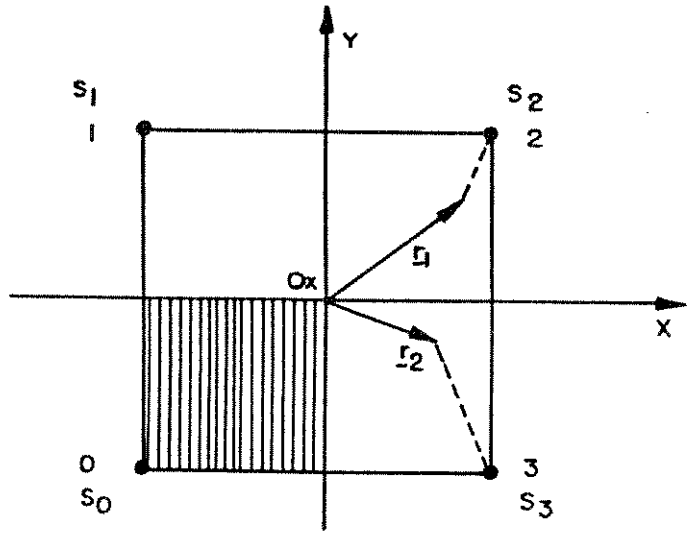


Fig. 6.3.2 Confiabilidade do vetor recebido e regiões de decisão para modulação 4PSK.

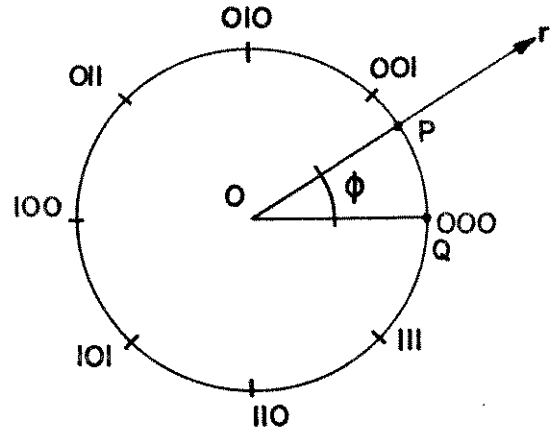


Fig.6.3.3 Confiabilidade relativa para 8-PSK.

Neste exemplo r é um ponto recebido e projetado na circunferência originando, desta maneira, o ponto P. Como pode ser visto na figura 6.3.3, ao ponto P corresponde um ângulo ϕ = $\angle$QOP. Esta correspondência pode ser generalizada para qualquer ponto recebido r. Assim, para o caso particular de 8-PSK, tem-se que a cada ponto recebido r corresponde um número real ϑ conforme a equação 6.3.4.

$$r \longrightarrow \vartheta = \frac{4\,\phi}{\pi} \qquad (6.3.4)$$

Desta maneira, o vetor recebido $\underset{\sim}{r}$ é mapeado em uma seqüência de valores de ϑ. A informação contida em ϑ e a normalização do vetor recebido pelo CAG, fazem com que o espaço bidimensional possa ser tratado em uma dimensão. Como a informação contida no módulo do vetor foi desprezada, fica claro que o sistema de decodificação construído desta forma é sub-ótimo.

6.3.4. Limiar GMD .

Da mesma forma que foi feito para a modulação BPSK, serão utilizados testes de limiares para acelerar o processo de decodificação. De acordo com os resultados obtidos na simulação em computador e, para simplificação do algoritmo, serão utilizados apenas os testes de síndrome e do limiar GMD.

Em [18] é apresentado um desenvolvimento do limiar GMD para códigos não binários. Cada símbolo q-ário recebido pelo demodulador é definido pelo seu valor estimado y_i, que é um elemento de GF(q), e por um número α_i, $0 \leq \alpha_i \leq 1$ que representa a confiabilidade do valor estimado. A confiabilidade é tanto maior quanto mais próximo de 1 α_i se encontrar. Como mencionado em [18] e [3], para o caso não binário não existe uma interpretação geométrica do limiar GMD, como a que foi feita para GF(2).

Como no caso binário, o vetor recebido $\underset{\sim}{y}$ é inicialmente normalizado com componentes entre -1 e 1, porém com o produto interno definido por

$$\underset{\sim}{y} \cdot \underset{\sim}{c}_r = \sum_{i=1}^{n} \alpha_i \, \delta(y_i, c_{ri}) \quad \text{onde} \quad \delta(y_i, c_{ri}) = \begin{cases} +1 \text{ para } y_i = c_{ri} \\ -1 \text{ para } y_i \neq c_{ri} \end{cases} \tag{6.3.5}$$

Com o produto interno definido desta maneira, em [18] é demonstrado o Teorema do GMD para códigos não binários. Assim, o Teorema do limitante GMD permanece inalterado, ou seja, existe quando muito uma palavra-código $\underset{\sim}{c}_r$ do código sobre GF(q) de comprimento n e distância mínima d_{Hmin}, para o qual

$$\underset{\sim}{y} \cdot \underset{\sim}{c}_r > n - d_{Hmin}.$$

Desta maneira, para a região de Voronoi da palavra-código zero têm-se o seguinte teste:

$$\sum_{i=1}^{n} \alpha_i \delta (y_i , c_{oi}) > n - d_{Hmin} \qquad (6.3.6)$$

. **Definição 6.3.3: Confiabilidade absoluta.** Seja uma constelação MPSK com raio unitário. Seja d a distância Euclidiana entre um símbolo recebido e o ponto da constelação mais próximo do mesmo. Será definida como confiabilidade absoluta α o valor $\alpha = d - 1$.

