UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica

Departamento de comunicações

outubro 1993

CÓDIGOS DE LINHA BINÁRIOS COM CONTROLE DE ERROS.

Este exemplar corresponde à redação final da tese defendida por Eduardo Nisenbaum e aprovada pela Comissão Julgadora em 27 10 1993.

Hélio Waldman
Orientador

Eduardo Nisenbaum

Orientador: Prof. Dr. Hélio Waldman

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

UNICAMP
BIBLIOTECA CENTRAL

Resumo: Nesse trabalho faz-se a investigação de códigos binários que reunem restrições de distância mínima e nulo espectral no DC simultaneamente, aqui chamados de Códigos de linha com controle de erros (CLCEs). Começamos pela busca computacional de códigos de bloco monoalfabéticos e bialfabéticos(Cap. 2). Em seguida investigamos centros e tamanhos de esferas extremas de Hamming no espaço DC restrito com fins de obter limitantes superiores e inferiores para a Taxa $\times$ distância mínima relativa (Cap. 3). O capítulo 4 mostra algumas técnicas construtivas para CLCEs que são implementáveis na prática. Por fim , no capítulo 5, fazemos uma análise de desempenho desses códigos num sistema exemplo: Sistema de comunicações óticas de deteção direta.

Agradecimentos:

Gostaria de agradecer a todos os amigos que com seus incentivos e amizades permitiram que esse trabalho fosse concluido, em especial:

Ao professor Hélio Waldman pela grande dedicação como orientador e pessoa humana.

Ao amigo e professor Celso de almeida pela ajuda e valorosas sugestões.

Aos professores Reginaldo Palazzo Jr. e Michel D. Jacoub pelo apoio durante seus cursos. Ao professor Renato Baldini filho pela confecção de algumas figuras.

Aos amigos Nelson Guimarães Filho , Carlos David Barbosa e demais colegas da Unicamp e CPqD-Telebrás

Finalmente a FAPESP pelo apoio financeiro no último ano da tese.

Dedico essa tese aos meus pais e irmãos

Conteúdo

1 INTRODUÇÃO À CLASSE DOS CÓDIGOS DE LINHA COM CONTROLE DE ERROS **7**

1.1 Histórico 7
1.2 Objetivos da tese 10
1.3 Nomenclatura usada ao longo da tese 12

2 Obtenção de CLCEs por busca exaustiva computacional **15**

2.1 Introdução 15
2.2 Algoritmo de busca completa 15
2.2.1 Tabela de códigos gerados pelo algoritmo de busca completa . 19
2.3 Algoritmo de busca única 19
2.3.1 Algumas observações sobre o algoritmo 20
2.3.2 Tabelas de códigos gerados pelo algoritmo de busca única . . . 20
2.4 Algoritmo de busca única bialfabética 27
2.4.1 introdução aos códigos bialfabéticos 27
2.4.2 O algoritmo 28
2.5 Tabelas de CLCEs bialfabéticos 29
2.6 Comentários sobre os resultados obtidos ao longo do capítulo 31

3 Limitantes na taxa $\times$ d_{min} para CLCEs com restrição DC **35**

3.1 Introdução 35
3.2 Definições e conceitos relevantes 37
3.3 Esferas de Hamming extremas de raio 2 43
3.4 Esferas de Hamming assintóticas : VSDC ímpares 44
3.5 Esferas de Hamming assintóticas: VSDC pares 51
3.6 Assimetrias assintóticas em espaços de Hamming restritos em DC . . 58
3.7 Conclusões 60
3.8 Apêndice A 62

4 ALGUNS MÉTODOS CONSTRUTIVOS PARA CLCEs **64**

4.1 Introdução 64
4.2 Linearidade $\times$ Balanceamento 64
4.2.1 Primeira construção 64
4.2.2 Segunda construção 65

4.2.3 Terceira construção 66
4.2.4 Quarta construção 66
4.3 Códigos "coset" balanceados 67
4.3.1 Introdução 67
4.3.2 Algorítmo de codificação 70
4.3.3 Algorítmo de decodificação 70
4.3.4 Exemplos 70
4.3.5 Melhoria no balanceamento para essa classe de códigos com J=1. 72
4.3.6 Tabela de alguns CLCEs coset balanceados e seus parâmetros 74
4.3.7 Conclusões 74
4.4 Códigos de bloco "fortemente" restritos em VSDC com capacidade de correção de 1 erro 74
4.4.1 Introdução 74
4.4.2 Conceitos envolvidos 74
4.4.3 Transmissão/Codificação 76
4.4.4 Recepção/decodificação(Hard) 77
4.4.5 Busca de um código de maior taxa 78
4.4.6 Conclusões 83
4.5 CLCEs Q_ários (Q > 2) 83
4.5.1 Método de Calderbanc e Mazo 84
4.5.2 CLCEs ternários 84
4.6 Outros métodos 84
4.6.1 Baseados em códigos de controle de erros não lineares 84
4.6.2 Baseados em códigos de treliça binários 84
4.7 Conclusões 85

5 Exemplo de aplicação de CLCEs 88
5.1 Introduçao 88
5.2 Modelamento dos elementos sistêmicos 90
5.2.1 Fonte ótica 90
5.2.2 Fibra ótica monomodo de comprimento L 90
5.2.3 Conversores ótico elétricos 91
5.2.4 Amplificadores/filtros 92
5.3 Equacionamento dos ruídos de receptor 93
5.4 Teoria simplificada para cálculo de ganhos por introdução do CLCE no sistema 94
5.4.1 Introdução 94
5.4.2 Aproximações 94
5.4.3 Cálculo da probabilidade de erro para o decodificador suave . 95
5.4.4 Cálculo do ganho 96
5.4.5 Ganhos a partir de simulação 96
5.4.6 Ganho para Front-End de alta impedância(HZ) a PIN-FET(canal de $1\mu m$) 97

5.4.7 Ganho para Front-End PIN-Transimpedância 97
5.4.8 Ganho de codificação para FRONT-END a APD 98
5.4.9 Perdas de ganho de codificação devido ao efeito de dispersão . 99
5.5 Ganho de codificação em sistemas com decodificação abrupta 100
5.6 Simulação computacional 100
5.6.1 Introdução 100
5.6.2 O sistema de simulação 100
5.6.3 Simulações 101
5.6.4 Determinação do espectro dos códigos e suas frequências de corte inferior 101
5.6.5 Curvas de taxa de erro X potência acoplada com comprimento de fibra nulo 105
5.6.6 Curvas comprimento de fibra X Potência ótica acoplada para taxa de erro 10^{-9} 106
5.7 Conclusões 106
5.7.1 Resultados da teoria 106
5.7.2 Resultados da simulação 107

6 Considerações finais **113**
6.1 Conclusões 113
6.2 Sugestões para trabalhos futuros e comentários finais 114
6.3 Apêndice B - programas de busca exaustiva computacional 116

Capítulo 1

INTRODUÇÃO À CLASSE DOS CÓDIGOS DE LINHA COM CONTROLE DE ERROS

1.1 Histórico

Codificação digital, no sentido amplo, é a técnica de adaptação do sinal gerado na fonte de informação ao canal de transmissão (ou armazenamento), com propósito de subsidiar uma comunicação confiável e eficiente. Confiável no sentido de minimizar erros de transmissão e eficiente no sentido de maximizar a velocidade de transmissão, ambos com nível de complexidade sistêmica aceitável.

A codificação é classicamente dividida em duas modalidades:

- CODIFICAÇÃO DE FONTE: Transforma o sinal "natural" da fonte de informação em uma sequência de digitos binários. O objetivo dos estudos nessa área é minimizar as redundâncias na sequência digital codificada procurando atingir o valor de entropia da fonte "natural" de informação (conforme primeiro teorema de Shannon da teoria da informação [1]).
- CODIFICAÇÃO DE CANAL: Converte a sequência digital proveniente do codificador de fonte em uma forma apropriada para transmissão através do meio em que se dá a comunicação . Basicamente o que se faz é introduzir redundâncias controladas na sequência digital de informação e associar a sequência assim formada à formas de onda que trafegarão pelo canal. Com isso alcançamos uma ou mais melhorias nas características:
  1. Vulnerabilidade à interferência entre símbolos (I.E.S)
  2. Extração de sincronismo de bit (relógio)
  3. Banda ocupada
  4. Imunidade a ruído
  5. Monitoração de taxa de erro

6. Acréscimo de canais de serviço de supervisão e controle
7. Deteção e correção de erros

Devido a grande variedade de meios e sistemas de comunicações existentes, a codificação de canal foi dividida em áreas que foram desenvolvidas separadamente,a citar:

-Códigos de controle de erro [2].
-Moduladores digitais [3].
-Códigos de linha [4].

Comentemos o conteúdo de cada área:

- CÓDIGOS DE CONTROLE DE ERROS (CCE's)-Com a formulação da teoria da informação em 1948, Shannon mostra que com esse tipo de codificação é possível tornar os sistemas robustos com relação ao ruído quando opera-se abaixo da "Capacidade de Canal"(segundo teorema de Shanon). Assim os códigos de controle de erros vem sendo vastamente estudados nas últimas décadas. Em particular, assumir a propriedade de linearidade permite uma abordagem algébrica extensa e elegante para o problema.
  Os CCE's dividem-se em "códigos de bloco" e "códigos convolucionais", e diferenciam-se pela existência ou não de memória nos codificadores/decodificadores. Nesta tese lidamos principalmente com códigos de bloco binários, que se formam através do mapeamento do conjunto das palavras de k bits(mensagem) num subconjunto das palavras de n bits(código)($n > k$). Nesses CCE's , qualquer par de palavras válidas difere em um número razoavelmente grande de posições de bit, o que permitirá distingui-las na recepção quando ocorrerem erros de transmissão. Ao número de posições de bit que diferencia duas palavras código denomina-se "distância de Hamming" entre elas. A distância mínima de Hamming (d_{min}) de um código de bloco binário é definida como a menor distância de Hamming entre qualquer par de palavras que compõem o código. A distância mínima é um parâmetro que indica bem a capacidade de correção do código de bloco pois: o código corrige pelo menos $(d_{min} - 1)/2$ erros por bloco e deteta pelo menos $(d_{min} - 1)$ erros por bloco.
- MODULADORES DIGITAIS- Quando o canal usado é do tipo passa-faixa (como por exemplo em radio-enlaces), a sequência binária de transmissão deve ser "impressa" numa portadora. Isso é feito por técnicas de modulação: Em amplitude, fase, frequência ou combinação das mesmas.A decisão pelo esquema de modulação leva em conta compromissos como: Largura de faixa disponível, relação sinal ruído na deteção, recuperação de portadora, etc.
- CÓDIGOS DE LINHA (CL's) - São códigos destinados a casar as características do sinal de linha com requisitos impostos pelo canal/sistema de comunicações. Esses requisitos variam com a natureza do canal/sistema, por exemplo:

-Casar nulos espectrais do código com nulos espectrais do canal.
-Comprimir largura de faixa do sinal de linha.
-Recuperar relógio a partir do sinal de linha.
-Facilitar técnicas de deteção.
-Monitorar taxa de erro.
-etc.

A sequência codificada pode entrar diretamente no meio de comunicações (sistemas em banda base) ou indiretamente através de moduladores (sistema passa faixa). Exemplifiquemos alguns códigos de linha:

-AMI, HDB3, duobinário (ternários).
-Códigos binários restritos nos comprimentos máximos de 0's e 1's seguidos (RLL).
-Códigos binários balanceados.

Nessa Tese iremos nos concentrar em CL's binários balanceados embora outros CLs binários como os RLL sejam campos férteis de pesquisa. O balanceamento é equivalente a restrição dos valores assumidos pela soma digital corrida ao longo da sequência codificada. Define-se " estado da soma digital corrida " na posição j de uma sequência binária $\vec{x} = \{x_i\}$, $i \in \mathcal{N}$ - $\{0\}$, $j \in \mathcal{N}$, $x_i = \{0,1\}$, como:

$$SDC(x_j) \equiv \sum_{i=1}^{j}(2x_i - 1) + SDC(x_0)$$

Disparidade de uma palavra binária $\vec{c}$ como:

$$disp(\vec{c}) \equiv SDC(c_n) \;\; dado \;\; SDC(c_0) = 0.$$

e a variação da soma digital corrida ou desbalanço de uma sequência binária $\vec{x}$ como:

$$m = VSDC \equiv \max_j\{SDC(x_j)\} - \min_k\{SDC(x_k)\}$$

O confinamento das $SDC(x_j)$ ou equivalentemente $VSDC$ finito proporciona:

-Nulo espectral no DC [5].
-Facilita recuperação de relógio pela limitação de 0's e 1's corridos.
-Limitação na I.E.S de pico no canal de acoplamento AC [5].

Reparamos pois que esses códigos de linha são propícios para sistemas em banda base que: requeiram nulo no DC como os sistemas de comunicações óticas de deteção direta, que têm acoplamento AC ; necessitem baixo conteúdo espectral no DC para não haver interferência com o sistema de controle das cabeças de gravação, como em sistemas de gravação ótica e magnética [6]. Normalmente os CLs balanceados são códigos de bloco que podem ter um ou mais estados de SDC terminais ,ou seja, ao final de cada bloco. Quando ao final de um bloco temos apenas um estado de SDC permitido, o codificador não tem

memória e trabalhamos com um único alfabeto de codificação. Denominamos a esses códigos de monoalfabéticos. Caso tenhamos vários estados de SDC terminais , a maneira mais eficiente de trabalho é a adoção de dois alfabetos: Um progessivo de palavras com disparidade positiva ou nula e um regressivo de palavras de disparidade negativa ou nula. Desse modo, embora o codificador seja dependente de estado (estados de SDC terminais), podemos alocar convenientemente os alfabetos às mensagens, de modo a obter decodificação independente de estado. A figura 1.1 exemplifica a representação em treliça de um código monoalfabético enquanto a figura 1.2 exemplifica a treliça de um código bialfabético; ambos com VSDC=4 , n=6 e cinco estados possíveis de SDC. O diagrama em treliças é a representação temporal dos estados e suas transições. Um parâmetro importante num CL balanceado é a sua frequência de corte inferior, que mede o quão eficiente é o corte nas baixas frequências provocado pelo código. Assim definimos a frequência de corte como aquela em que a potência atinge a metade do valor de patamar(regime) dentro da curva de espectro do código.

Observamos na exposição introdutória acima que as melhorias buscadas com os trabalhos nas três áreas que compõem a codificação de canal coincidem em vários pontos: imunidade a ruído, banda ocupada, I.E.S. Assim, a procura pelo melhor desempenho levou recentemente pesquisadores a unir as áreas citadas, a saber:

-Moduladores + CCE's [7]
-Moduladores + CL's [9]
-CL's + CCE's [8]

Esse último é o objeto de estudo dessa tese, e forma a classe dos "códigos de linha com controle de erros" (CLCE's). Essa classe tenta melhorar o desempenho de sistemas que tradicionalmente usavam o cascateamento de um codificador/decodificador de linha e um codificador/decodificador de controle de erros. As principais vantagens esperadas com a substituição pelos CLCE's são:

-Redução de complexidade (1 cod/dec. em vez de 2)
-Evitar a multiplicação de erros que normalmente ocorre no decodificador de linha
-Aumentar a taxa total da codificação, onde definimos taxa de um código como $r = k/n$. No caso de concatenação temos $r_{total} = k_1/n_1 \times k_2/n_2$, onde o índice 1 representa o codificador de linha e o indice 2 o codificador de controle de erros(onde normalmente $k_1 = n_2$).

Para melhor compreensão compare as figuras 1.3 e 1.4.

1.2 Objetivos da tese

Como essa é uma classe de códigos ainda pouco estudada na literatura, objetivamos investigar as suas propriedades, métodos de construção e aplicações.

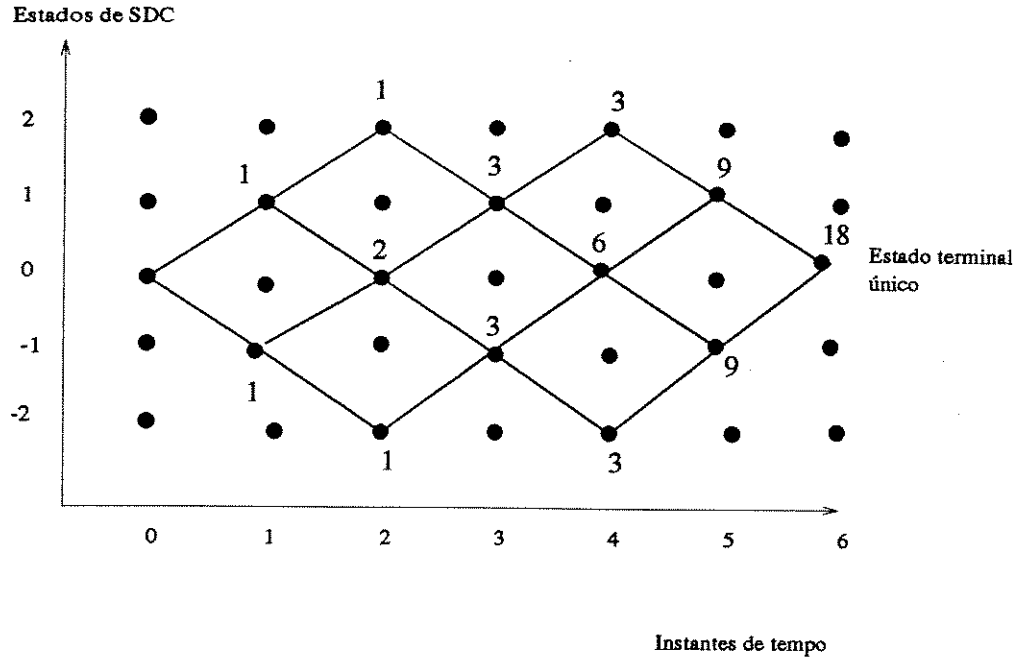


Figura 1.1: Representação em treliça de um CL balanceado monoalfabético

Começamos pela investigação de CLCEs de máximas performances pelo método de busca exaustiva computacional . Limitações nesse método nos apontaram a necessidade de elaboração de uma teoria de pisos e tetos de eficiência para grandes comprimentos de bloco (capítulo 3). Depois partimos para métodos que possibilitassem uma implementação prática dos CLCEs(capítulo 4).

Para completar o trabalho, fizemos a avaliação dos CLCE's obtidos aplicando-os a um tipo de sistema de comunicações óticas(capítulo 5). Foi elaborada uma teoria simplificada para estimação de ganhos de codificaçao nesses sistemas cujos resultados foram apurados por simulação computacional do sistema completo.

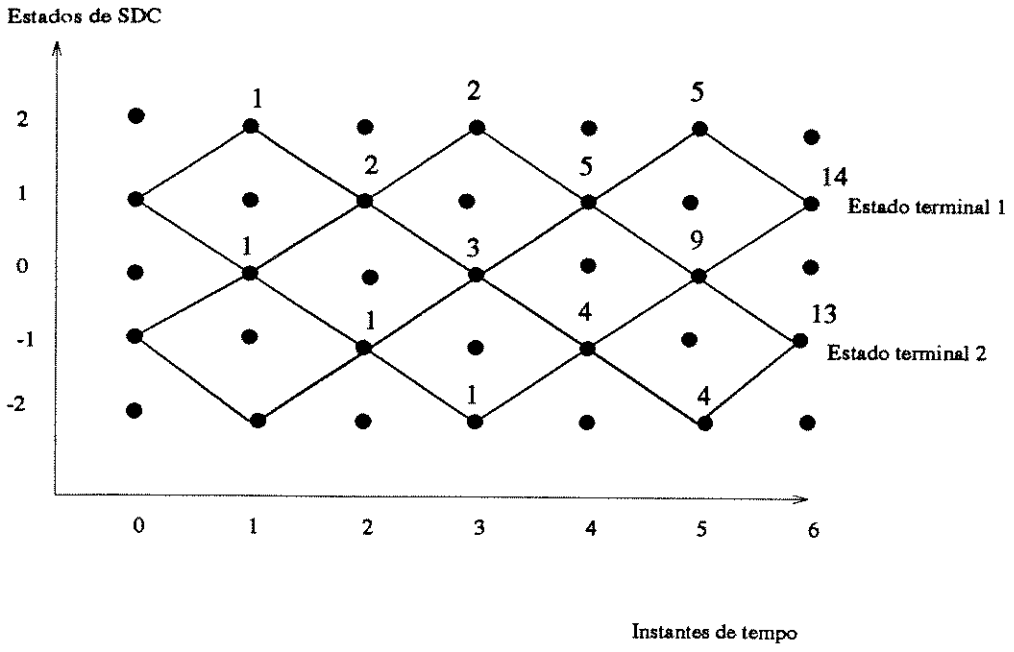


Figura 1.2: Representação em treliça de um CL balanceado bialfabético

Figura 1.3: Sistema que usa codificações de linha e de controle de erro em cascata

1.3 Nomenclatura usada ao longo da tese

- SDC ... Soma digital corrida
- $SDC(x_j)$ ... estado de SDC no instante j para a sequência $\vec{x}$. (Quando não especifica a palavra é tratada como $SDC(j)$).
- $|a|$ ... Cardinalidade do conjunto a.
- CL ... Código de linha.
- CCEL ... Código de controle de erros (linear).

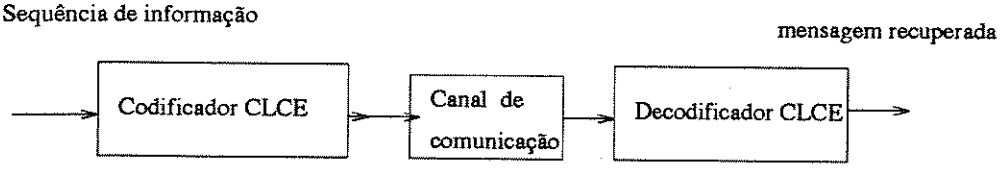


Figura 1.4: Sistema que usa codificações de linha e de controle de erro combinadas

- CLCE ... Código de linha com controle de erros.
- r ... Taxa de código.
- d_{min} ... Distância mínima de Hamming.
- W($\vec{c}$) ... Peso de Hamming da palavra binária $\vec{c}$.
- SCL ... Subcódigo de linha.
- n ... Comprimento de bloco.
- k ... Número de bits de informação no bloco codificado.
- VSDC ou m ... Variação da soma digital corrida ou desbalanço.
- $1^{(i)}0^{(j)}$ ... Formação de palavra equivalente a $\underbrace{11...11}_{i\ vezes}\underbrace{00...00}_{j\ vezes}$.
- $\vec{x}$ ... sequência binária genérica.
- $\vec{x}_i$ ... i_ésima palavra de um código binária.
- x_j ... j_ésimo dígito da palavra binária $\vec{x}$.
- X_m... conjunto das sequências binárias de desbalanço m.
- δ... distância relativa: distância de Hamming dividida pelo comprimento de bloco.

Bibliografia

[1] Haykin,S.:"Digital communications",John Wiley & Sons, Singapore, 1988.

[2] Lin,S. e Costello,D.J.:"Error control coding: Fundamentals and applications",Prentice Hall, 1983.

[3] Proakis,J.G:"Digital communications", Mc Graw-Hill, Singapure, 1989.

[4] Cattermole,K.N.: "Codes for digital line transmission",Electron.Lett,vol.11,pp.548-550,1975.

[5] Alcântara,H.J.M:"Códigos de bloco binários de faixa mínima para comunicações via fibra ótica", FEE/Unicamp, julho 1988.

[6] Siegel,P.H. & Wolf,J.K.:"Modulation and coding for information storage",IEEE Comm. Magazine, dezembro 1991.

[7] Ungerboeck,G.:"Trellis coded modulation with redundant signal sets part I: itroduction", IEEE Comm. Magazine,vol.25, fevereiro 1987.

[8] O'Reilly,J.J. & Popplewell A.:"Class of disparity reducing transmission codes with embedded error protection",IEE proceedings, vol.137, abril 1990.

[9] Anderson,J.B.,Sundberg, C.E.W.,Aulin,T.,Rydbeck,N.:"Power-bandwidth performance of smoothed phase modulation codes", IEEE Trans. Comm., vol.29,pp.187-195,1981.

Capítulo 2

Obtenção de CLCEs por busca exaustiva computacional

2.1 Introdução

Através da força bruta computacional é possível gerar conjuntos de palavras código sujeitas a restrições simultâneas de linha (VSDC) e de distância mínima de Hamming (d_{min}). Essa abordagem se tornou o nosso primeiro instrumento de investigação das propriedades de CLCEs.

Nesse capítulo descreveremos três algoritmos de busca . Dois deles geram CLCEs monoalfabéticos enquanto que o terceiro gera CLCEs bialfabéticos (vide capítulo 1 para explicações na diferença entre códigos de linha monoalfabéticos e bialfabéticos).

O primeiro algoritmo dá como resultado todos os CLCEs monoalfabéticos com maior quantidade de palavras código para determinada restrição. Chamaremos esses códigos com tamanho máximo de "ótimos". Todavia, por motivos explicados adiante, esse algoritmo é limitado a busca de códigos de comprimento de bloco pequeno. Isso fez com que na prática o segundo algoritmo (não ótimo) fosse o mais usado. O capítulo traz como resultados de interesse, tabelas comparativas e comentários sobre os CLCEs obtidos pela implementação desses algoritmos.

2.2 Algoritmo de busca completa

Descrevemos a seguir o algoritmo que gera todos os códigos monoalfabéticos que contêm o máximo número possível de palavras para os parâmetros/restrições de entrada: n, d_{min} e VSDC.

> PASSO 1 - Selecionar, a partir das 2^n palavras de n bits, aquelas que obedecem a restrição VSDC monoalfabética. Resulta, desse modo, o código somente de linha (CL) cujas palavras são armazenadas no vetor

SEMENTE[*][1] e cujo número de elementos é dado pela variável SEMF.

PASSO 2 - Gerar a árvore de palavras código representada na figura 2.1 com:

-Profundidade dada pela variável NIVEL, onde a profundidade denota o número de vezes que o algoritmo executa o passo 2.2.

-Um conjunto ordenado de nós em cada profundidade que assumem valores de palavra código.

Para gerar a árvore siga a sequência:

2.0 NIVEL $\leftarrow$ 1.

2.1 Nós do NIVEL $\leftarrow$ SEMENTE[*].

2.2 Gere novos braços a partir de cada nó da profundidade atual (NIVEL) da seguinte maneira:

–Chamaremos ao nó gerador em processamento de "nó origem atual".

–Olhe para o nível anterior e identifique o nó que gerou o "nó origem atual" e chame-o de "nó origem anterior". Considere que para NIVEL=1 todos os nós têm a mesma origem anterior.

–Calcule as distâncias de Hamming entre o "nó origem atual" e cada nó abaixo dele que tenha o "nó origem anterior" comum. Se a distância a um desses nós for maior ou igual a d_{min}, faça surgir um novo ramo apontando para um novo nó e atribua a ele valor igual ao do nó comparado. Se , ao termino dos cálculos, o "nó origem atual" não gerou ramos; chame-o de "nó não sobrevivente", caso contrário será um "nó sobrevivente".

2.3 NIVEL $\leftarrow$ NIVEL + 1.

2.4 Se houve pelo menos um nó sobrevivente, vá para 2.2

2.5 NIVEL $\leftarrow$ NIVEL - 1.

PASSO 3 - Volte na árvore através dos braços começando nos nós sobreviventes na última expansão (de profundidade NIVEL-1 para NIVEL) e acabando na profundidade NIVEL=1. Desse modo os códigos ótimos são dados pela coleção de nós ao longo de cada caminho inverso na árvore.

PASSO 4 - Fim.

Ao implementarmos o algoritmo, verificamos a existência de um mecanismo que limita a sua aplicação a CLCEs de comprimento de bloco pequeno, a saber:

[1] Ao longo dos algoritmos a notação de vetor deve ser encarada como a estrutura de armazenamento de dados no computador que matematicamente representa um conjunto ordenado. No caso, o i_ésimo elemento é dado por SEMENTE[i]

A quantidade de nós sobreviventes nos primeiros níveis (antes de começarem a morrer) cresce muito rapidamente. Essa explosão requer quantidade muito grande de memória de computador pois todos os nós devem ser memorizados enquanto sobreviventes. Na nossa implementação conseguimos trabalhar apenas com $n < 10$ para VSDC=4, o que consumiu aproximadamente 2.5 M bytes de memória. Vide a figura 2.2 para melhor visualização do problema.

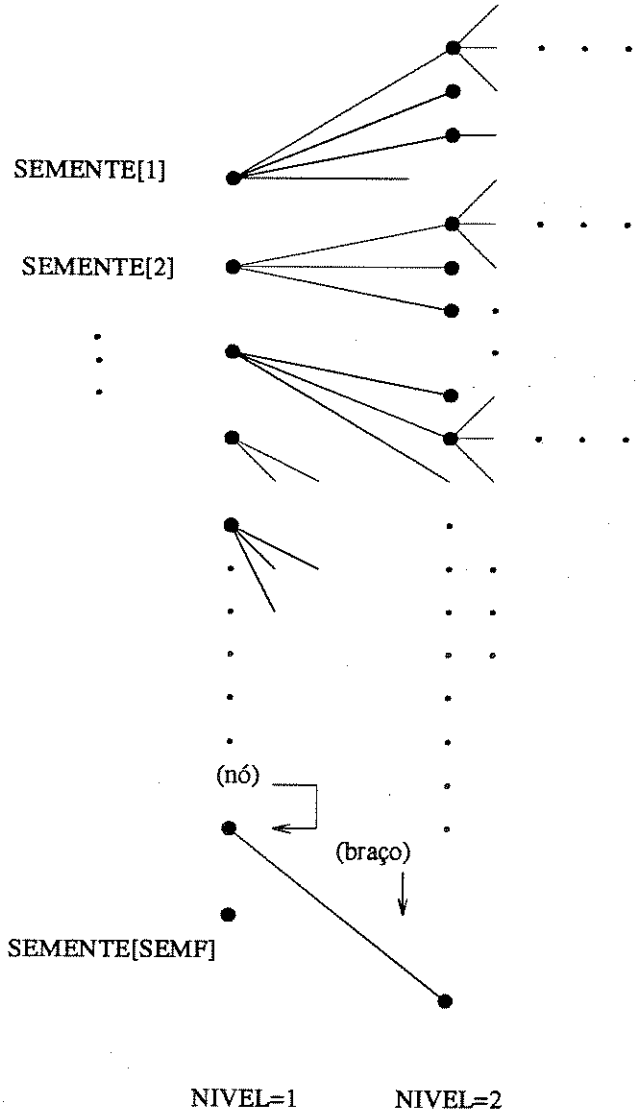


Figura 2.1: Árvore de busca completa

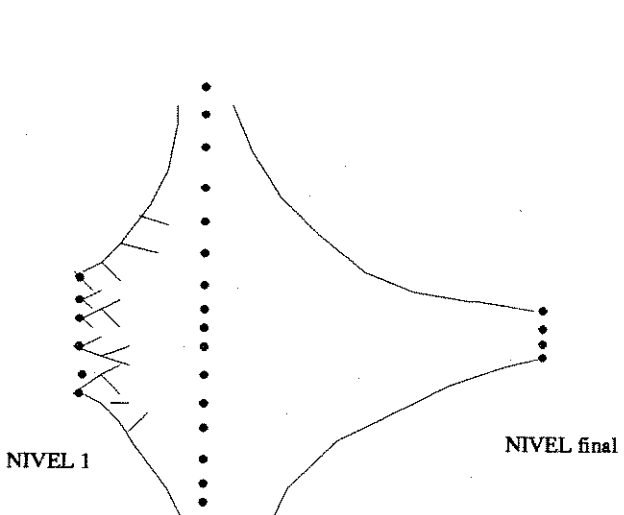


Figura 2.2: Explosão de nós no algoritmo de busca completa

O algoritmo descrito foi implementado em linguagem Pascal(VAX) e se encontra no apêndice B sob o título "Programa CODRESTR.PAS".

2.2.1 Tabela de códigos gerados pelo algoritmo de busca completa

A tabela abaixo lista (em representação hexadecimal) os 12 códigos ótimos gerados pelo algoritmo para as entradas: n=8 , VSDC=4 , d_{min} = 4. São doze palavras código por bloco, correspondendo a 0.448 bits por símbolo; na ausência de correção de erros(d_{min} = 1), a mesma restrição (VSDC=4) e o mesmo tamanho de blocos(n=8) nos dariam 54 palavras para um código monoalfabético, correspondendo a 0.719 bits por símbolo. A correção de erros impõe portanto, neste caso, um aumento de 60% na velocidade de transmissão de símbolos, a ser compensado pelo ganho de imunidade ao ruído na detecção

código1	D4	C3	B2	A5	99	8E	71	66	5A	4D	3C	2B
código2	D4	C3	B1	A6	9A	8D	72	65	59	4E	3C	2B
código3	D2	C5	B4	A3	99	8E	71	66	5C	4B	3A	2D
código4	D2	C5	B1	A6	9C	8B	74	63	59	4E	3A	2D
código5	D1	C6	B4	A3	9A	8D	72	65	5C	4B	39	2E
código6	D1	C6	B2	A5	9C	8B	74	63	5A	4D	39	2E
código7	CC	C3	AA	A5	99	96	69	66	5A	55	3C	33
código8	CC	C3	A9	A6	9A	95	6A	65	59	56	3C	33
código9	CA	C5	AC	A3	99	96	69	66	5C	53	3A	35
código10	CA	C5	A9	A6	9C	93	6C	63	59	56	3A	35
código11	C9	C6	AC	A3	9A	95	6A	65	5C	53	39	36
código12	C9	C6	AA	A5	9C	93	6C	63	5A	55	39	36

Para superar o problema de explosão de memória partimos para o algoritmo de "busca única", descrito a seguir.

2.3 Algoritmo de busca única

Esse algoritmo nada mais faz do que computar somente o caminho formado pelos braços superiores dentro da árvore descrita anteriormente .
A saber:

PASSO 1 - Idem ao passo 1 do algoritmo de busca completa; chamaremos SEMF[*] calculado nesse passo de "conjunto inicial de busca" .

PASSO 2 - O conjunto Codigo[*] está vazio e NIVEL $\leftarrow$ 0.

PASSO 3 - Tomar o primeiro elemento de SEMENTE[*] como novo integrante do conjunto CODIGO[*] e NIVEL ← NIVEL+1.

PASSO 4 - Percorrer o conjunto SEMENTE[*] expurgando do mesmo todas as palavras com distância de Hamming menor que d_{min} em relação à palavra escolhida no passo 3 .