É necessário enfatizar que, de acordo com [3], **a notação $\underset{\sim}{y} . \underset{\sim}{c}_r$ é conveniente porém imprecisa,** uma vez que $\underset{\sim}{y} . \underset{\sim}{c}_r$ depende também da confiabilidade absoluta de cada símbolo, α_i. No algoritmo proposto a seguir será utilizada a equação (6.3.5) em conjunto com a definição (6.3.3) por apresentarem facilidades computacionais.

6.3.5. Algoritmo proposto para decodificação de códigos multiníveis sobre o anel de inteiros.

A seguir, é apresentado um algoritmo onde é feita a incorporação do algoritmo de apagamento ("erasure") ao algoritmo de conjuntos de informação modificado para códigos multiníveis sobre o anel de inteiros, com modulação MPSK.

Para um melhor entendimento do algoritmo proposto é conveniente lembrar que um código de bloco com distância mínima de Hamming d_{Hmin} é capaz de corrigir qualquer padrão de t ou

menos erros se $d_{Hmin} \geq 2t + 1$. É importante lembrar também que um código de bloco de distância $d_{Hmin} \geq 2l + s$ é capaz de corrigir qualquer padrão de l erros e s apagamentos.

Um procedimento genérico para a decodificação de código de bloco binário com apagamento é o seguinte [3]:

1-Apague o bit menos confiável e o substitua por zero ou por um de maneira a satisfazer as n-k equações de paridade.

2- Se for impossível satisfazer as n-k equações de paridade com um apagamento, tente combinações com dois ou mais apagamentos de maneira que o número de apagamentos s não ultrapasse o valor n-k. Pare quando as equações de paridade forem satisfeitas. A palavra-código decodificada é então aquela que satisfez as equações de paridade. Se, dadas as condições mencionadas, as equações de paridade não puderem ser satisfeitas então tem-se uma falha de decodificação.

No algoritmo aqui proposto o apagamento é feito da seguinte maneira: quando a palavra-código gerada por um conjunto de informação não reduzir o peso analógico do vetor $\underset{\sim}{y}$, são feitas perturbações sucessivas nos s símbolos menos confiáveis (apagados) dentre os k símbolos do conjunto de informação utilizado. Em seguida tenta-se novamente uma redução de peso com os novos vetores obtidos, utilizando o mesmo conjunto de informação. Desta maneira aumenta-se o número de palavras geradas pelos conjuntos de informação.

O algoritmo será apresentado na forma de rotina, da seguinte maneira:

$\underset{\sim}{y}$ = vetor recebido normalizado;

$\underset{\sim}{\bar{y}}$ = vetor recebido com decisão abrupta;

$\underset{\sim}{c}_o$ = palavra-código zero em C;

s = número de apagamentos utilizado;

N_o = coleção de vizinhos de zero;

I = numero máximo de conjuntos de informação a ser utilizado;

i = 1;

$\underset{\sim}{\beta} = \underset{\sim}{c}_o$;

j=1;

Se $S = \bar{\underset{\sim}{y}}.\ H^T = \underset{\sim}{0}$, então decodifique $\underset{\sim}{y}$ como sendo a palavra-código $\bar{\underset{\sim}{y}}$. Caso contrário, faça $\underset{\sim}{z} = \underset{\sim}{y}$ e continue.

1- Tome o i-ésimo conjunto de informação da coleção N_I e forme uma palavra-código $\underset{\sim}{\alpha}$ a partir de $\underset{\sim}{z}$. Se

$$W(\underset{\sim}{z}) > W(\underset{\sim}{z} \boxplus \underset{\sim}{\alpha}), \qquad (6.3.7)$$

vá para o passo 2. Caso contrário continue.

Faça perturbações nos s símbolos menos confiáveis do conjunto de informação atual, de maneira a criar $j_{max} = 2s$ vetores. Para j=1, substitua $\underset{\sim}{z}$ pelo vetor criado.

Faça j=j+1.

Se $j \leq j_{max} = 2s$, vá para o passo 1.

Faça i = i + 1. Se i > I, vá para o passo 3. Caso contrário, retorne ao passo 1.

2- Faça

$$\underset{\sim}{z} = \underset{\sim}{z} \boxplus \underset{\sim}{\alpha} \qquad (6.3.8)$$

$$\underset{\sim}{\beta} = \underset{\sim}{\beta} + \underset{\sim}{\alpha}$$

Se $\underset{\sim}{z}$ satisfaz o limiar GMD com $\underset{\sim}{c} = \underset{\sim}{c}_o$, vá para o passo 5.

Caso contrário, faça i = i + 1.

Se i > I, vá ao passo 3. Caso contrário retorne ao passo 1.

3- Encontre $\underset{\sim}{\alpha} \in N_o$ tal que :

$$W(\underset{\sim}{z}) > W(\underset{\sim}{z} \boxplus \underset{\sim}{\alpha}) \qquad (6.3.9)$$

Se tal α não existir vá ao passo 5. Caso contrário prossiga.

4- Faça

$$\underset{\sim}{z} = \underset{\sim}{z} \boxplus \underset{\sim}{\alpha} \qquad (6.3.10)$$

$$\underset{\sim}{\beta} = \underset{\sim}{\beta} + \underset{\sim}{\alpha}$$

Se $\underset{\sim}{z}$ não satisfaz o limiar GMD com $\underset{\sim}{c} = \underset{\sim}{c}_o$, vá para o passo 3.

5- Decodifique $\underset{\sim}{y}$ como sendo a palavra-código $\underset{\sim}{\beta}$.

Os resultados de simulação em computador para o código plésio-cíclico multinível C(4;4,16,2) definido em [1] estão mostrados na tabela 6.3.3 e na fig.6.3.4 . Na generalização do ZNA para decisão suave (ZNA-G) foi utilizado um conjunto de 6 palavras de peso mínimo de Hamming.

A tabela 6.3.4 e a figura 6.3.5 ilustram o desempenho da simulação feita em computador para o código plésio-cíclico multinível C(4;10,1024,4) definido em [1]. Para o ZNA-G foi utilizada uma lista contendo 84 palavras-código, sendo 16 palavras de peso 4 e 68 palavras de peso 5. Foi utilizado s=1 para ambos os códigos.

E_b/N_o (dB)		1	2	3	4
ALGORITMO DE CORREL. COMPLETA	TEB	0,0596	0,0357	0,0150	0,0055
ACIM	TEB	0,0782	0,0529	0,0300	0,0152
	Nº MÉD. CORREL/PAL	2,60	2,30	1,92	1,50
ACIM COM PERTURB.	TEB	0,0627	0,0402	0,0196	0,0093
	Nº MÉD. CORREL/PAL	4,34	3,87	3,20	2,50
ACIM + ZNA	TEB	0,0664	0,0416	0,0187	0,0090
	Nº MÉD. CORREL/PAL	6,70	5,97	4,89	3,80
ACIM COM PERTURB. + ZNA	TEB	0,0596	0,0357	0,0150	0,0060
	Nº MÉD. CORREL/PAL	8,30	7,40	6,10	4,75

Tab. 6.3.3 Desempenho dos algoritmos com conjuntos de informação para o código (4,2) .

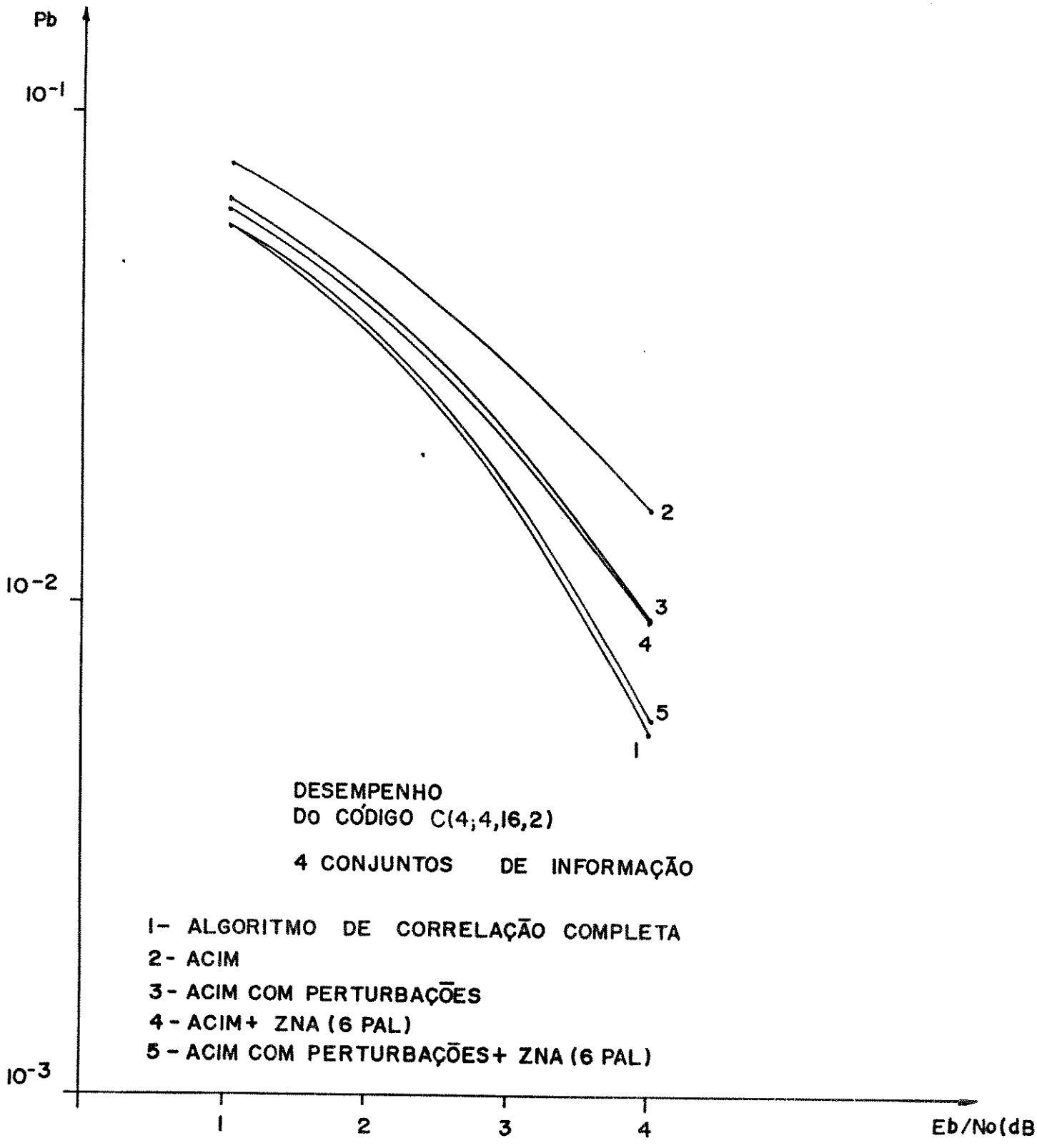


Fig. 6.3.4 Desempenho dos algoritmos propostos para o código C(4;4,16,2), proposto em [1] ,(s=1).