PASSO 5 - Caso o número de elementos de SEMENTE[*] seja maior que zero volte ao passo 3.

PASSO 6 - Fim. O conjunto CODIGO[*] contém o CLCE gerado pelo algoritmo e a quantidade de elementos do código é dada pela variável NIVEL, como no primeiro algoritmo.

2.3.1 Algumas observações sobre o algoritmo

- O número de bits de informação do CLCE é dado por $k = int[log_2(NIVEL)]$ onde $int[.]$ denota a parte inteira de.
- O código gerado não é necessariamente ótimo pois adotamos, a priori, um dos possíveis caminhos ao longo da árvore. Contudo, pudemos constatar empiricamente que k e $k_{ótimo}$ (correspondente ao código ótimo) estão muito próximos, o que nos leva a crer que esse algoritmo gera limitantes inferiores (pisos) apertados para o tamanho máximo dos CLCEs monoalfabéticos.
- Nota-se que para um mesmo conjunto de parâmetros de entrada, podemos tentar aumentar o tamanho do CLCE gerado, variando as ordenações do conjunto inicial de busca. É um procedimento empírico pois não sabemos qual ordenação levará ao código ótimo. Só sabemos que ela existe.
- O limitante desse algoritmo é a explosão no tempo de computação com a quantidade de elementos do conjunto inicial de busca devido ao aumento " exponencial" do número de comparações de distância requeridas.

O algoritmo implementado em linguagem Pascal(VAX) encontra-se listado no apêndice B sob o título de "Programa CODRESUNICO.PAS".

2.3.2 Tabelas de códigos gerados pelo algoritmo de busca única

As tabelas a seguir mostram os resultados da aplicação desse algoritmo para "conjuntos iniciais de busca" com quantidade de elementos menor que 400.000. Nelas colocamos as quantidades de palavras código do CLCE (NIVEL) e o número de bits de informação na notação (n,k). Com fins de comparação acrescentamos as colunas: CL com os parâmetros (n,k) do código de linha monoalfabético com restrição VSDC e CCEL com os parâmetros (n,k) do melhor código de controle de erros linear[2] com

[2] Códigos extraídos da referência [1]

Tabela 2.1: CLCEs monoalfabéticos com $d_{min} = 4$ e $VSDC = 2$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
4	2	(4,1)	4	(4,2)	(4,1)
6	2	(6,1)	8	(6,3)	(6,2)
8	8	(8,3)	16	(8,4)	(8,4)
10	8	(10,3)	32	(10,5)	(10,5)
12	32	(12,5)	64	(12,6)	(12,7)
14	32	(14,5)	128	(14,7)	(14,9)
16	128	(16,7)	256	(16,8)	(16,11)
18	128	(18,7)	512	(18,9)	(18,12)
20	512	(20,9)	1024	(20,10)	(20,14)
22	512	(22,9)	2048	(22,11)	(22,16)

Tabela 2.2: CLCEs monoalfabéticos com $d_{min} = 4$ e $VSDC = 4$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
4	2	(4,1)	6	(4,2)	(4,1)
6	3	(6,1)	18	(6,4)	(6,2)
8	9	(8,3)	54	(8,5)	(8,4)
10	21	(10,4)	162	(10,7)	(10,5)
12	47	(12,5)	486	(12,8)	(12,7)
14	119	(14,6)	1458	(14,10)	(14,9)
16	345	(16,8)	4374	(16,12)	(16,11)
18	1009	(18,9)	13122	(18,13)	(18,12)
20	2877	(20,11)	39366	(20,15)	(20,14)
22	8283	(22,13)	118098	(22,16)	(22,16)

n e d_{min} em questão. Nota-se que a coluna SEMF gerada pelo algoritmo dá a quantidade de palavras do CL.

Tabela 2.3: CLCEs monoalfabéticos com $d_{min} = 4$ e $VSDC = 6$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
4	2	(4,1)	6	(4,2)	(4,1)
6	4	(6,2)	20	(6,4)	(6,2)
8	14	(8,3)	68	(8,6)	(8,4)
10	26	(10,4)	232	(10,7)	(10,5)
12	92	(12,6)	792	(12,9)	(12,7)
14	222	(14,7)	2704	(14,11)	(14,9)
16	663	(16,9)	9232	(16,13)	(16,11)
18	1871	(18,10)	31520	(18,14)	(12,18)
20	5550	(20,12)	107616	(20,16)	(20,14)
22	17087	(22,14)	367424	(22,18)	(22,16)

Tabela 2.4: CLCEs monoalfabéticos com $d_{min} = 4$ e $VSDC = 8$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
4	2	(4,1)	6	(4,2)	(4,1)
6	4	(6,2)	20	(6,4)	(6,2)
8	14	(8,3)	70	(8,6)	(8,4)
10	25	(10,4)	250	(10,7)	(10,5)
12	84	(12,6)	900	(12,9)	(12,7)
14	242	(14,7)	3250	(14,11)	(14,9)
16	733	(16,9)	11750	(16,13)	(16,11)
18	2204	(18,11)	42500	(18,15)	(18,12)
20	7034	(20,12)	153750	(20,17)	(20,14)

Tabela 2.5: CLCEs monoalfabéticos com $d_{min} = 4$ e $VSDC = 10$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
4	2	(4,1)	6	(4,2)	(4,1)
6	4	(6,2)	20	(6,4)	(6,2)
8	14	(8,3)	70	(8,6)	(8,4)
10	22	(10,4)	252	(10,7)	(10,5)
12	84	(12,6)	922	(12,9)	(12,7)
14	248	(14,7)	3404	(14,9)	(14,9)
16	863	(16,9)	12630	(16,13)	(16,11)
18	2439	(18,11)	46988	(18,15)	(18,12)
20	7924	(20,12)	175066	(20,17)	(20,14)

Tabela 2.6: CLCEs monoalfabéticos com $d_{min} = 4$ e $VSDC = 12$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
4	2	(4,1)	6	(4,2)	(4,1)
6	4	(6,2)	20	(6,4)	(6,2)
8	14	(8,3)	70	(8,6)	(8,4)
10	22	(10,4)	252	(10,7)	(10,5)
12	68	(12,6)	924	(12,9)	(12,7)
14	247	(14,7)	3430	(14,11)	(14,9)
16	799	(16,9)	12838	(16,13)	(16,11)
18	2492	(18,11)	48314	(18,15)	(18,12)
20	8140	(20,12)	182476	(20,17)	(20,14)

Tabela 2.7: CLCEs monoalfabéticos com $d_{min} = 6$ e $VSDC = 4$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
4	1		6	(4,2)	
6	2	(6,1)	18	(6,4)	(6,1)
8	2	(8,1)	54	(8,5)	(8,1)
10	5	(10,2)	162	(10,7)	(10,2)
12	10	(12,3)	486	(12,8)	(12,4)
14	22	(14,4)	1458	(14,10)	(14,5)
16	49	(16,5)	10925	(16,12)	(16,7)
18	115	(18,6)	13122	(18,13)	(18,9)
20	283	(20,8)	30366	(20,15)	(20,10)
22	702	(22,9)	118098	(22,16)	(22,12)
24	1805	(24,10)	354294	(24,18)	(24,14)

Tabela 2.8: CLCEs monoalfabéticos com $d_{min} = 6$ e $VSDC = 6$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
8	2	(8,1)	68	(8,6)	(8,1)
10	6	(10,2)	232	(10,7)	(10,2)
12	20	(12,4)	792	(12,9)	(12,4)
14	38	(14,5)	2704	(14,11)	(14,5)
16	69	(16,6)	9232	(16,13)	(16,7)
18	176	(18,7)	31520	(18,14)	(18,9)
20	461	(20,8)	107616	(20,16)	(20,10)
22	1214	(22,10)	367424	(22,18)	(22,12)
24	3314	(24,11)	1254464	(24,20)	(24,14)

Tabela 2.9: CLCEs monoalfabéticos com $d_{min} = 6$ e $VSDC = 8$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
6	2	(6,1)	20	(6,4)	(8,1)
8	2	(8,1)	70	(8,6)	(10,2)
10	6	(10,2)	250	(10,7)	(12,4)
12	14	(12,3)	900	(12,9)	(14,5)
14	30	(14,4)	3240	(14,11)	(16,7)
16	77	(16,6)	11750	(16,13)	(18,9)
18	202	(18,7)	42500	(18,15)	(20,10)
20	535	(23,9)	153750	(20,17)	(22,12)
22	1493	(22,10)	556250	(22,19)	(24,14)

Tabela 2.10: CLCEs monoalfabéticos com $d_{min} = 6$ e $VSDC = 10$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
6	2	(6,1)	20	(6,4)	(6,1)
8	2	(8,1)	70	(8,6)	(8,1)
10	6	(10,2)	252	(10,7)	(10,2)
12	10	(12,3)	922	(12,9)	(12,4)
14	27	(14,4)	3404	(14,11)	(14,5)
16	77	(16,6)	12630	(16,13)	(16,7)
18	207	(18,7)	46988	(18,15)	(18,9)
20	560	(20,9)	175066	(20,17)	(20,10)
22	1588	(22,10)	652764	(22,19)	(22,12)

Tabela 2.11: CLCEs monoalfabéticos com $d_{min} = 8$ e $VSDC = 4$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
8	2	(8,1)	54	(8,5)	(8,1)
10	2	(10,1)	162	(10,7)	(10,1)
12	3	(12,1)	486	(12,8)	(12,2)
14	5	(14,2)	1458	(14,10)	(14,3)
16	15	(16,3)	4374	(16,12)	(16,5)
18	22	(18,4)	13122	(18,13)	(18,6)
20	49	(20,5)	39366	(20,15)	(20,8)
22	104	(22,6)	118098	(22,16)	(22,10)
24	246	(24,7)	354294	(24,18)	(24,12)
26	553	(26,9)	1062882	(26,20)	(26,12)

Tabela 2.12: CLCEs monoalfabéticos com $d_{min} = 8$ e $VSDC = 6$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
8	2	(8,1)	68	(8,6)	(8,1)
10	2	(10,1)	232	(10,7)	(10,1)
12	3	(12,1)	792	(12,9)	(12,2)
14	7	(14,2)	2704	(14,11)	(14,3)
16	23	(16,4)	9232	(16,13)	(16,5)
18	30	(18,4)	31520	(18,14)	(18,6)
20	67	(20,6)	107616	(20,16)	(20,8)
22	158	(22,7)	367424	(22,18)	(22,10)
24	370	(24,8)	1254464	(24,20)	(24,12)

Tabela 2.13: CLCEs monoalfabéticos com $d_{min} = 8$ e $VSDC = 8$;

n	NÍVEL	CLCE	SEMF	CL	CCE linear
8	2	(8,1)	70	(8,6)	(8,1)
10	2	(10,1)	250	(10,7)	(10,1)
12	3	(12,1)	900	(12,9)	(12,2)
14	8	(14,3)	3250	(14,11)	(14,3)
16	30	(16,4)	11750	(16,13)	(16,5)
18	33	(18,5)	42500	(18,15)	(18,6)
20	72	(20,6)	153750	(20,17)	(20,8)
22	179	(22,7)	556250	(22,19)	(22,10)

Tabela 2.14: CLCEs monoalfabéticos com $d_{min} = 8$ e $VSDC = 10$;

n	NÍVEL	CLCE	SEMF	CL	CCEL
8	2	(8,1)	70	(8,6)	(8,1)
10	2	(10,1)	252	(10,7)	(10,1)
12	4	(12,4)	922	(12,9)	(12,2)
14	8	(14,3)	3404	(14,11)	(14,3)
16	30	(16,4)	12630	(16,13)	(16,5)
18	33	(18,5)	46988	(18,15)	(18,6)
20	74	(20,6)	175066	(20,17)	(20,8)
22	184	(22,4)	652764	(22,19)	(22,10)

2.4 Algoritmo de busca única bialfabética

2.4.1 introdução aos códigos bialfabéticos

O emprego de dois alfabetos - progressivo e regressivo - é uma técnica comum para códigos de linha restritos em VSDC [2] e permite aumentar a taxa do CL. O alfabeto progessivo engloba palavras de disparidade maior ou igual a zero enquanto que o regressivo contém palavras de disparidade menor ou igual a zero. A decisão de transmissão de mensagem via alfabeto progressivo ou regressivo é tomada pelo codificador baseado no valor do estado de SDC ao final do bloco transmitido anteriormente de modo a mante-la confinada. Dessa forma, embora trabalhemos com vários estados de SDC terminais aos blocos (ao contrário do caso monoalfabético com um só estado), conseguimos aumentar a quantidade de palavras por alfabeto em relação ao monoalfabético e consequentemente atingimos maiores taxas de CL (dado um comprimento de bloco n). O algoritmo adota como ponto de partida códigos bialfabéticos de linha cuja estrutura esta descrita detalhadamente em [2]. Trata-se de CLs mB-nB cujos alfabetos são simétricos ,ou seja, toda palavra código $\vec{x}$ tem uma correspondente complementar $\bar{\vec{x}}$. Com essa exigência garantimos probabilidades P(0)=P(1)=0.5 em todas as janelas temporais do CL evitando o aparecimento de raias espectrais em frequências múltiplas da taxa de bloco desde que as palavras fonte sejam equiprováveis[3]. Os parâmetros que caracterizam esses CLs quanto a VSDC e número de estados terminais de SDC são:

D - Extensão para cima e para baixo do intervalo de estados de SDC terminais.

$disp_{max}$ - máxima disparidade dentre as palavras código.

Esses parâmetros impõem para toda palavra $\vec{x}$ do alfabeto regressivo, $\alpha 1$, as seguintes restrições:

Dada $SDC(x_0) = 0$ e $\vec{x} \in \alpha 1 \Rightarrow$
$-D - disp_{max} \leq SDC(x_i) \leq D$ para $0 < i < n-1$;
$-disp_{max} \leq SDC(x_n) \leq 0$.

A lei que rege a adoção de um ou outro alfabeto tem por objetivo confinar os estados de SDC e estabelece que : (Convencionando o mínimo valor atingido pela SDC como zero) Quando a SDC terminal vale $disp_{max}$ ou mais aplica-se o alfabeto regressivo, caso contrário aplica-se o alfabeto progressivo. Resulta assim a relação entre os parâmetros citados e VSDC :

VSDC= $2(disp_{max} + D) - 1$; $disp_{max}$ impar.
VSDC= $2(disp_{max} + D) - 2$; $disp_{max}$ par > 0.
VSDC= $2D$; $disp_{max} = 0$.

2.4.2 O algoritmo

O algoritmo de "busca única bialfabética" trabalha com os alfabetos regressivo($\alpha 1$) e progressivo($\alpha 2$) do CL descrito acima e aplica sobre a união desses alfabetos ($\alpha 3$) o algoritmo de "busca única" para obter o conjunto CODIGO3[*] com restrição de distância mínima de Hamming. Se levarmos em conta que a propriedade de decodificação independente de estado[3] deve estar presente nos CLCEs gerados pelo algoritmo pois problemas de propagação de erros não seriam tolerados, exige-se que palavras comuns aos dois alfabetos correspondam ao mesmo bloco de informação. Assim, obteremos cada alfabeto do CLCE como a intersecção de CODIGO3[*] com o correspondente alfabeto do CL: CODIGO1[*]= CODIGO3[*]$\bigcap \alpha 1$; CODIGO2[*]= CODIGO3[*]$\bigcap \alpha 2$. Vemos claramente que a quantidade de palavras código que efetivamente carregam informação é dada pelo tamanho do menor entre os alfabetos CODIGO1[*] e CODIGO2[*], portanto para otimizar a distribuição das intersecções entre os dois alfabetos CLCE temos que ordenar adequadamente o conjunto união $\alpha 3$ antes da aplicação do algoritmo de "busca única".

O algoritmo de "busca única bialfabética" está descrito a seguir:

PASSO 1 - Gerar os alfabetos restritos em D e $disp_{max}$: $\alpha 1$ e $\alpha 2$.

PASSO 2 - Gerar $\alpha 3 = \alpha 1 \bigcup \alpha 2$ ordenado segundo os critérios:

Primeiro- Palavras balanceadas cujas complementares estejam no mesmo alfabeto e apresentem VSDC=2.

Segundo- Restante das balanceadas cujas complementares estejam no mesmo alfabeto.

Terceiro- Palavras balanceadas não comuns aos dois alfabetos com alfabetos de origem alternados (pares complementares).

Quarto- Palavras não balanceadas com alfabeto de origem alternados(pares complementares).

PASSO 3 - Achar CODIGO3[*] aplicando o algoritmo de "busca única" sobre o conjunto inicial $\alpha 3$ com a restrição d_{min} requerida.

PASSO 4 - Achar os alfabetos CLCE através das intercecções CODIGO1[*]= $\alpha 1 \bigcap$ CODIGO3[*] e CODIGO2[*]= $\alpha 2 \bigcap$ CODIGO3[*].

PASSO 5 - Fim.

O algoritmo de busca "bialfabética única" foi implementado em linguagem PASCAL(SUN) e apresenta-se listado no apêndice B sob o título "Programa COALF.PAS". A seguir mostramos algumas tabelas com resultados da aplicação desse algoritmo.

[3] Esses estados referem-se as SDCs terminais do código bialfabético

2.5 Tabelas de CLCEs bialfabéticos

As tabelas 2.15 2.16 2.17 expõem alguns códigos gerados pelo algoritmo. Apresentamos resultados em forma similar às tabelas de CLCEs monoalfabéticos, acrescidas as colunas :

NIVEL1 - número de palavras no alfabeto regressivo do CLCE.