E_b/N_o (dB)		1	2	3	4
ALGORITMO DE CORREL. COMPLETA	TEB	0,0528	0,0211	0,0078	0,0011
ACIM	TEB	0,0658	0,0313	0,0124	0,0036
	Nº MÉD. CORREL/PAL	23,86	22,44	20,30	16,67
ACIM COM PERTURB.	TEB	0,0568	0,0245	0,0078	0,0017
	Nº MÉD. CORREL/PAL	45,72	43,00	38,90	31,90
ACIM + ZNA	TEB	0,0564	0,0238	0,0113	0,0022
	Nº MÉD. CORREL/PAL	103,91	96,34	86,11	70,03
ACIM COM PERTURB. + ZNA	TEB	0,0528	0,0229	0,0086	0,0014
	Nº MÉD. CORREL/PAL	123,84	115,43	104,07	85,00

Tab. 6.3.4 Desempenho dos algoritmos com conjuntos de informação para o código (10,5).

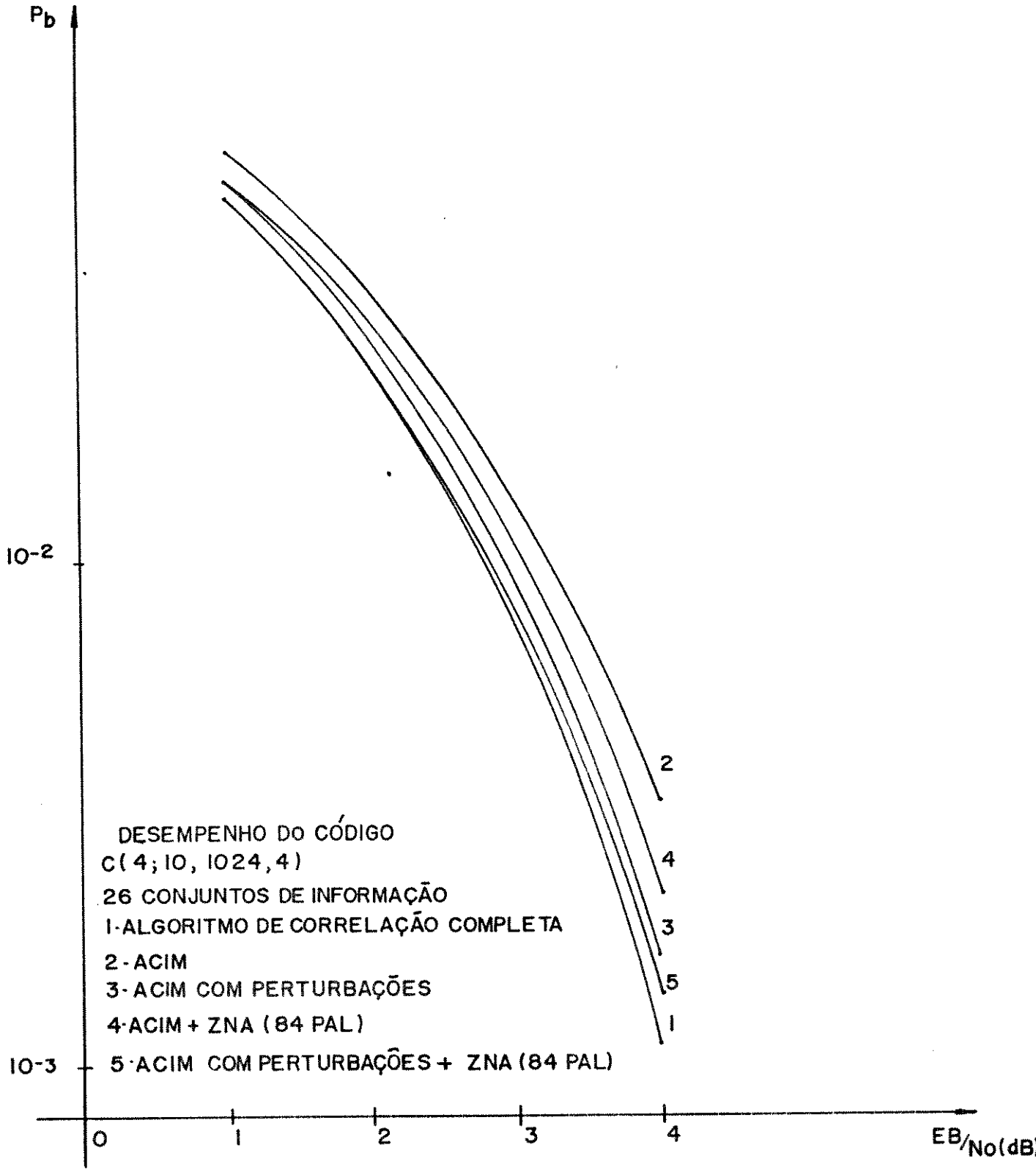


Fig. 6.3.5 Desempenho dos algoritmos propostos para o código C(4;10,1024,4), (s=1).

6.4 ANÁLISE DOS CÓDIGOS MULTINÍVEIS PROPOSTOS EM [1], ATRAVÉS DE RETICULADOS.

Nesta seção será apresentada uma abordagem via reticulados da decodificação dos códigos multiníveis propostos em [1].

Os códigos multiníveis descritos em [1] foram obtidos através da geração exaustiva de todas as possíveis linhas de G, sendo que para cada uma foi calculada a distância mínima do código através do cálculo de peso de uma palavra de peso mínimo.

Um exemplo dado em [1] e que será utilizado neste trabalho para ilustrar a decodificação desta classe de códigos multiníveis através das propriedades de reticulados, é o código otimizado C(4;4,16,2) no anel Z_4, para uma modulação 4-PSK, cuja matriz geradora pode ser colocada na forma

$$G = \begin{vmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{vmatrix} \qquad (6.4.1)$$

Este código é definido basicamente pelo vetor 1210. A exemplo do que foi feito no capítulo 3, este vetor pode ser representado em binário da seguinte maneira:

$$\begin{matrix} 1010 \\ 0100 \end{matrix} \qquad (6.4.2)$$

Neste exemplo, pode ser notado que a estrutura deste código corresponde ao esquema genérico de concatenação generalizada visto no capítulo 3. Mais especificamente, se escrito em forma de arranjo de coordenadas, de acordo com o capítulo 2, este código pode ser formado por dois códigos binários:

-um para definir um subconjunto,

-outro que, em função do subconjunto escolhido, codifica os bits binários restantes.

Isto pode ser melhor entendido através da fig. 6.4.1.

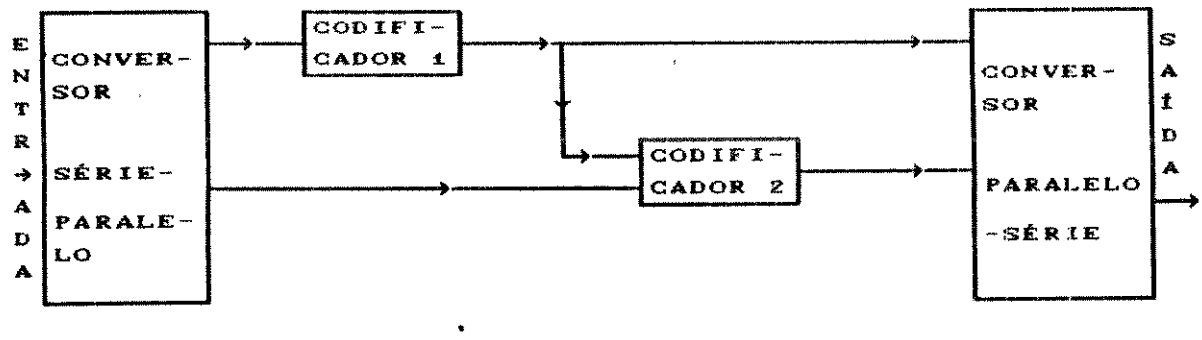


Fig. 6.4.1 Esquema de codificação para uma classe de códigos multiníveis com modulação 4PSK.

Para o caso específico do código multinível C(4;4,16,2) visto, tem-se 4 subconjuntos definidos pelo código binário C(2;4,4,2). O codificador 1 codifica o primeiro trem de bits de acordo com este código. O codificador 2 codifica os bits restantes dentro do subconjunto definido pelo codificador 1. O código C(4;4,16,2) possui as seguintes propriedades que poderão ser utilizadas para a sua decodificação:

-a soma dos símbolos de cada palavra-código é um número divisível por quatro;

-se a palavra do código binário C(2;4,4,2) que define o subconjunto possuir peso de Hamming igual ou menor que dois, então a soma dos símbolos será igual ou menor que quatro. Caso contrário, esta soma será igual a 8.

Assim, por exemplo, os símbolos de informação 12 na entrada do esquema da figura 6.4.1 serão divididos em dois trens de bits

pelo conversor série-paralelo: **10** e **01**. O codificador 1 codificará o primeiro conjunto de informação na palavra-código **1010**. O codificador 2 codificará o segundo conjunto de informação em função da palavra-código obtida pelo codificador 1 de maneira que os bits de paridade serão obtidos pela soma módulo 2 entre os bits de informação **01** com os terceiro e quarto bits da palavra-código obtida pelo codificador 1. A palavra codificada pelo codificador 2 é então **0100**. O conversor paralelo-série formará finalmente a palavra-código multinível **1210**. Assim o codificador pode ser analisado de maneira semelhante ao do apresentado por Imai e Hirakawa [29], com a particularidade na formação dos códigos binários e na ligação entre os mesmos.

O mapeamento na constelação 4PSK pode ser feito da seguinte maneira:

símbolos	**coordenadas**
0	(1,0)
1	(0,1)
2	(-1,0)
3	(0,-1)

A distância Euclidiana mínima é facilmente calculada, levando-se em consideração as palavras-código zero e de peso mínimo 1 2 1 0. Assim, de acordo com o mapeamento, tem-se que :

$$d_E^2 = d_E^2(1,0) + d_E^2(2,0) + d_E^2(1,0) + d_E^2(0,0) = 2 + 4 + 2 + 0 = 8$$

Utilizando as propriedades acima mencionadas do arranjo definido pelo código multinível C(4;4,16,2), pode ser feita a decodificação do referido código. O arranjo obtido neste caso é

uma típica construção A.