NIVEL2 - número de palavras no alfabeto progressivo do CLCE.

NIVEL3 - número de palavras do conjunto CODIGO3[*]

SEMF1 - número de palavras do conjunto união $\alpha 3$.

Tabela 2.15: CLCEs bialfabéticos com $d_{min} = 3$, D=1 e $disp_{max} = 1$ (VSDC=3)

n	NIVEL1	NIVEL2	NIVEL3	CLCE	SEMF1	CL	CCEL
8	10	10	12	(8,3)	55	(8,5)	(8,4)
10	20	20	32	(10,4)	144	(10,7)	(10,6)
12	57	57	82	(12,5)	377	(12,8)	(12,8)
14	120	120	208	(14,7)	987	(14,9)	(14,10)
16	349	349	570	(16,8)	2584	(16,11)	(16,11)

Tabela 2.16: CLCEs bialfabéticos com $d_{min} = 3$, D=1 e $disp_{max} = 2$ (VSDC=4)

n	NIVEL1	NIVEL2	NIVEL3	CLCE	SEMF1	CL	CCEL
8	10	10	12	(8,3)	81	(8,6)	(8,4)
10	22	22	36	(10,4)	243	(10,7)	(10,6)
12	62	62	92	(12,5)	729	(12,9)	(12,8)
14	151	151	270	(14,7)	2187	(14,11)	(14,10)
16	431	431	734	(16,8)	6561	(16,12)	(16,11)

Tabela 2.17: CLCEs bialfabéticos com $d_{min} = 3$, D=2 e $disp_{max} = 2$ (VSDC=6)

n	NIVEL1	NIVEL2	NIVEL3	CLCE	SEMF1	CL	CCEL
8	13	13	14	(8,3)	136	(8,6)	(8,4)
10	26	26	36	(10,4)	470	(10,8)	(10,6)
12	95	95	136	(12,6)	1616	(12,10)	(12,8)
14	251	251	388	(14,7)	5540	(14,11)	(14,10)

Tabela 2.18: CLCEs bialfabéticos com $d_{min} = 4$, D=2 e $disp_{max} = 2$(VSDC=6)

n	NIVEL1	NIVEL2	NIVEL3	CLCE	SEMF1	CL	CCEL
8	13	13	14	(8,3)	136	(8,6)	(8,4)
10	18	18	20	(10,4)	470	(10,8)	(10,5)
12	71	71	84	(12,6)	1616	(12,10)	(12,7)
14	162	162	206	(14,7)	5540	(14,11)	(14,9)

2.6 Comentários sobre os resultados obtidos ao longo do capítulo

1. Ao acompanharmos a evolução de k com n nas tabelas de códigos monoalfabéticos com d_{min}=4 2.2,2.3 e 2.4; a sequência: (k=9, VSDC=4) (k=10, VSDC=6) (k=11, VSDC=8) mostra que à medida que a restrição de linha é afrouxada, k tende a um valor limitado pela restrição de distância dada pela coluna CCEL . Por outro lado o domínio da restrição de linha pode ser observado na tabela 2.1 com VSDC=2, $d_{min} = 4$.

2. Ao tentarmos escolher o melhor código dentro de cada tabela é necessário levarmos em conta dois fatores antagônicos:

   A - Aumento de faixa e perda de energia por diminuição de taxa.

   B - Complexidade de implementação.

   O fator A melhora com o aumento do tamanho de bloco (n) e é dado por k/n. O fator B piora com o aumento do tamanho de bloco (para códigos implementados por tabela). A atribuição de valores a B é subjetiva e depende principalmente da quantidade 2^k de palavras do CLCE que varia tanto com as restrições d_{min} quanto VSDC. Com fins de montar um mapa comparativo simplificado de códigos úteis atribuimos a B[4] o valor $n^{1/2}$. Assim podemos escolher em cada tabela de CLCE monoalfabético o código que maximiza $k/n^{1.5}$, chamando-o de código "útil".

3. Definamos o "Excesso de velocidade" de um código de linha como $(r^{-1} - 1)\times 100\%$. Esse parâmetro nos dá noção do aumento da velocidade (ou faixa) de transmissão do sistema codificado em relação ao não codificado.

   O mapa tridimensional $d_{min}\times$ VSDC $\times$ "Excesso de velocidade" para alguns CLCEs monoalfabéticos úteis está representado na figura 2.3. Os limites teóricos de excesso de velocidade para $n \rightarrow \infty$($[c^{-1} - 1]\times$ 100% , onde c= capacidade de Shannon para o canal discreto sem ruído limitado em VSDC) aparecem no mapa como uma parede pois na condição $n \rightarrow \infty$ a restrição d_{min}=constante não contribui mais para o excesso de velocidade. Entretanto , para os valores de n tabelados , a restrição d_{min} comanda o excesso de velocidade com excessão do caso da forte restrição de linha VSDC=2 .

4. A condição P(0)=P(1)=0.5 não é garantida em nenhum dos algoritmos dados pois implicaria numa redução ainda maior de taxa de CLCE. Logo, os códigos gerados devem apresentar espectro com raias em frequências multiplas da taxa de bloco.

5. Comparando-se as tabelas de códigos bialfabéticos e monoalfabéticos com capacidade de correção de um erro podemos constatar que:

[4] Esse critério é subjetivo

A - Os bialfabéticos com $d_{min} = 3$ superam os monoalfabéticos com $d_{min} = 4$.

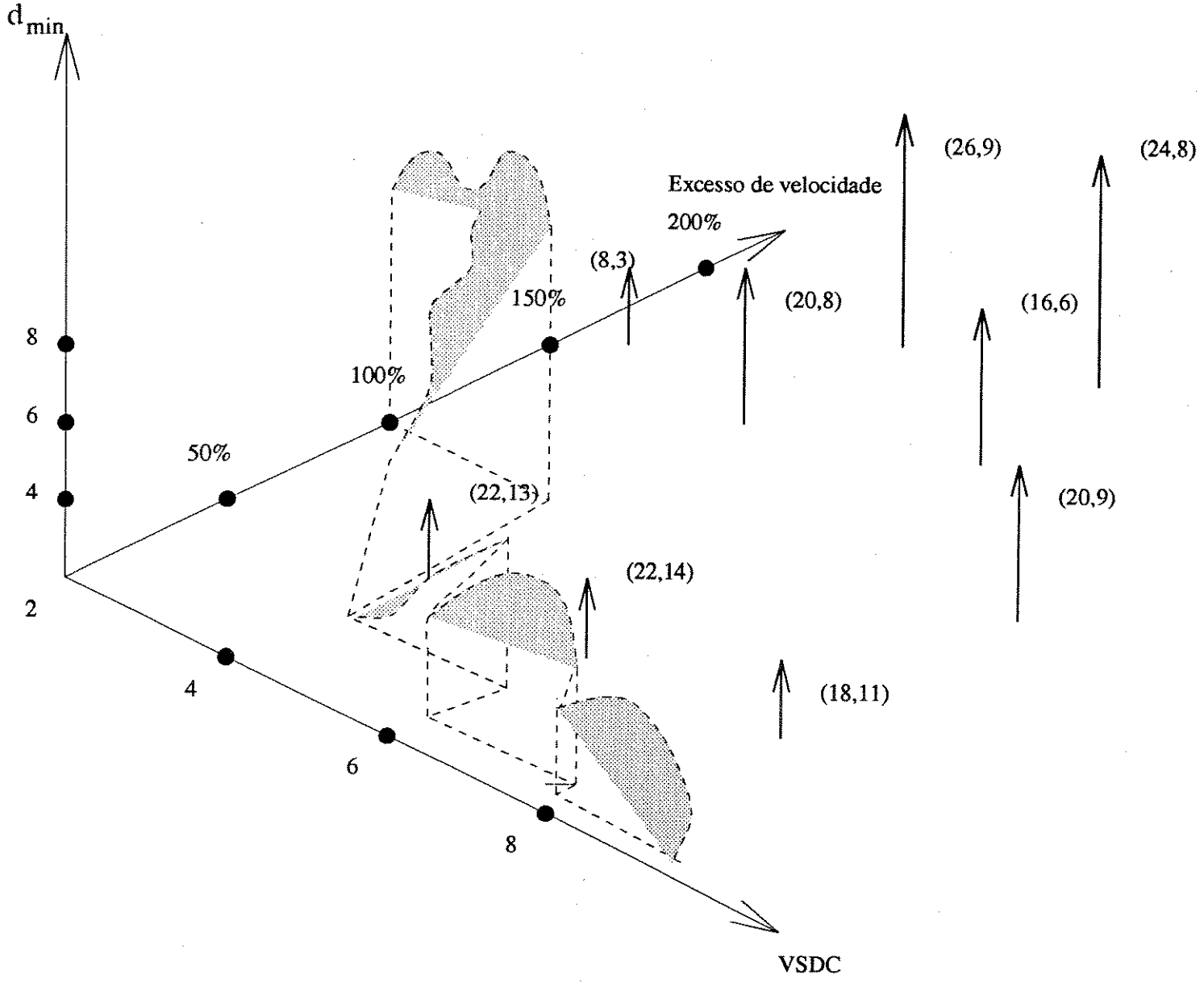
B - Esse quadro se inverte para os bialfabéticos com d_{min}=4. Nesse caso era esperado que o tamanho do código união (NIVEL3) superasse sempre o tamanho do monoalfabético (NIVEL) pois SEMENTE[*]$\subset \alpha 3$, porém isso não ocorre.

Explica-se essas observações por:

A - Os CLs bialfabéticos têm disponibilidade de palavras com distâncias ímpares e pares enquanto os monoalfabéticos somente distâncias pares, logo os primeiros são favorecidos para distância 3.

B - Reafirmação da dependência do algoritmo de busca com a ordenação inicial dos "conjuntos de busca" que no caso são bem diferentes. Além disso podemos conjecturar que o conjunto de busca monoalfabético engloba a maioria das palavras que distam 4 entre si o que torna a busca mais direcionada.

Figura 2.3: Mapa de códigos monoalfabéticos úteis



Bibliografia

[1] Verhoeff,T.:"An update table of minimum-distance bounds for binary linear codes",IEEE Trans. on Inf. Theory, vol.IT-33,setembro 1987.

[2] Romano,J.M.T. & Waldman,H:"Códigos balanceados para transmissão digital por fibra ótica",Tese de mestrado FEE-Unicamp.

[3] Benett,W.R.:"Statistics of regenerative digital transmission",BSTJ,vol.37,pp.1501-1542,1958.

Capítulo 3

Limitantes na taxa $\times$ d_{min} para CLCEs com restrição DC

3.1 Introdução

No capítulo anterior utilizamos a busca exaustiva computacional para achar códigos de linha com controle de erros, porém o crescimento rápido do tempo de computação com o aumento do comprimento de bloco (n), sugeriu uma aproximação analítica para o problema: determinação de limitantes na eficiência de CLCEs para grandes comprimentos de bloco, ou seja, limitantes assintóticos.

Argumentos de exaustão levam a um limitante inferior ou piso [2] e argumentos de empacotamento esférico levam a um limitante superior ou teto. Dado o conjunto X_m das sequências binárias de comprimento n que satisfazem à restrição DC com VSDC=m, procede-se exaustivamente na busca de um código de distância mínima d_{min} da mesma forma que no algorítmos de busca única do capítulo 2. Escolhe-se inicialmente uma sequência $\vec{x}_1 \in X_m$ e exclui-se de X_m todas as sequências com distancia de Hamming $d < d_{min}$ em relação a $\vec{x}_1$. Ao conjunto de sequências excluidas nesse procedimento denominamos T_1 e ao conjunto de busca resultante denominamos $X'_m = X_m - T_1$. Em seguida escolhe-se um ponto $\vec{x}_2 \in X'_m$ e exclui-se de X'_m todos os pontos com distância de Hamming $d < d_{min}$ em relação a $\vec{x}_2$. Ao conjunto de sequências excluidas nesse procedimento denominamos T_2. Prosseguimos o algorítmos até obter a M_ésima palavra do código, quando o conjunto de busca é esgotado. Agora definamos a esfera de Hamming no espaço X_m de raio r e centro $\vec{x}$ como o conjunto de sequências em X_m que distam r ou menos do centro $\vec{x}$. Ao número de sequências nessa esfera chamaremos de tamanho da esfera de centro $\vec{x}$ e raio r : $V_m(\vec{x}, r)$. Nota-se que a cada passo da busca exaustiva o conjunto T_i pode ser majorado pela esfera de raio $d_{min} - 1$ cujo centro escolhe-se devidamente para maximizar o tamanho da esfera de Hamming com esse raio no espaço restrito. Aqui cabe a observação que no espaço irrestrito o tamanho de esferas é igual para qualquer centro escolhido enquanto que no espaço restrito esse tamanho pode variar

com o centro. Assim, podemos escrever:

$$\sum_{i=1}^{M} |T_i| = |X_m| \tag{3.1}$$

$$|T_i| \leq \max_{\vec{x} \in X_m} \{V_m(\vec{x}, d_{min} - 1)\} \tag{3.2}$$

de (3.1) e (3.2) resulta o piso

$$M \geq \frac{|X_m|}{\max_{\vec{x} \in X_m} \{V_m(\vec{x}, d_{min} - 1)\}} \tag{3.3}$$

No caso assintótico irrestrito esse argumento leva ao conhecido piso de Gilbert-Varshamov [3, 4].

O argumento de empacotamento esférico ou de Hamming diz que se existir um código com d_{min} ímpar num espaço de Hamming cujas esferas centradas nas palavras código com raio $(d_{min} - 1)/2$ todas quase se tangenciam , então esse seria o maior código possível(vide figura 3.1). Seja M a quantidade possível de palavras para um código irrestrito com comprimento de bloco n e distância mínima d_{min}, então:

$$M \leq \frac{|X_m|}{V_m(\vec{x}, (d_{min} - 1)/2)} \tag{3.4}$$

Porém para manter o argumento no espaço restrito temos que limitar inferiormente

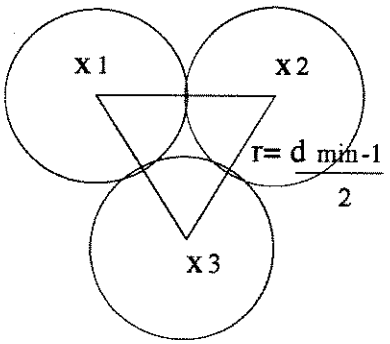


Figura 3.1: Esquema de empacotamento esférico em 2 dimensões

o denominador em (3.4) e teríamos o piso de Hamming para o espaço m balanceado:

$$M \leq \frac{|X_m|}{\min_{\vec{x} \in X_m} \{V_m(\vec{x}, (d_{min} - 1)/2)\}} \tag{3.5}$$

Para derivar esses limitantes torna-se necessário identificar as sequências centro que maximizam e minimizam as esferas no espaço X_m e calcular o tamanho dessas esferas. Assim, esse capítulo se aprofunda nessa discussão e através dela chega-se a algumas propriedades morfológicas do espaço DC-restrito assintótico.

3.2 Definições e conceitos relevantes

Por convenção SDC(0)=SDC(n)=0 e SDC(i) pode tomar valores $(-m/2, 1-m/2, ..., m/2)$ para m par; e $-(m-1)/2, 1-(m-1)/2, ..., (m+1)/2$ para m ímpar.

O número de sequências em X_m foi estudada anteriormente em [1] e é assintoticamente dada por $[\lambda_{max}(m)]^{n+\Omega(n)}$ para n grande, onde $\lambda_{max}(m)$ é o maior autovalor da matriz de adjacência da restrição m-DC. Essa matriz $(m+1) \times (m+1)$ é uma matriz de transição cujos elementos (i, j) representam o número de caminhos de SDC que podem ser feitos por um único dígito indo do estado i para o estado j , onde o estado k é definido por: $[SDC_{max} - (k-1)] = SDC$ no estado k, $k \in \{1, 2, ..., m, m+1\}$, $SDC_{max} = m/2$ para m par e $SDC_{max} = (m+1)/2$ para m ímpar, assim:

$$A_m = \begin{vmatrix} 0 & 1 & & & & & . & . & 0 \\ 1 & 0 & 1 & & & & & & . \\ & 1 & 0 & 1 & & & & & . \\ & & & & . & . & & & \\ . & & & & & . & . & & \\ . & & & & & & 1 & 0 & 1 \\ 0 & . & . & & & & & 1 & 0 \end{vmatrix}$$

A treliça da figura 3.2 ilustra o significado da matriz de transição para m=6, que é

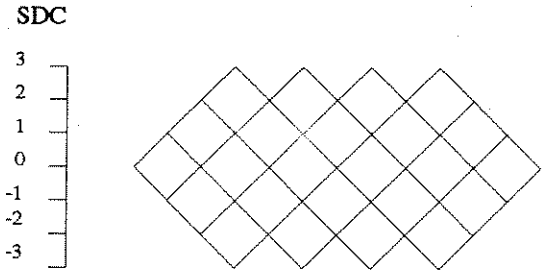


Figura 3.2: Treliça de palavra restrita com m=6

valida para a parte da treliça em regime. $\Omega(n)$ é uma sequência de números tal que $\Omega(n)/n$ tende a zero quando n vai para infinito. Ao número de sequências que estão a uma dada distância de Hamming d de uma sequência centro $\vec{v} \in X_m$ denominamos tamanho da superfície esférica de Hamming de raio d e centro $\vec{v}$: $S_m(\vec{v}, d)$. A soma cumulativa dos tamanhos das superfícies esféricas de Hamming de raios d de zero até r e centro $\vec{v}$ leva ao tamanho da esfera de Hamming de centro $\vec{v}$ e raio r:

$$V_m(\vec{v}, r) = \sum_{d=0}^{r} S_m(\vec{v}, d) \tag{3.6}$$

Vamos considerar as taxas assintóticas para tamanhos de esferas e superfícies esféricas:

$$\dot{V}_m(\vec{x}, \delta) = \lim_{n \to \infty} \frac{1}{n} \log_2 V_m(\vec{x}, \delta n) \tag{3.7}$$

e

$$\dot{S}_m(\vec{x},\delta) = \lim_{n\to\infty} \frac{1}{n} \log_2 S_m(\vec{x},\delta n) \tag{3.8}$$

e δ é chamado de distância relativa de Hamming (distância dividida pelo comprimento de bloco). O seguinte lema introduz duas propriedades úteis dessas duas taxas para qualquer centro $\vec{x}$:

Lema 1

$$\dot{V}_m(\vec{x},\delta) = \max_{0\le\epsilon\le\delta}\{\dot{S}_m(\vec{x},\epsilon)\} \tag{3.9}$$

$$\max_{0\le\delta\le 1}\{\dot{S}_m(\vec{x},\delta)\} = C_m \tag{3.10}$$

onde C_m é a capacidade do canal sem ruído com a restrição m-DC.

Prova:A equação (3.6) leva a seguinte desigualdade:

$$\max_{0\le d\le r}[S_m(\vec{x},d)] \le V_m(\vec{x},r) \le r \max_{0\le d\le r}[S_m(\vec{x},d)] \tag{3.11}$$

A primeira equação do lema segue da desigualdade tomando os limites assintóticos como em (3.7) e (3.8). A segunda equação segue de:

$$\dot{V}_m(\vec{x},1) = C_m = \log_2[\lambda_{max}(m)] \tag{3.12}$$

Uma fórmula conveniente para $\lambda_{max}(m)$ pode ser retirada de ([5],p.222), resultando

$$C_m = \log_2[2\cos\frac{\pi}{m+2}] \tag{3.13}$$

A distância relativa $\delta_t(\vec{x},m)$ na qual $\dot{S}_m(\vec{x},\delta) = C_m$ é particularmente significante. Ela diz a taxa assintótica na qual os dígitos de uma sequência típica em X_m (ou seja , uma sequência tirada de um conjunto com tamanho assintoticamente aproximando 100% de$|X_m|$) será diferente dos dígitos correspondentes de $\vec{x}$. Por essa razão $\delta_t(\vec{x},m)$ será chamada de distância típica de $\vec{x}$ em X_m.

Expressões em forma fechada para $\dot{S}_m(\vec{x},\delta)$ podem ser obtidas para alguns centros $\vec{x}$ e para alguns valores pequenos de m, mas sua derivação pode ser difícil devido a complexidade combinatorial. A equação (3.10) irá então fornecer uma checagem conveniente para tais expressões combinatoriais. O seguinte exemplo ilustra esse ponto.

Exemplo 1: Consideremos $\vec{x}$=(...000111000111...) em X_3. Vamos tomar células de 3 dígitos nas quais a sequência central é 000 ou 111. Células com os 3 bits errados em relação a sequência central são proibidas na treliça X_3. Definamos aqui um evento de erro como um conjunto de dígitos que resultam num afastamento temporário do caminho da sequência centro dentro da treliça SDC, mantendo a condição da restrição DC[1]. Todos os eventos de erro têm exatamente 2 erros, e não podem acabar na mesma célula que começaram. Erros irão ocorrer em "corridas" de células erradas, isto é, esses erros aparecem em uma sequência de células com

[1] vide figuras 3.3 e 3.4 para melhor compreensão

pelo menos duas células de duração cada."Corridas" de células erradas podem ou não ser contíguas, porém uma célula com 1 erro irá sempre começar ou finalizar uma "corrida" de células erradas e existem sempre 2 possibilidades diferentes para ocorrência de células de 1 erro. Finalmente, uma célula com 2 erros pode ocorrer estritamente dentro de uma "corrida" de células erradas, com 3 possibilidades de ocorrência.

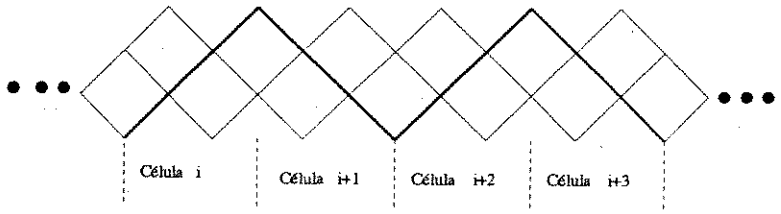


Figura 3.3: Divisão de uma sequência restrita em células

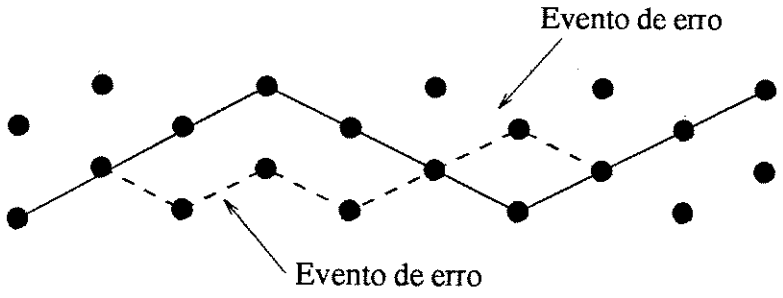


Figura 3.4: Eventos de erro em relação a sequência centro

Se a sequência tem M "corridas" de células erradas, sendo m_i o comprimento(em células) da i_ésimo "corrida", a distância total de Hamming do centro $\vec{x}$ será:

$$d = \sum_{i=1}^{M}(2m_i - 2) = 2\sum_{i=1}^{M} m_i - 2M$$

Seja a sequência de número de erros nas células sucessivas chamada palavra de erro. Desde que o número de possibilidades por "corrida" de erros é 4.3^{m_i-2}, então o número de palavras por palavra de erro é:

$$\prod_{i=1}^{M}\{(4/9)\ .3^{m_i}\} = (4/3)^M\ .3^{d/2}$$

Uma sequência com M "corridas" e distância de Hamming d ao centro $\vec{x}$ irá necessariamente ter $2M$ células de 1 erro e $(d/2 - M)$ células de 2 erros e portanto $(n/3 - d/2 - M)$ células de zero erros. Somando sobre todos os valores possíveis de M, obtemos a seguinte expressão para o tamanho da superfície esférica de Hamming:

$$S_3(\vec{x}, d) = \sum_{M=1}^{d/2} \begin{bmatrix} d/2-1 \\ M-1 \end{bmatrix} \begin{bmatrix} n/3-d/2 \\ M \end{bmatrix} (4/3)^M 3^{d/2},\ \ d \leq 2n/3 \qquad (3.14)$$

$$= 0 \qquad , d > 2n/3$$

Onde o primeiro fator em cada termo da soma é o número de partições das $(d/2 - M)$ células de 2 erros entre M partes não negativas correspondendo as M "corridas", e o segundo fator é o número de partições das $(n/3 - d/2 - M)$ células de zero erros entre $(M+1)$ partes não negativas correspondendo aos (possivelmente vazios) interstícios entre "corridas". O produto dos dois fatores é o número de palavras de erro.Tomando (3.14) aos seus limites assintóticos com $M = \mu n$ resulta (vide apêndice A) a seguinte expressão de taxa que é plotado na figura 3.5:

$$\dot{S}(\vec{x},\delta) = 2\mu_0 + \log_2 \frac{(\frac{\delta}{2})^{\frac{\delta}{2}}(\frac{1}{3}-\frac{\delta}{2})^{\frac{1}{3}-\frac{\delta}{2}}.3^{\frac{\delta}{2}-\mu_0}}{\mu_0^{2\mu_0}(\frac{\delta}{2}-\mu_0)^{\frac{\delta}{2}-\mu_0}(\frac{1}{3}-\frac{\delta}{2}-\mu_0)^{\frac{1}{3}-\frac{\delta}{2}-\mu_0}}, \quad \delta \leq \frac{2}{3} \qquad (3.15)$$

$$= 0 \qquad , d > 2n/3$$

Onde $\mu_0 = \frac{2}{3} - \sqrt{\frac{4}{9} - \delta(\frac{2}{3}-\delta)}$. O parâmetro μ_0 é o valor de μ num termo assintótico dominante em (3.14) e portanto 1/6 da taxa de células de um erro dentre células numa sequência típica à distância relativa δ do centro. A superfície assintótica em (3.15) atinge seu valor máximo C_3 a distância típica $\delta_t(\vec{x},3) = (5+\sqrt{5})/15 \simeq .4824$. Isso significa que uma sequência típica em X_3 é diferente da sequência $\vec{x}$ considerada em aproximadamente 48% dos seus dígitos, enquanto que a sequência não restrita típica é diferente de $\vec{x}$ na metade dos seus dígitos. Isso conclui a discussão do exemplo 1.

Nas próximas seções, nós discutiremos os tamanhos e centros de esferas de Hamming extremas em X_m para vários raios. A discussão será ajudada pelo conceito de camada horizontal na treliça da restrição m-DC como mostrada na figura 3.6.

Devemos dizer que o dígito x_i da sequência $\vec{x}$ está na k_ésima camada de X_m quando o par $\{SDC(x_{i-1}), SDC(x_i)\}$ é $\{S-k, S-k+1\}$ ou $\{S-k+1, S-k\}$ onde S é o maior valor de SDC permitido pela restrição m-DC. Uma "corrida em camadas" de $\vec{x}$ na k_ésima camada é definida como uma sequência de dígitos de $\vec{x}$ que estão todos na k_ésima camada, com os dígitos precedentes e sucessores não contidos na k_ésima camada. "Corridas em camadas" têm a forma (...010101010...). Na primeira camada todas as "corridas em camadas" têm a forma (1010...10) e portanto tem comprimento par. Na última camada (m_ésima), "corridas em camadas" também têm comprimento par e são do tipo (0101...01). Nas camadas internas ,"corridas em camadas" têm comprimento par quando seus dígitos antecessores e sucessores estão na mesma camada e têm comprimento ímpar quando estão em diferentes camadas.

É uma característica da restrição DC que a SDC seja par após um número par de dígitos, e ímpar após um número ímpar de dígitos. Portanto, somente parte dos estados da SDC é acessível como estado inicial em qualquer posição [2] dada na treliça. Se existe um dígito $b \in \{0,1\}$ que pode ser gerado de qualquer estado inicial

[2]Posição dentro da treliça é equivalente ao instante de tempo analisado na treliça

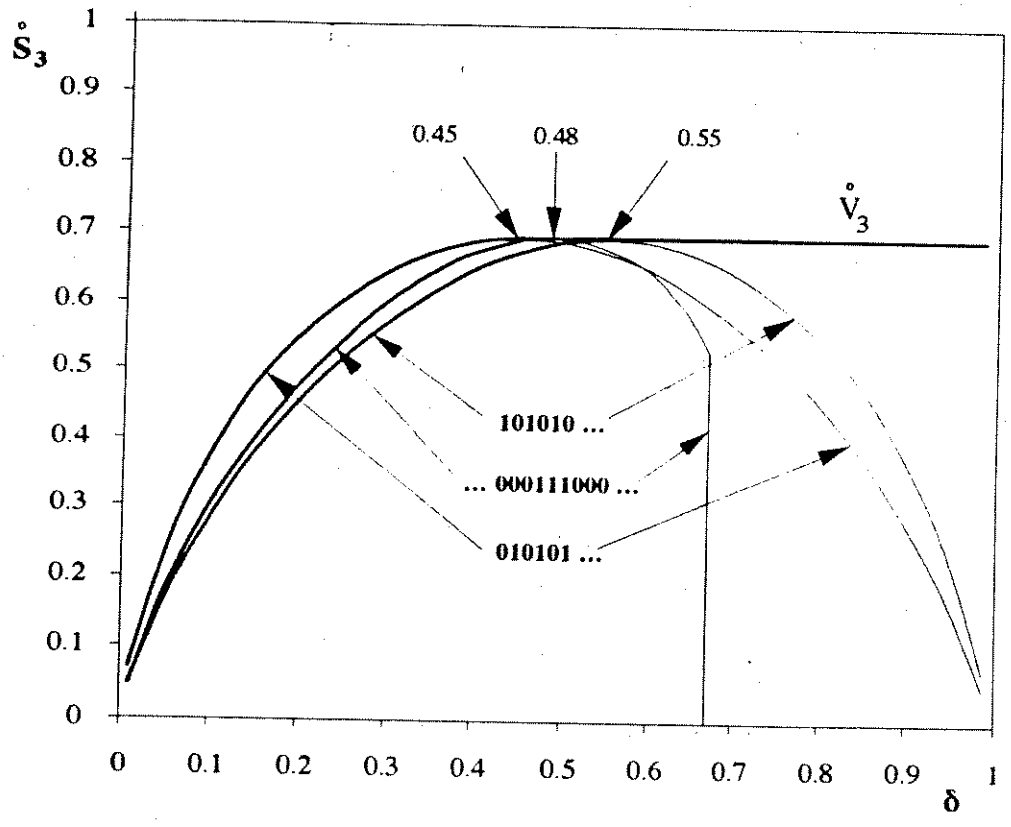


Figura 3.5: Tamanhos de esferas assintóticas em X_3 para alguns centros

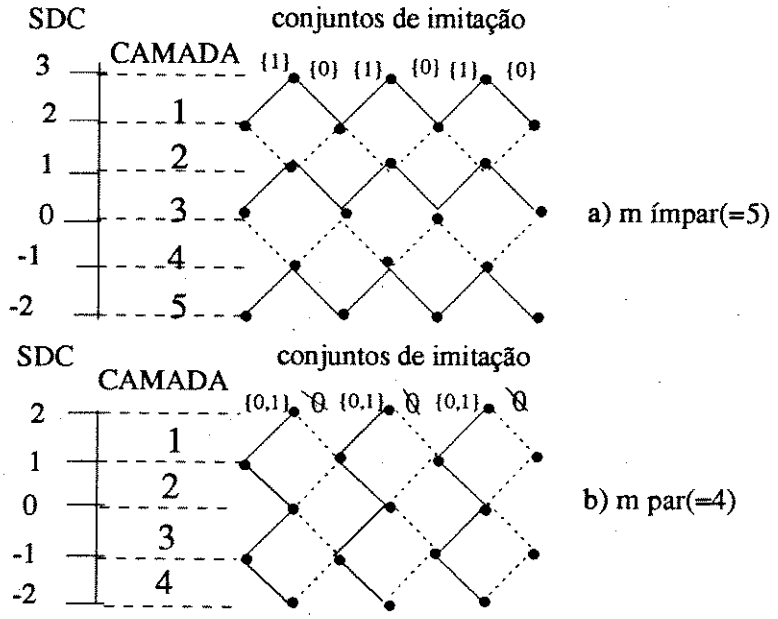


Figura 3.6: padrões de ocorrência de dígitos de imitação

acessível na posição p, devemos dizer que b é um dígito de imitação na posição p. Caso contrario, ele será chamado de dígito de não-imitação na posição p.

Quando m é ímpar, o conjunto de dígitos de imitação alterna entre $\{1\}$ e $\{0\}$ em posições sucessivas após um prefixo finito. Quando m é par, eles alternam entre $\{0,1\}$ e $\emptyset$[3]. A figura 3.6 , na qual dígitos de imitação são representados por linhas cheias e dígitos de não-imitação por linhas partidas, mostra os diferentes padrões de ocorrência de dígitos de imitação na treliça restrita m-DC para m ímpar e m par. Sempre que possível, a identificação de centros e tamanhos de esferas extremas de Hamming, conforme visto anteriormente, levarão à formulação dos limitantes para as taxas assintóticas alcançáveis , definidas como:

$$R_m(\delta) = \lim_{n\to\infty} \frac{1}{n} \log_2 M_m(\delta n) \tag{3.16}$$

onde $M_m(d)$ é o tamanho de um código restrito em X_m e distância mínima de Hamming d entre duas palavras código de comprimento n. O limite assintótico de (3.16) aplicado à expressão (3.3) nos leva ao seguinte piso assintótico para códigos restritos em DC:

$$R_m(\delta) \geq C_m - \max_{\vec{x}\in X_m} [\dot{V}_m(\vec{x}, \delta)] \tag{3.17}$$

O limite assintótico de (3.16) aplicado a expressão (3.5) nos leva ao teto assintótico para códigos restritos em DC:

$$R_m(\delta) \leq C_m - \min_{\vec{x}\in X_m} [\dot{V}_m(\vec{x}, \delta/2)] \tag{3.18}$$

[3] Conjunto vazio

3.3 Esferas de Hamming extremas de raio 2

Nessa seção iremos discutir esferas de Hamming de raio 2, aqui denominadas esferas-2. O problema de identificar mínimo e máximo dentre essas esferas é mais fácil do que para qualquer outro raio positivo.Seja $\overline{X_m}$ o conjunto de sequências binárias balanceadas que não estão contidas em X_m, e sejam $S > 0$ e $s < 0$ os valores máximo e mínimo de SDC permitidos pela restrição m-DC, respectivamente.