Assim, dada a estrutura de codificação da fig.6.4.1, pode ser utilizado o esquema de decodificação descrito em [53]. Para o esquema de codificação da fig.6.4.1, a complexidade do algoritmo descrito em [53] é proporcional ao número de subconjuntos da primeira partição, ou seja, é proporcional a $2^{\Sigma k_i}$. Isto também é válido para o algoritmo de decodificação sub-ótimo proposto por Sayegh [44] e mencionado no capítulo 3 deste trabalho.

6.5 CONCLUSÕES.

Neste capítulo foi apresentada uma generalização para modulação MPSK dos algoritmos de decodificação propostos nos capítulos 4 e 5 para o caso binário. Foram também propostas variantes dos algoritmos desenvolvidos onde se incorpora o algoritmo de apagamento (erasure).

Os resultados de simulação em computador apresentados para os códigos multiníveis propostos em [1] evidenciam a aplicação dos algoritmos apresentados. A variante que mais se destacou pelo compromisso entre complexidade e desempenho foi a do algoritmo onde se empregam perturbações em alguns símbolos menos confiáveis conjugadas com o algoritmo ACIM generalizado para códigos multiníveis.

Foi também apresentada uma análise dos códigos multiníveis propostos em [1] baseada no particionamento dos mesmos em códigos binários. É importante deixar claro que esta análise foi feita com o objetivo de mostrar que os mesmos podem ser gerados de acordo com o esquema da figura 6.4.1. As propriedades observadas da geração de tais códigos através do esquema da figura 6.4.1 podem auxiliar na decodificação dos mesmos através de esquemas

semelhantes aos propostos em [53] ou [44] conjugados, por exemplo, com os algoritmos propostos nos capítulos 5 e 6 deste trabalho. Por outro lado, é conveniente notar que uma decodificação dos códigos multiníveis propostos em [1], fundamentada somente nas propriedades do particionamento do esquema da figura 6.4.1, será dificultada pela pequena separação entre os subconjuntos de palavras-código, ditada pelos parâmetros do codificador 1. Neste caso o esquema de codificação de Imai-Hirakawa [29] fornece melhores resultados. De acordo com o esquema de Imai-Hirakawa analisado no capítulo 3 deste trabalho (fig. 3.4.3), tem-se, por exemplo, que para codificar 4 bits de informação podem ser utilizados dois códigos binários: $C_1(2;4,2,4)$ que é um código de repetição e $C_2(2;4,8,2)$, que é um código de um único dígito de paridade. Com estes dois códigos obtem-se somente dois subconjuntos de oito palavras-código distanciados entre sí de $d_H = 4$ enquanto que o esquema da figura 6.4.1 apresenta quatro subconjuntos distanciados entre sí de $d_H = 2$. Assim, se forem utilizadas somente as propriedades dos reticulados fica evidenciada a vantagem do esquema de Imai-Hirakawa em relação ao esquema proposto pela figura 6.4.1.

Deixa-se como sugestão, para futuros trabalhos, a busca de códigos multiníveis utilizando o esquema proposto na figura 6.4.1, bem como a elaboração de algoritmos de decodificação para os mesmos.

CAPÍTULO 7

CONSIDERAÇÕES FINAIS.

O presente trabalho procurou explorar dois tópicos de grande interesse atual na área de codificação: concatenação generalizada e decodificação de códigos de blocos com decisão suave.

Foram desenvolvidos e avaliados algoritmos para decodificação em esquemas de concatenação generalizada com modulação codificada. A análise de sistemas concatenados foi desenvolvida usando-se como ferramenta a teoria de reticulados.

Foi apresentado um algoritmo fundamentado nos algoritmos de conjuntos de informação e de vizinhos de zero, tanto para códigos binários quanto para códigos multiníveis definidos sobre um anel. Os resultados de simulação em computador mostraram o caráter adaptativo do sistema de decodificação proposto, bem como uma baixa complexidade mesmo para códigos mais longos.

A implementação do algoritmo proposto para o código de Golay C(2;23,4096,7) foi realizada em circuito microprocessado, utilizando-se o microprocessador 8086. A versão inicial compreende somente o algoritmo de Conjuntos de Informação Modificado e faz uso de 100 conjuntos de informação selecionados de modo a abranger o maior número possível de padrões de erros. O protótipo é constituído de 23 CI's, sendo 11 para o decodificador propriamente dito e o restante para os circuitos de entrada e saída de dados. O desempenho do protótipo do decodificador confirmou os resultados de simulação, onde se obteve um desempenho próximo ao do decodificador de máxima verossimilhança.

Para uma relação sinal/ruído (Eb/No) de 5 dB, a taxa de transmissão máxima obtida no teste do circuito foi de 2KBPS, ao passo que para 10 dB a taxa de transmissão máxima atingida situou-se em torno de 5KBPS.

Como desenvolvimento futuro deste trabalho, sugere-se:

1- a aplicação do algoritmo de decodificação suave parcial, desenvolvido no capítulo 3, para códigos mais longos;

2- a implementação de algoritmos que utilizem o particionamento alternado proposto no capítulo 3;

3- a decodificação dos códigos multiníveis apresentados em [1] através do particionamento de tais códigos, proposto no capítulo 6;

4- a elaboração de limitantes mais apertados para a localização de um vetor dentro da região de Voronoi da palavra-código zero.

5- a utilização do algoritmo proposto no capítulo 6 para a decodificação de códigos de Reed-Solomon.

6- a utilização dos algoritmos propostos nos capítulos 4, 5 e 6, onde se usariam conjuntos de informação escolhidos com o auxílio da confiabilidade dos símbolos do vetor recebido.