Lema 2 *Para qualquer $m > 2$, $V_m(\vec{x}, 2)$ é minimizado pela sequência ...$(1)^{m-2}(0)^{m-2}$ $(1)^{m-2}(0)^{m-2}$..., onde $(b)^k$ representa uma sequência de dígitos b com comprimento k.*

Prova:Já que $V_m(\vec{x}, 2) = 1 + S_m(\vec{x}, 2)$, devemos achar $\vec{x}$ que minimize $S_m(\vec{x}, 2)$ que é o número de eventos de 2 erros [4] em relação a $\vec{x}$ que levam a sequências em X_m. Seja $\overline{S_m}(\vec{x}, 2)$ o número de eventos de 2 erros em relação a $\vec{x}$ que levam a sequências balanceadas em $\overline{X_m}$. Já que

$$S_m(\vec{x}, 2) + \overline{S_m}(\vec{x}, 2) = (\frac{n}{2})^2 \tag{3.19}$$

para qualquer $\vec{x} \in X_m$, podemos procurar pela sequência $\vec{x} \in X_m$ que maximiza $\overline{S_m}(\vec{x}, 2)$. Consideremos dois tipos de evento de erro: eventos de erro positivos, nos quais a sequência errada tem valores de SDC maiores que $\vec{x}$ durante o evento; e eventos de erro negativos, caso contrário.Vamos considerar primeiro os eventos de erro positivos.

Seja $\{u_i\}$ o conjunto de todos os inteiros tal que

$$SDC(x_{u_i}) = S - 1 = u_0, \qquad i = 1, 2, 3..$$

No intervalo das posições $[u_i + 1, u_{i+1}]$, existem $(u_{i+1} - u_i)/2$ zeros e $(u_{i+1} - u_i)/2$ uns. Um evento positivo de erro pode começar em qualquer zero, e levará a uma palavra em $\overline{X_m}$ se e somente se ele acaba em um 1 que ocorre após a posição u_{i+1}. O número total de eventos positivos de 2 erros levando a palavras em $\overline{X_m}$ será:

$$\overline{S_m^+}(\vec{x}, 2) = \frac{1}{4}\sum_{i=0}^{I-1}(u_{i+1} - u_i)(n - u_{i+1} - u_0) \tag{3.20}$$

Onde I é o número de vezes que o nível de SDC u_0 é tocado pela sequência centro.

Definindo $\{v_i\}$ como o conjunto de inteiros tal que

$$SDC(x_{v_j}) = s + 1 = -v_0, \qquad j = 1, 2, 3..$$

O mesmo argumento nos conduz ao número total de eventos negativos de 2 erros levando a palavras em $\overline{X_m}$:

$$\overline{S_m^-}(\vec{x}, 2) = \frac{1}{4}\sum_{j=0}^{J-1}(v_{j+1} - v_j)(n - v_{j+1} - v_0) \tag{3.21}$$

[4] Nessa definição de evento de erro a sequência errada pode sair fora da restrição m-DCm-DC,diferindo portanto da definição no exemplo 1

onde J é o número de vezes que o nível de SDC $-v_0$ é tocado pela sequência centro.

As somas (3.20) e (3.21) são representadas pelas áreas brancas rotuladas com + e - na figura 3.7, respectivamente. Desde que a área do triângulo é quatro vezes o tamanho da esfera-2 dada em (3.19), segue que a área sombreada é quatro vezes o tamanho da esfera-2 restrita. Portanto $V_m(\vec{x}, 2)$ será minimizada por qualquer sequência $\vec{x}$ que gere um conjunto $\{u_i\} \cup \{v_j\}$ que minimize a área sombreada na figura 3.7 com a condição imposta pela treliça de espaçamento mínimo $(m-2)$ entre qualquer membro de $\{u_i\}$ e qualquer membro de $\{v_j\}$.

O centro da esfera-2 mínima em X_m é então formada por corridas alternadas de $(m-2)$ zeros e $(m-2)$ uns, tal que os estados externos de SDC não são tocados exceto quando m ≤ 2. Já que isso implica num movimento de zigzag entre os estados internos de SDC, isso será chamado de padrão zigzag interno.

Lema 3 *$V_m(\vec{x}, 2)$ é maximizado por:*

a) Por qualquer sequência $x \in X_{m-4}$, para $m \geq 5$.

b) Por (110101...0100), para m=3.

c) Por qualquer sequência x=b (10 ou 01)$^{(n-2)/2}$(1-b), para qualquer $b \in \{0,1\}$, para m=4.

d) por qualquer sequência em X_2, para m=2.

Prova: Se $\vec{x} \in X_{m-4}$, $\{u_i\} \cup \{v_j\} = \{u_0, v_0\}$, tal que $V_m(\vec{x}, 2)$ é igual ao tamanho da esfera-2 irrestrita. Se $m = 3$ ou $m = 4$, a área sombreada na figura 3.7 não pode tomar todo o triângulo, mas pode ser maximizado pela supressão de um dos lados brancos: qualquer um para $m = 4$, e o lado esquerdo maior para $m = 3$(já que $u_0 > v_0$ para m ímpar).

O caso $m = 2$ é trivial.

3.4 Esferas de Hamming assintóticas : VSDC ímpares

Nessa seção devemos discutir a extensão dos resultados da seção anterior para qualquer raio quando o desbalanço(VSDC) m é ímpar.

Vamos considerar todas as posições entre um prefixo de tamanho $(m-1)/2$ e um sufixo igualmente longo na treliça fechada[5]. Se $m = 4k+1$ para algum inteiro k, o maior estado de SDC tem SDC= $(2k+1)$ e portanto é acessível após um número ímpar de dígitos: O dígito 0 é então o único dígito de imitação de posições pares, enquanto o dígito 1 é o único dígito de imitação das posições ímpares (ver figura 3.6). Se $m = 4k+3$, o maior SDC permitido é $(2k+2)$, então o dígito 0 é o único dígito de imitação de posições ímpares e o dígito 1 é o único dígito de imitação de posições pares. Em ambos os casos, pode-se ver na treliça de m ímpar (figura 3.6) que todos os dígitos na primeira, última(m_ésima) e todas as outras camadas internas ímpares são dígitos de imitação, enquanto todos os dígitos nas camadas internas pares são dígitos de não imitação.

[5] Treliça fechada é aquela que começa e acaba nos estados de SDC=0

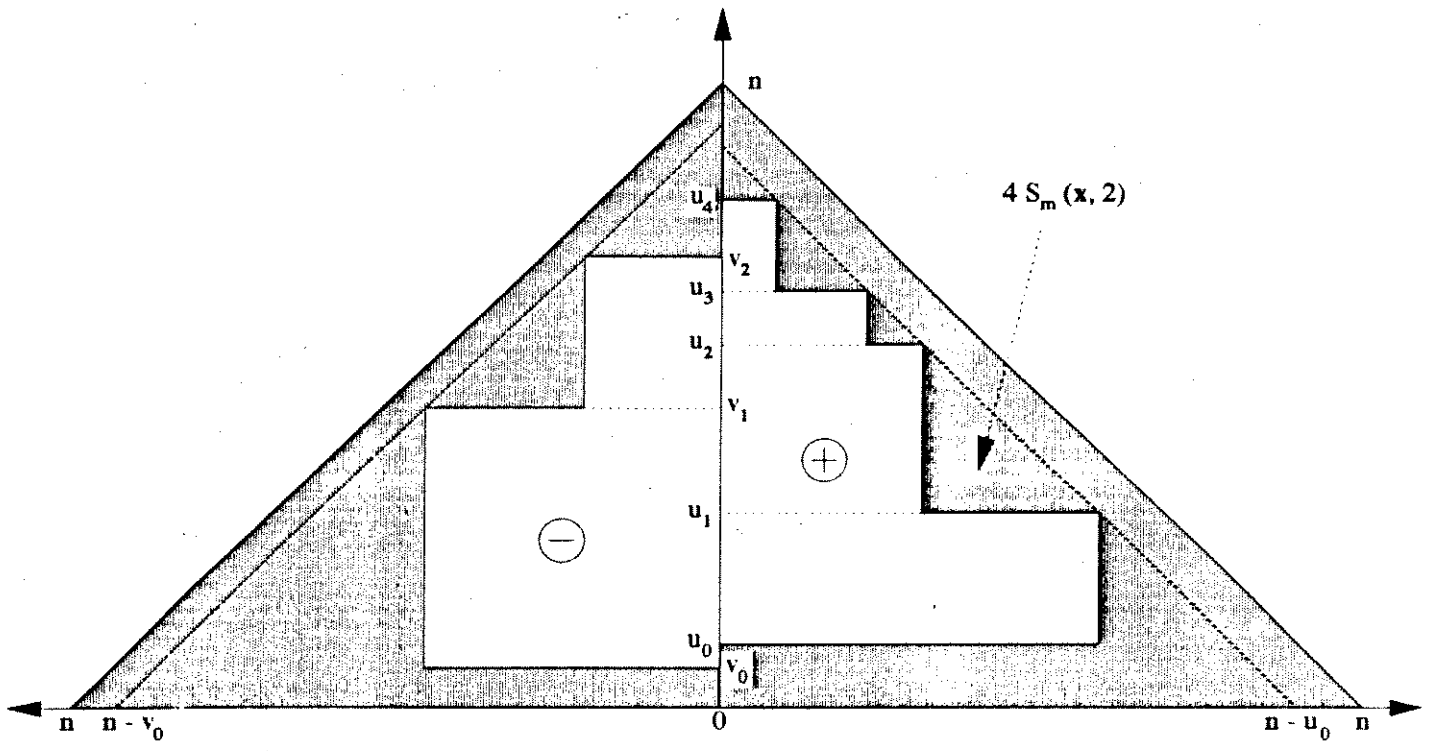


Figura 3.7: Representação geométrica para o tamanho da esfera-2 restrita

Teorema 1 *Seja Z_m o conjunto de sequências em X_m que minimizam $V_m(\vec{x}, r)$ para um dado $m \geq 3$ ímpar. Então existe pelo menos uma sequência $\vec{z} \in Z_m$ com as seguintes propriedades:*

a) $\vec{z} \in X_{m-2}$

b) Se L é uma "corrida em camadas" em qualquer camada ímpar de z em X_m, então L tem comprimento no máximo um.

Prova:Vamos considerar um par de dígitos de imitação $(x_{p+1}, x_{p+2}) = (10)$ em qualquer camada da sequência $\vec{x}$ exceto na m_ésima camada. Seja $\vec{x}'$ a sequência que resulta da substituição de 10 por 01 nas posições (p+1,p+2) de $\vec{x}$: Na treliça , essa substituição corresponde a uma movimentação descendente do caminho 10 na camada par inferior adjacente. Claramente $\vec{x}'$ é restrito se $\vec{x}$ também for.

Vamos numerar os estados de SDC iniciais acessíveis da posição $(p+1)$ como $1, 2, ..., k = (m+1)/2$, em ordem de SDC decrescente. Sejam $V_i(r, q)$ e $V_i'(r, q)$ os tamanhos dos conjuntos(aqui chamados de esferas truncadas) de todas as treliças abertas de sequências restritas de comprimento $q \leq n$ acabando no estado i com distância de Hamming no máximo r de $\vec{x}$ e de $\vec{x}'$, respectivamente.Podemos então escrever(vide figura 3.8):

$$\begin{aligned}
V_1(r, p+2) &= V_1'(r, p+2) = V_1(r, p) + V_1(r-2, p) + V_2(r-1, p) \\
V_i(r, p+2) &= V_i'(r, p+2) = V_{i-1}(r-1, p) + V_i(r, p) + V_i(r-2, p) + V_{i+1}(r-1, p), \quad 2 \leq i \leq k-1 \\
V_k(r, p+2) &= V_{k-1}(r-1, p) + V_k(r, p) \\
V_k'(r, p+2) &= V_{k-1}(r-1, p) + V_k(r-2, p) \leq V_k(r, p+2)
\end{aligned} \tag{3.22}$$

Já que todos os dígitos de $\vec{x}$ e $\vec{x}'$ nas posições $(p+3)$ a n são os mesmos, $V_m(\vec{x}, r)$ e $V_m(\vec{x}', r)$ são dados pelas mesmas combinações lineares de valores de $V_i(r, p+2)$ e $V_i'(r, p+2)$, respectivamente. Portanto, desde que de (3.22): $V_i'(r, p+2) \leq V_i(r, p+2)$ para todo i, segue que substituindo um par de dígitos de imitação 10 em qualquer camada ímpar da sequência central, menos a última(m_ésima), por 01 conduzirá a uma esfera de tamanho menor ou igual a original.O mesmo argumento é válido para qualquer par de imitação 01 em qualquer camada ímpar exceto a primeira, quando é substituido por 10.

Agora pegue qualquer sequência $\vec{x} \in Z_m$. Substitua todos os pares 10 na primeira camada por 01, substitua todos os pares 01 na m_ésima camada por 10, e substitua pares 01 e 10 escolhidos arbitrariamente em camadas internas ímpares por 10 e 01 respectivamente, até que nenhum desses pares (em camadas ímpares) estejam disponíveis. A sequência resultante também está em Z_m e satisfaz às propriedades colocadas. □ vspace*1cm Deve ser notado que o padrão zigzag do lema 2 estão em conformidade com as condições do teorema 1 com máxima ocupação de camadas internas ímpares de imitação. Infelizmente, o teorema 1 não aponta em geral o centro mínimo para qualquer $r > 2$, mas o faz para $m = 3$ conforme mostra o seguinte corolário.

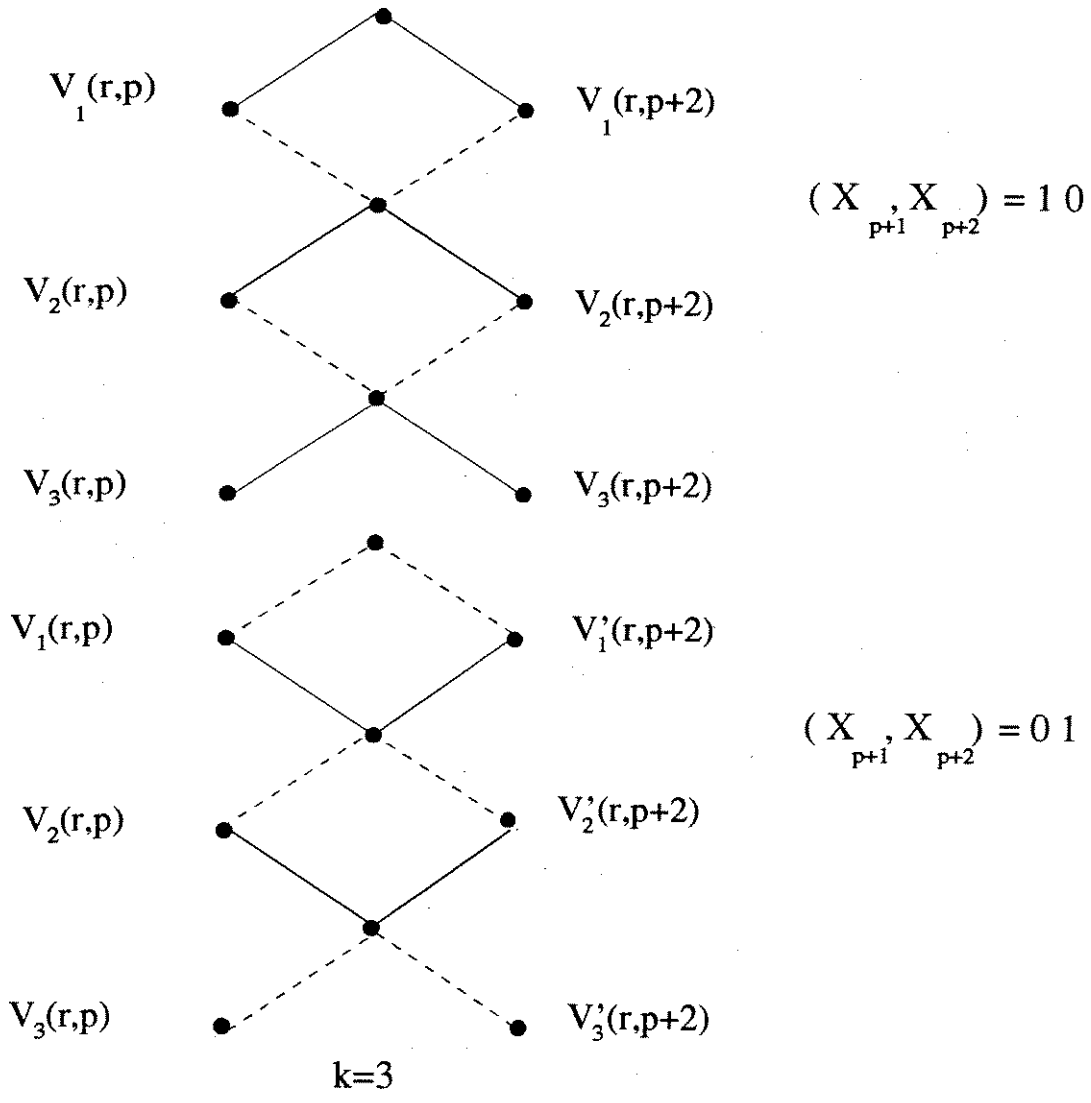


Figura 3.8: Esquema de treliça para as esferas truncadas usadas na prova do teorema 1

corolário 1 *$V_3(\vec{x}, r)$ é minimizado por $\vec{x} = (1010...10)$ para qualquer r.*

Prova:Já que $(1010...10)$ é a única sequência em X_1, ela é também a única sequência em X_3 que satisfaz a primeira condição do teorema 1. □

Os argumentos do teorema 1 podem também ser usados para derivar condições para centros maximizantes no espaço m-DC ímpar , mas os resultados são mais fracos, como veremos no próximo teorema.