REFERÊNCIAS BIBLIOGRÁFICAS.

1. BALDINI, R.F. Códigos Pseudo-Cíclicos Multiníveis. 7º *Simpósio Brasileiro de Telecomunicações*. Florianópolis, SC, Set.1989

2. BATTAIL, G., DECOUVELAERE M.C. and GODLEWSKI, Replication Decoding P. *IEEE Trans. on Inf. Theory*, vol. IT-25, No 3 May 1979.

3. BLAHUT, R.E. *Theory and Practice of Error Control Codes*, Addison-Wesley Pub. Co., Canada, 1984.

4. BLOKH, E.L. and V.V. ZYABLOV Coding of Generalized Concatenated Codes, Problemy *Peredatchi Informatsii*, vol. 10, No 3, July-September 1974, pp-45-50.

5. BOSSERT, M. Concatenation of Block Codes, *Report, DFG (Deutsche Forschungsgemeinschaft)* August-1987 to April 1988.

6. BRINE, A.,P.G.FARRELL. Study of Reduced Complexity Concatenated Coding Schemes, *Final Report to ESA/ESTEC, Univ. Manchester*, June, 1988.

7. CHASE, D. A Class of Algorithms for Decoding Block Codes With Channel Measurement Information, *IEEE Trans. on Inf. Theory*, vol IT-18 Jan 1972.

8. CLARK, G.C., Jr.J.B.CAIN, *Error-Correction Coding for Digital Communications*, Plenum Press, New York, 1981.

9. CONWAY, J.H, N.J.A. SLOANE. A Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes *IEEE Trans. on Inf. Theory*, vol. IT-28, No 2,March. 1982.

10. CONWAY, J.H, N.J.A. SLOANE. Voronoi Regions of Lattices, Second Moments of Polytopes, and Quantization, *IEEE Trans. on Inf. Theory* , vol. IT-28, No 2 March 1982.

11. CONWAY, J.H, N.J.A. SLOANE. A Fast Encoding Method for Lattice Codes and Quantizers, *IEEE Trans. on Inf. Theory*, vol. IT-29, Nov. 1983.

12. CONWAY, J.H, N.J.A. SLOANE. *Sphere, Packings, Lattices and Groups*, Springer-Verlag, N.Y., 1988.

13. CONWAY, J.H, N.J.A. SLOANE. Soft Decoding Techniques for Codes and Lattices, Including the Golay Code and the Leech Lattice, *IEEE Trans. on Inf. Theory* ,vol. IT-32, No 1, January 1986.

14. COXETER, H.S.M. *Regular Polytopes*. Dover, New York, 1973.

15. FARRELL, P.G., M. RICE, F. TALEB. Minimum Weight Decoding for Cyclic Codes, *IMA Conference on Cryptography and Coding*, Cirencester, Dec. 15-17th, 1986.

16. FARRELL, P.G., M. RICE, F. TALEB. Division Algorithms for Hard and Soft Decision Decoders, *International Conference on Digital Signal Processing, Florence, Italy.*, Sept.7-10, 1987.

17. FORNEY, G.D Jr. Generalized Minimum Distance Decoding, *IEEE Trans. on Inf. Theory*, IT-12, 1966.

18. FORNEY, G. D. Jr. *Concatenated Codes*, Cambridge, Massachusetts, MIT Press, 1966.

19. FORNEY, G. D, Jr.,R.G. GALLAGER, R.L. GORDON, M.L. FRED, and S. QURESHI. Efficient Modulation for Band-Limited Channels. *IEEE Journal on Selected Areas in Communications*, vol, Sac-2, No. 5, September, 1984.

20. FORNEY, G. D, Jr. Coset Codes-Part I: Introduction and Geometrical Classification, Part II: Binary Lattices and Related Codes. *IEEE Trans. on Information Theory*, vol. 34 No 5 September 1988.

21. GODOY, W. Jr., D.S. ARANTES. Sub-Optimum Soft-Decision Decoding of Block Codes Using the Zero-Neighbors Algorithm, *IEEE Int. Symposium on Inf. Theory*, Kobe, Japan, 1988.

22. GODOY, W.Jr.,D.S. ARANTES. Decodificação de Códigos de Bloco com Decisão Suave e Correlação Incompleta. *7° Simpósio da SBT, Florianópolis, SC*, 1989.

23. GODOY W. Jr., D.S. ARANTES. An Algorithm for Soft-Decision Minimum Weight Decoding of Block Codes, *International Simposium on Signals, Systems and Electronics, Erlangen, Germany*, Sept. 18-20, 1989.

24. GODOY W. ,Jr.,D.S.ARANTES. Um Sistema de Decodificação que Utiliza Um Algoritmo de Peso Mínimo Com Decisão Suave *3ª Reunion En Procesamiento de La Informacion RPI' 89* La Plata, Argentina, 1989.

25. GODOY W.Jr., R. Jr PALAZZO. Construções de Códigos Concatenados Generalizados Via Reticulados e Decodificação Suave Parcial. *7° Simpósio da SBT*, Florianópolis, SC, 1989.

26. GODOY, W. Jr., W.VALENTE, I.LAZIER. Implementação de um Decodificador Adaptativo para Códigos Corretores de Erro. *Trabalho premiado no INFOPAR-89*, Curitiba, PR, 1989.

27. GREENBERGER, H. An Iterative Algorithm for Decoding Block Codes Transmitted over a Memoryless Channel, *J.P.L. DSN Progress Report 42-47*, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, July/Aug. 1978.