Teorema 2 *Seja Y_m o conjunto de sequências em X_m que maximiza $V_m(\vec{x}, r)$ para um dado m ímpar ≥ 3. Então existe pelo menos uma sequência $\vec{y} \in Y_m$ com a*

seguinte propriedade: se L é uma "corrida em camadas" em qualquer camada par de $\vec{y} \in X_m$,então L tem comprimento no máximo um.

Prova:Como no teorema 1. □

Um limitante superior para $R_3(\delta)$ pode ser derivado do corolário 1. Outros limitantes podem ser derivados a partir da identificação de esferas extremas para raios assintóticos δn quando n vai para infinito, como no próximo teorema.

Teorema 3 *Sejam I_p e $\overline{I}_p$ os conjuntos de dígitos de imitação e de não imitação na posição p do espaço restrito m-DC,respectivamente. Então para m ímpar ≥ 3, $\dot{V}_m(\vec{x}, \delta)$ é:*

a) Maximizado por qualquer sequência $\vec{y}$ tal que $y_k \in I_k$ para todo $k \in [(m+1)/2, n-(m-1)/2]$;

b) Minimizado por qualquer sequência $\vec{z}$ tal que $z_k \in \overline{I}_k$ para todo $k \in [(m+1)/2, n-(m-1)/2]$

Prova: Se $S = (m+1)/2$, $k \in [S, n-S+1]$, e m é ímpar, temos da figura 3.6 que $I_k = \{1\}$ e $\overline{I}_k = \{0\}$ para $k = S, S+2, S+4, ...$; e $I_k = \{0\}$ e $\overline{I}_k = \{1\}$ para $k = S+1, S+3, S+5, ...$. Portanto I_k e $\overline{I}_k$ nunca são vazios no intervalo em consideração, e sequências com as propriedades citadas existem.

Seja $\vec{x} \in X_m$ uma sequência com γn dígitos de não imitação no intervalo $[S, n-S+1]$ de posições, para algum $\gamma > 0$. Seja $\vec{y}$ resultante da substituição de todos os dígitos de não imitação de $\vec{x}$ por dígitos de imitação. Claramente, ou $\vec{y}$ tambem é restrito, ou ele pode ser trazido de volta para o espaço restrito trocando um número finito de dígitos no sufixo de comprimento $(m-1)/2$.

Vamos considerar a máquina de estados finitos que gera palavras em X_m, nas posições $p \in [S, n-S+1]$. Qualquer que seja a história passada da máquina, ela é sempre livre para imitar $\vec{y}$, sempre com um estado onde ela é compelida a imitar $\vec{y}$. Por outro lado, existem $\gamma n > 0$ passos nos quais a máquina é livre para gerar um erro em relação a $\vec{x}$, com o erro sendo compulsório a partir de um dos estados. Portanto, a máquina terá pelo menos o mesmo número de oportunidades de gerar palavras com no máximo $\delta(n-m+1)$ erros em (n-m+1) passos a partir de $\vec{y}$ que a partir de $\vec{x}$. Levando n para infinito, a primeira parte do teorema é provado. Argumentos similares provam a segunda parte. □

O teorema 3 permite identificar centros assintoticamente maximizantes e minimizantes para todo δ e todo m ímpar, como veremos no seguinte corolário.

corolário 2 *Para todo m ímpar, $\dot{V}_m(\vec{x}, \delta)$ é:*

a) Maximizado por $\vec{x} = (010101...)$ *e minimizado por* $\vec{x} = (101010...)$, *para* $m = 3, 7, 11, ...$;

b) Maximizado por $\vec{x} = (101010...)$ *e minimizado por* $\vec{x} = (010101...)$ *para* $m = 5, 9, 13, ..$;

Prova: Imediata a partir do teorema 3. □

Já que centros assintóticamente maximizantes e minimizantes estão agora identificados, o cálculo do limitante inferior (3.17) e do limitante superior (3.18) requer somente a determinação da taxa da superfície esférica assintótica $\dot{S}_m(\vec{x}, \delta)$ para tais centros e a aplicação de (3.13) e o lema 1. Uma vez que $\dot{S}_m(\vec{x}, \delta)$ é determinado para uma das esferas extremas para um dado m, a outra pode ser derivada de:

$$S_m(x, d) = S_m(\vec{1} - \vec{x}, n - d), \tag{3.23}$$

$$\Rightarrow \dot{S}_m(101010..., \delta) = \dot{S}_m(010101..., 1 - \delta), \tag{3.24}$$

onde $\vec{1}$ é o vetor tudo-1, e (3.23) pressupõe que $(\vec{1} - \vec{x}) \in X_m$. Os próximos dois exemplos discutem tais derivações de limitantes superiores e inferiores para $m = 3$ e $m = 5$.

Exemplo 2: Vamos considerar a combinatória de erros a partir do centro $\vec{x} = (101010...)$ em X_3. Todos os eventos de erro têm exatamente 2 erros. Um evento pode começar em qualquer posição, ma deve acabar na próxima posição: erros ocorrem em pares. O tamanho da superfície esférica de Hamming com centro em $\vec{x}$ e raio d é então:

$$S_3(101010..., d) = \begin{bmatrix} n - d/2 \\ d/2 \end{bmatrix} \tag{3.25}$$

Levando (3.25) ao limite assintótico resulta:

$$\dot{S}_3(101010..., \delta) = (1 - \frac{\delta}{2}) \log_2 (1 - \frac{\delta}{2}) - (1 - \delta) \log_2 (1 - \delta) - \frac{\delta}{2} \log_2 \frac{\delta}{2} \tag{3.26}$$

$$\delta_t(101010..., 3) = 1 - \frac{\sqrt{5}}{5} \simeq 0.5528 \tag{3.27}$$

A equação (3.26) é plotada na figura 3.5, onde ela pode ser comparada com $\dot{S}_3(010101..., \delta)$ obtida da mesma através de (3.24) , e com $\dot{S}_3(...000111000...)$ do exemplo 1.

Exemplo 3: Vamos considerar a combinatória dos erros a partir do centro maximizante $x = (101010...)$ em X_5. Somente eventos de 2 erros são permitidos. Eventos podem começar em qualquer lugar, e devem ter comprimento par. Vamos chamar de palavra de evento a sucessão de comprimentos de eventos e interstícios entre eles. Então existe uma palavra por palavra de evento, e as palavras de evento são dadas por $a_0 e_1 a_1 e_2 a_2 ... e_{d/2} a_{d/2}$, onde a_i é qualquer número não negativo, e_i é qualquer número positivo par , e

$$\sum_{i=0}^{d/2} a_i + \sum_{i=1}^{d/2} e_i = n$$

A superfície esférica de Hamming com centro $\vec{x}$ e raio d tem o seguinte tamanho:

$$S_5(101010..., d) = \sum_{i=d/2}^{n/2} \begin{bmatrix} i-1 \\ d/2-1 \end{bmatrix} \begin{bmatrix} n-2i+d/2 \\ d/2 \end{bmatrix} \tag{3.28}$$

Onde i é a metade da soma total de todos os comprimentos de eventos; o primeiro fator em cada somando da soma é o número de partições de i entre $d/2$ partes positivas correspondentes aos $d/2$ meio comprimentos de evento; e o segundo fator é o número de partições dos $(n-2i)$ dígitos restantes entre as $(1+d/2)$ partes não negativas correspondendo a (possivelmente vazios) interstícios entre eventos. Levando (3.28) ao limite assintótico e fazendo $i = \mu n$, resulta:

$$\dot{S}_5(1010..., \delta) = \delta + \log_2 \frac{{\mu_o}^{\mu_0}(1-2\mu_0+\delta/2)^{1-2\mu_0+\delta/2}}{\delta^{\delta}(\mu_0-\delta/2)^{\mu_0-\delta/2}(1-2\mu_0)^{1-2\mu_0}} \tag{3.29}$$

Onde $\mu_0 = \frac{5\delta+12-\sqrt{16-8\delta-7\delta^2}}{32}$.

A equação (3.29) , na qual μ_0 é o valor assintóticamente dominante de (i/n), é plotada na figura 3.9, em conjunto com taxas assintóticas de superfícies esféricas de centros maximizantes de outras restrições VSDC. Uma determinação numérica de seu pico revela que

$$\delta_t(101010..., 5) \simeq 0.4834 \tag{3.30}$$

Em princípio a metodologia dos exemplos 2 e 3 poderia ser usada em conjunto com (3.17),(3.18),(3.13),(3.9),(3.10),(3.23) e corolário 2, para derivar limitantes inferiores e superiores para $m = 7, 9, 11$,etc., porém a combinatória de evento de erros apartir de centros(...01010...) torna-se extremamente complicada com o alargamento da treliça. Uma aproximação sistemática para esse problema é um tópico interessante para pesquisa.

A identificação de esferas extremas foi conseguida nessa seção para distâncias $d = \delta n$ e m ímpar, e para $d = 2$ na seção anterior para qualquer m. Para d finito, somente o centro minimizante para $m = 3$ foi identificado no corolário 1. Para os outros casos de d finito, os teoremas 1 e 2 nos dão somente famílias de candidatos possíveis para centros minimizantes e maximizantes, respectivamente. Já que sabemos os padrões de maximização e minimização nos dois extremos de d, podemos formular conjecturas sobre o comportamento intermediário através de uma educada interpolação. No caso de centros maximizantes, por exemplo, conhecemos que pelo menos alguns dos centros maximizantes da esfera-2 no lema 3 também se qualificam como centros maximizantes assintóticos sob as condições do teorema 3: Seria então razoável esperar que esses centros são uniformemente maximizantes, ou seja, que maximizem esferas de qualquer raio finito.

No caso de centros minimizantes, a interpolação é mais intrigante. Para $m = 3$, lema 2 e corolários 1 e 2 todos identificam o mesmo centro (para todo d e todo δ) uniformemente minimizante (101010...) em X_3. Para $m = 5$, entretanto, o teorema 1 dá somente a família de possíveis candidatos para centros minimizantes, enquanto

o lema 2 e o corolário 2 apontam para dois diferentes membros da família: enquanto $V_5(\vec{x},2)$ é minimizado pelo padrão zigzag interno (0111000111...), $V_5(\vec{x},\delta)$ é minimizado pelo padrão tudo não imitação (010101...). A implicação é que com o crescimento de d a partir de 2 até algum valor assintótico δn com $\delta > 0$ e n indo para infinito, o padrão minimizante evolui do zigzag interno para tudo não imitação, sempre obedecendo as condições do teorema 1. É interessante notar que a taxa α de dígitos de imitação no padrão zigzag interno dada por:

$$\alpha_{zz} = \frac{m-3}{2m-4} \tag{3.31}$$

é a maior taxa de imitação possível dentro da família de possíveis centros minimizantes do teorema. Assim, parece ser uma boa conjectura que os centros minimizantes para d finito têm sua taxas de imitação α entre zero e α_{zz}, com α decrescente para d crescente. Para $m = 3$, $\alpha_{zz} = 0$ e a família de possíveis padrões minimizantes é reduzida a somente uma sequência (corolário 1).

3.5 Esferas de Hamming assintóticas: VSDC pares

Quando m é par, o conjunto de dígitos de imitação é vazio em uma posição sim outra não da treliça entre os prefixos e sufixos de comprimento $(m-2)/2$. Por essa razão, o teorema 3 não se aplica. Entretanto, a mesma abordagem geral da seção anterior pode ainda ser aplicada através do uso de pares de dígitos de imitação e de não imitação, ao invés de dígitos únicos. Porém , essa abordagem requer que o par seja definido na fase apropriada da treliça.

Na seguinte discussão, devemos considerar pares dos quais a primeira posição tem a mesma paridade que $m/2$, tal que as posições de começo de pares são $..., m/2, 2+m/2, 4+m/2, ...$. Isso significa que todos os estados de SDC iniciais acessíveis são estados internos(não é o primeiro nem o último), com SDC $\in \{-1+m/2, -3+m/2, ..., 1-m/2\}$. Para $m = 4k$, tais pares deixarão o primeiro e o último (n_ésimo) dígitos da sequência sem pares. Para $m = 4k+2$, não restarão esses dígitos sem pares. Um exame na treliça m-par (figura 3.6) mostra que o conjunto de pares de imitação é $\{01, 10\}$ em todas as posições .Para $m \geq 4$, o conjunto de pares de não imitação é $\{00, 11\}$ em todas as posições, enquanto que para $m = 2$ ele é vazio sempre. Logo, todas as sequências em X_2 são sequências de tudo imitação em X_2. O próximo lema estabelece uma relação de equivalência entre pares de imitação.

Lema 4 *Sejam $\vec{x}$ e $\vec{y}$ duas palavras no espaço restrito X_m, com m par, e seja $i \in \{..., m/2, 2+m/2, ...\}$. Se $(x_i, x_{i+1}) = (01)$, $(y_i, y_{i+1}) = (10)$ e $x_k = y_k$ para todo $k \in [1, i-1] \cup [i+2, n]$, então $V_m(\vec{x}, r) = V_m(\vec{y}, r)$ para qualquer $r \leq n$.*

Prova: Vamos numerar os estados iniciais acessíveis de SDC na posição i como $1, 2, ..., m/2$ em ordem de SDC decrescente. Sejam $V_j(r,q)$ e $V_j'(r,q)$ os tamanhos das esferas truncadas de sequências de comprimento $q \leq n$ acabando no estado j e

com raio r dos centros $\vec{x}$ e $\vec{y}$, respectivamente. Podemos então escrever:

$$\begin{aligned} V_1'(r,i+1) &= V_1(r,i+1) = V_1(r,i-1) + V_1(r-2,i-1) + V_2(r-1,i-1) \\ V_{m/2}(r,i+1) &= V_{m/2}'(r,i+1) = V_{m/2}(r,i-1) + V_{m/2}(r-2,i-1) + V_{-1+m/2}(r-1,i-1) \\ V_j(r,i+1) &= V_j'(r,i+1) = V_j(r,i-1) + V_j(r-2,i-1) + V_{j-1}(r-1,i-1) + \\ & \quad V_{j+1}(r-1,i-1) \end{aligned} \tag{3.32}$$

$$j = 2,3,...,-1+m/2.$$

Desde que todos os dígitos de $\vec{x}$ e $\vec{y}$ nas posições $(i+2)$ até n são os mesmos, $V_m(\vec{x},r)$ e $V_m(\vec{y},r)$ são dados pela mesma combinação linear de valores de $V_j(r,i+1)$ e $V_j'(r,i+1)$, respectivamente. O lema segue então da igualdade entre $V_j(r,i+1)$ e $V_j'(r,i+1)$, mostrada acima para todo j. □

Quando $m = 2$, o lema 4 resulta no fato que $V_2(\vec{x},r)$ é independente do centro $\vec{x} \in X_2$. O próximo lema apresenta uma relação de simetria.

Lema 5 *Se m é par, então:*

$$S_m(\vec{x},d) = S_m(\vec{x},n-d) \quad \forall \vec{x} \in X_m$$

Prova: Se m é par, $(\vec{1}-\vec{x}) \in X_m$ para todo $\vec{x} \in X_m$, e mapeando cada sequência $\vec{x}$ em $(\vec{1}-\vec{x})$ resulta em:

$$S_m(\vec{x},d) = S_m(\vec{1}-\vec{x},d) \tag{3.33}$$

O lema segue de (3.23) e (3.33). □

Combinando os lemas 1 e 5, temos:

$$\dot{S}_m(\vec{x},1/2) = \dot{V}_m(\vec{x},1/2) = C_m, \tag{3.34}$$

para todo $\vec{x} \in X_m$, m par,

e portanto $\delta_t(\vec{x},m) = 1/2$ para todo $\vec{x}$ quando m é par. Agora apresentaremos um teorema para os centros maximizantes das esferas em X_m.