28. HARTMANN, C.R.P.,L. D. RUDOLPH. An Optimum Symbol by Symbol Decoding rule for Linear Codes, *IEEE Trans. on Inf. Theory*, vol. IT-22, Sept. 1976.

29. IMAI, H., S. HIRAKAWA. A New Multilevel Coding Method Using Error-Correcting Codes, *IEEE Trans. on Inf. Theory*, vol.IT-23, May 1977.

30. LEVITIN, L.B., C.R.P. HARTMANN. A New Approach to the General Minimum Distance Decoding Problem: The Zero-Neighbors Algorithm. *IEEE Trans. on Inf.Theory*, vol. IT-31, No. 3, May 1985.

31. MacWILLIAMS, F.J., N. J. A. SLOANE. *The theory of Error Correcting Codes*, New York: North Holland, 1977.

32. MASSEY, J. L. *Theshold Decoding* . Cambridge. Massachusetts: MIT Press. 1963.

33. MASSEY, J.L. Coding and Modulation in Digital, Communication Zürich Seminar, 1974.

34. MATIS, K. R.,J. M. MODESTINO. Reduced-Search Soft-Decision Trellis Decoding of Linear Block Codes, *IEEE Trans. on Inf. Theory*, vol. IT-28, March 1982.

35. NILSSON, J., I. KEREKES. Comparison Between Different Soft Decoding Algorithms. *Department of Electrical Engineering. Linköping University.*

36. OMURA, J.K. *A Probabilistic Decoding Algorithm for Binary Group Codes.* Stanford Research Institute, Menlo Park, California, March 1969.

37. PADOVANI, R., J, K. WOLF. Coded Phase/Frequency Modulation. *IEEE Trans. on Com.*, vol. Com-34, No. 5, May, 1986.

38. POTTIE, G. J., D. P. TAYLOR. Multilevel Codes Based on Partitioning. *IEEE Trans. on Inf. Theory*, vol. 35, No. 1, January, 1989.

39. PORTNOY, S. L. Characteristics of Coding and Modulation Systems from the Standpoint of Concatenated Codes *Problemy Peredatchi Informatsii*, vol.21, No 3, July-Sept.

40. PRANGE, E. The Use of Information Sets in Decoding Ciclic Codes. *IRE Trans. on Inf. Theory* IT-8, S5-S9, 1962.

41. PROAKIS, J.G. *Digital Communications*, McGraw Hill,1983

42. RICE, M., D.J. TAIT, P.G.FARRELL. A Soft-Decision Reed-Solomon Decoder *IEEE Int. Symposium on Inf. Theory*, Kobe, Japan, 1988.

43. SAGASTUME, M.,B. GABRIEL. *Problemas, Lenguajes y Algoritmos*, Campinas, Editora da Unicamp, 1986.

44. SAYEGH, S. I. A Class of Optimum Block Codes in Signal Space *IEEE Trans. on Communications*, Vol. Com-34 1986.

45. SLOANE, N.J.A. Sphere Packings Constructed from BCH and Justesen Codes, *Mathemátika 19* , 1972.

46. SLOANE, N.J.A. Binary Codes, Lattices, and Sphere-Packings In: P.J. CAMERON (ed.) *Commbinatorial Surveys.* Academic, London and N.Y., 1977.

47. SLOANE, N.J.A. Tables of Sphere-Packings and Spherical Codes *IEEE Trans. Inf. Theory*, vol. IT-27 N 3 May 1981.

48. SCHONHEIN, J. On Coverings, *Pac. J. Math.*, 14, 1964.

49. TALEB, F.,P.G. FARRELL. Soft Minimum Weight Decoding of Reed-Solomon Codes, *IEEE International Simposium on Information Theory*, Kobe, Japan, June, 1988.

50. UNGERBOECK, G. Channel Coding with Multilevel/Phase Signals. *IEEE Trans. on Inf. Theory*, vol. IT-28, No. 1, January, 1982.

51. VOUKALIS, D.C. A New Series of Concatenated Codes Subjected to Matrix Type-*B* Codes, *IEEE Trans. on Inf. Theory*, vol. IT-28, No 3, May 1982.

52. WELDON, E.J.Jr. Decoding Binary Block Codes on Q-ary Output Channels. *IEEE Trans. on Inf. Theory* IT-17, November 1971.

53. WILLIAMS R.G.C., P.G. FARRELL. Reducing the Decoding Complexity of Block Coded Modulation Schemes, *MIEEE, Communications Research Group. Department of Electrical Engineering, University of Manchester. M13 9PL*, (1988).

54. WOLF, J.K. Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis, *IEEE Trans. on Inf. Theory*, vol IT-24, Jan. 1978.

55. WOZENCRAFT, J.M., I.M. JACOBS. *Principles of Communication Engineering*, John Wiley & Sons, 1965.

56. ZINOVIEV, V.A. Generalized Cascade Codes. *Problemy Peredatchi Informatsii*, vol. 12, 1976.

57. ZINOVIEV, V.A. Generalized Concatenated Codes for Channels with Error Bursts and Independent Errors, *Probl. Inform. Transm. (USA)* vol. 17, No 4 Oct-Dec. 1981.

58. ZINOVIEV,V.A., V. V. ZIABLOV, S.L. PORTNOY. Concatenated Methods for Construction and Decoding of Codes in Euclidean Space. *USSR Academy of Science, Institute for Problems of Information Transmission*, 1987.