Teorema 4 *Para qualquer m par, $\dot{V}_m(\vec{x},\delta)$ é maximizado:*

a) por qualquer sequência $b(10$ ou $01)^{(n-2)/2}(1-b)$, para qualquer $b \in \{0,1\}$, para $m = 4,8,12...$

b) por qualquer sequência $(10$ ou $01)^{n/2}$, para $m = 2,6,10...$

Prova: Seja $N(\alpha)$ o número de sequências em X_m que tem $\alpha n/2$ pares de imitação (01 ou 10) e $(1-\alpha)n/2$ pares de não imitação (11 ou 00). Pares podem ser arranjados a partir dos conjuntos de imitação e de não imitação em quaquer ordem de conjuntos sucessivos. Um par de imitação pode sempre ser 01 ou 10, independente da história passada. Nas séries de pares de não imitação , as regras para ocorrência de 11 e 00 são as mesmas das de ocorrência de dígitos em $X_{(m-2)/2}$, já que cada par causa um passo de SDC ± 2 e move a SDC ao longo do conjunto dos $m/2$ estados iniciais permitidos, enquanto pares de imitação não mudam o estado de SDC. Portanto:

$$N(\alpha) = \begin{bmatrix} n/2 \\ \alpha n/2 \end{bmatrix} 2^{\alpha n/2} \lambda_0^{(1-\alpha)n/2+\Omega[(1-\alpha)n/2]} \tag{3.35}$$

onde:

$$\lambda_0 = \lambda_{max}\left[\frac{m-2}{2}\right] = 2\cos\frac{2\pi}{m+2} \tag{3.36}$$

e $\Omega(n)$ é tal que $\Omega(n)/n$ desaparece quando n vai para infinito, com $\Omega(0) = 0$. Tomando (3.35) aos limites assintóticos resulta:

$$\dot{N}(\alpha) = \lim_{n\to\infty} \frac{1}{n} \log_2 N(\alpha) = \frac{1}{2}[\alpha + (1-\alpha)\log_2 \lambda_0 + H(\alpha)], \tag{3.37}$$

onde $H(z) = -z\log_2 z - (1-z)\log_2(1-z)$ é a função entropia. A taxa $\dot{N}(\alpha)$ é maximizada pela seguinte taxa de pares de imitação:

$$\alpha_0 = \frac{2}{2+\lambda_0} = \frac{2}{[\lambda_{max}(m)]^2} \tag{3.38}$$

Portanto, uma sequência restrita aleatoriamente escolhida tem probabilidade 1 de ter taxa α_0 de pares de imitação. Consideremos agora um centro $\vec{x}$ tendo $\beta n/2$ pares de imitação e $(1-\beta)n/2$ pares de não imitação. A probabilidade de distribuição assintótica da distância de Hamming d entre o centro $\vec{x}$ e a sequência restrita aleatoriamente escolhida será:

$$prob(d) = \frac{S_m(\vec{x}, d)}{|X_m|} = prob\left[\sum_{i=0}^{\beta n/2} a_i + \sum_{i=0}^{(1-\beta)n/2} b_i\right], \tag{3.39}$$

onde a_i e b_i são variáveis aleatórias correspondentes a incremento de distância durante pares centrais de imitação e de não imitação, respectivamente. A distribuição de probabilidade assintótica das variáveis a_i e b_i são, para assintoticamente todo i:

$$p(a_i) = \alpha_0/2 \quad , a_i = 0, a_i = 2 \tag{3.40}$$

$$p(a_i) = 1-\alpha_0 \quad , a_i = 1$$

$$p(b_i) = (1-\alpha_0)/2 \quad , a_i = 0, a_i = 2 \tag{3.41}$$

$$p(b_i) = \alpha_0 \quad , b_i = 1$$

Já que $< a_i >=< b_i >= 1$, todos os centros levarão a $< d >= n/2$, que corrobora (3.34). Desde que $0 \leq \lambda_0 < 2$ a partir de (3.36), temos $1/2 < \alpha_0 \leq 1$ de (3.38),

e portanto $\sigma_a^2 = \alpha_0 > (1-\alpha_0) = \sigma_b^2$, onde σ_a^2 e σ_b^2 são as variançcas das variáveis assintóticas a_i e b_i, respectivamente. Já que qualquer par de imitação pode ser sempre 01 ou 10 em qualquer momento de ocorrência, independente de sua posição e pares anteriores, as váriaveis a_i são independentes entre si e de qualquer b_i. Portanto, $(\sum a_i)/\sqrt{n})$ é assintóticamente gaussiana, do teorema do limite central. Já que a sucessão de pares de não imitação 11 e 00 não é livre, mas é regrado dentro da restrição, as variáveis b_i são correlatadas. Portanto, $(\sum b_i)$ tem uma distribuição mais estreita do que ela teria se os b_i's fossem independentes, que é mais estreita do que uma soma de a_i's independentes com o mesmo número de somandos.

Podemos dizer que a distribuição (3.39) é mais larga quando $\beta = 1$, ou seja, quando o centro é uma sequência de tudo imitação. No domínio d, no qual prob(d) deve ter soma unitária, a distribuição mais larga é realmente a mínima numa vizinhança da média comum $n/2$ e máxima nas caudas. Já que a distribuição é gaussiana para $\beta = 1$ no domínio $\epsilon = (d/\sqrt{n})$, ela irá cruzar qualquer outra distribuição para $\beta < 1$ em no máximo um ponto em cada lado da média $< \epsilon >= \sqrt{n}/2$, a algum espaçamento finito desta média. No domínio $\delta = d/n$, esse espaçamento colapsa assintoticamente para zero, e o teorema é provado. O lema 1 e (3.34) mostram que as distâncias de cruzamento são realmente mapeadas em $\delta = 1/2$ no domínio δ □.

Para $m = 2$ calcula-se o piso $R_2(\delta) \geq [1 - H(\delta)]/2$ para $\delta \leq 1/2$ como consequência direta do método construtivo do ítem 4.2.3 do próximo capítulo, ou seja: existe uma correspondência de um para um entre códigos irrestritos (n, k, d) e códigos restritos em X_2 $(2n, k, 2d)$; portanto temos como limitante inferior a metade do limitante inferior irrestrito(Gilbert-Varshamov). Uma extensão para esse limitante inferior para todo $m \geq 4$ par pode agora ser obtido do seguinte teorema.

Teorema 5 *Para todo $m \geq 4$ par e $\delta \leq 1/2$, existem códigos tais que:*

$$R_m(\delta) \geq C_m + \frac{1}{2} - (\delta - 2\mu_0)\log_2 \lambda_0 + \mu_0 \log_2 \mu_0 + (\delta - 2\mu_0)\log_2(\delta - 2\mu_0) + (\frac{1}{2} - \delta + \mu_0)\log_2(\frac{1}{2} - \delta + \mu_0), \tag{3.42}$$

onde:

$$\mu_0 = \frac{1}{2}\{\delta + \frac{\lambda_0^2}{2(4-\lambda_0^2)} - \sqrt{[\delta + \frac{\lambda_0^2}{2(4-\lambda_0^2)}]^2 - \frac{4\delta^2}{4-\lambda_0^2}}\} \tag{3.43}$$

e λ_0 é dado por (3.36)

Prova: Seja m par, e consideremos a sequência central $\vec{x} = (..., 10 \text{ ou } 01, 10 \text{ ou } 01, 10 \text{ ou } 01, ...)$ como especificado pelo teorema 4. Em qualquer célula de 2 dígitos definida pelas vírgulas, sequências em X_m podem apresentar zero, um ou dois erros em relação a $\vec{x}$. Existe sempre uma possibilidade de célula de zero erros e uma possibilidade de célula de 2 erros de qualquer estado dado. As regras para ocorrência das células de 1 erro (11 ou 00) são as mesmas que para ocorrência de dígitos em

$X_{(m-2)/2}$, já que células de um erro causam um passo de SDC de ± 2 e move a SDC entre os $m/2$ estados iniciais permitidos.

Vamos chamar à sequência dos números de erro por célula de 2 dígitos de uma palavra de erro. As palavras de erro consistem em sequências de zeros, uns e dois em qualquer ordem. Portanto, se $i \leq d/2$ é o número de células de 2 erros, e (d-2i) o número de células de um erro, o tamanho da superfície esférica com centro $\vec{x}$ e raio d será:

$$S_m(\vec{x}, d) = \sum_{i=0}^{d/2} \begin{bmatrix} n/2 \\ i \end{bmatrix} \begin{bmatrix} n/2 - i \\ d - 2i \end{bmatrix} \lambda_0^{d-2i+\Omega(d-2i)}, \tag{3.44}$$

onde $\Omega(0) = 0$ e $\Omega(n)/n$ desaparece quando n vai para infinito. Em cada somando de (3.44), o termo exponencial é o número de palavras por palavra de erro, e o produto das combinatórias é o número de palavras de erro.

Fazendo $i = \mu n$ e levando (3.44) aos limites assintóticos resulta:

$$\dot{S}_m(\vec{x}, \delta) = \max_{0 \leq \mu \leq \delta/2} \{-\frac{1}{2} + (\delta - 2\mu)\log_2 \lambda_0 - \mu \log_2 \mu - (\delta - 2\mu)\log_2(\delta - 2\mu) - (\frac{1}{2} - \delta + \mu)\log_2(\frac{1}{2} - \delta + \mu)\} \tag{3.45}$$

Diferenciando a expressão entre chaves e tomando a única raiz no intervalo permitido resulta μ_0 como em (3.43). Substituindo ele em (3.45) e usando (3.17) e o teorema 4 irá então provar o teorema . □

A figura 3.9 mostra as curvas assintóticas $\dot{S}(\vec{x}, \delta)$ quando $\vec{x}$ é um centro maximizante para vários valores pares de m, com $m = 3$ e $m = 5$ incluidos para comparação. A curva para $m = 100$ está dentro da resolução gráfica da função entropia irrestrita $H(\delta)$. O próximo teorema começa a discussão de centros minimizantes.

Teorema 6 *Se $m \geq 4$ é par, então $\dot{V}_m(\vec{x}, \delta)$ é minimizado por uma sequência $\vec{y}$ tal que $y_i = y_{i+1}$ para todo $i \in \{m/2, 2 + m/2, ...n - m/2\}$.*

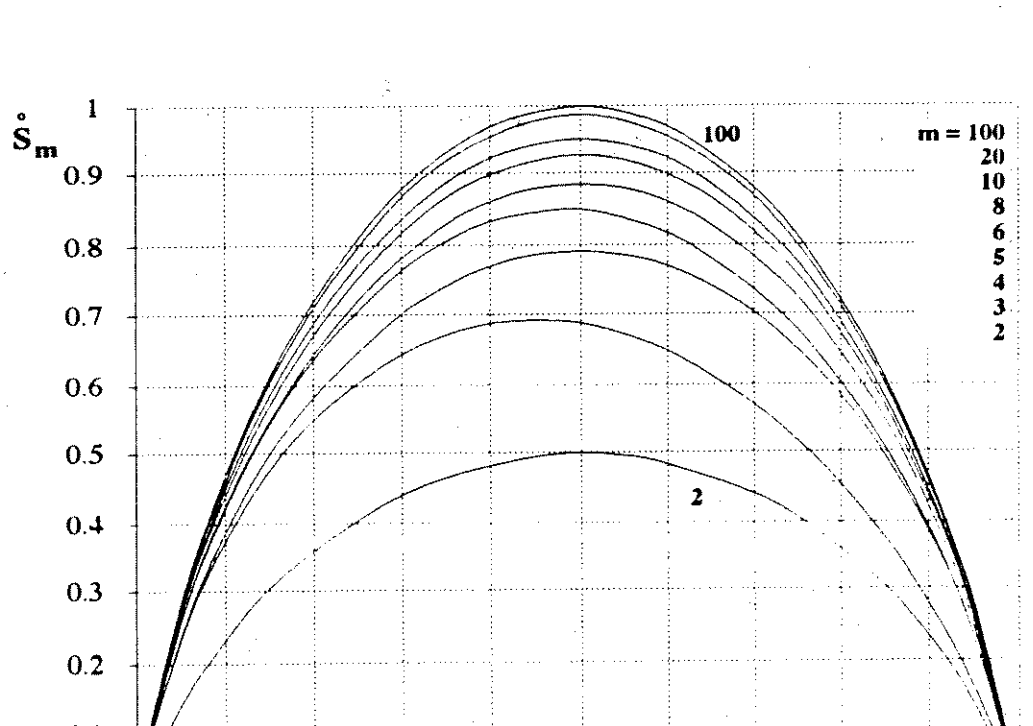
Prova: Os mesmos argumentos usados na prova do teorema 4 mostram que $\dot{V}_m(\vec{x}, \delta)$ é minimizado para uma sequência central com $\beta = 0$, na qual todos os pares são os de não imitação 00 e 11.

Embora os teoremas 4 e 6 sejam de natureza muito similar, os resultados que podem ser derivados a partir do teorema 6 são mais fracos, devido a ausência de uma relação de equivalência entre os pares de não imitação 11 e 00, tal como é estabelecido pelo lema 4 entre os pares de imitação 01 e 10. Uma exceção é o caso $m = 4$, para o qual centros assintóticamente minimizantes são identificados no seguinte corolário.

corolário 3 *$\dot{V}_4(\vec{x}, \delta)$ é minimizado por $b(1-b)^2b^2(1-b)^2...$ para qualquer $b \in \{0, 1\}$.*

Prova: As duas sequências propostas são as únicas que satisfazem as condições do teorema 6 em X_4, e geram esferas de tamanhos iguais. □

Figura 3.9: $\dot{S}_m(\vec{x}, \delta)$ quando $\vec{x}$ é o centro da esfera maximizante para vários m.



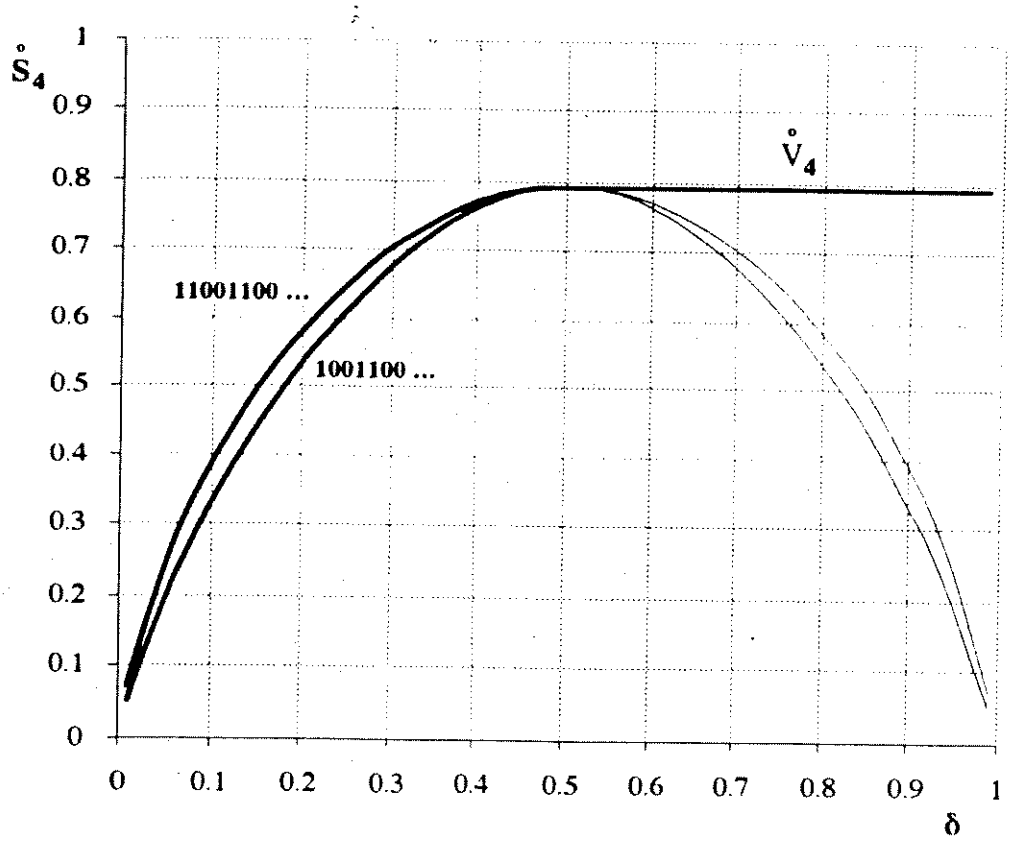


Figura 3.10: $\dot{S}_4(\vec{x}, \delta)$ e $\dot{V}_4(\vec{x}, \delta)$ para centros de esfera máxima (11001100...) e mínima (1001100...).

A combinatória de erros a partir do centro (1001100...) em X_4 pode ser encontrada com o mesmo método analítico do exemplo 1, usando células de 2 dígitos nas quais o centro é 11 ou 00. Se $M = \mu n$ é o número de corridas de células com erro e μ_0 o valor assintótico de μ, temos:

$$\dot{S}_4(1001100..., \delta) = -\frac{1}{2}[1+H(\delta)]+2\mu_0(1-\log_2 \mu_0)-(\frac{\delta}{2}-\mu_0)\log_2(\frac{\delta}{2}-\mu_0)-(\frac{1-\delta}{2}-\mu_0)\log_2(\frac{1-\delta}{2}-\mu_0) \tag{3.46}$$

e $\mu_0 = \frac{1-\sqrt{1-3\delta(1-\delta)}}{3}$,

onde $4\mu_0$ é a taxa de células de um erro e $(\delta - 2\mu_0)$ a taxa de células de 2 erros na sequência típica de X_4 a distância relativa δ de (1001100...). Um limitante superior para $R_4(\delta)$ pode ser calculado a partir do corolário 3, (3.18),(3.9),(3.13) e (3.46). A figura 3.10 mostra curvas para $\dot{S}_4(\vec{x}, \delta)$ para os centros maximizantes e minimizantes (11001100...) e (1001100...), respectivamente.

Considerando os centros minimizantes para $m \geq 6$ par, devemos especular. O teorema 6 qualifica um número infinito de sequências candidatas, e nenhuma

relação de equivalência é conhecida entre os tamanhos das esferas que eles centram em X_m. Parece que o centro minimizante é a sequência central que faz as variáveis b_i em (3.39) maximamente correlatadas, pois isso geraria a distribuição $\dot{S}_m(\vec{x}, \delta)$ mais estreita, e todas as distribuições têm pico no mesmo ponto. Como deve o centro se comportar para que se façam as variáveis b_i maximamente correlatadas? Uma maneira natural é usar um padrão periódico com período igual ou comensurável com o período médio de ocorrência de pares de não imitação nas sequências de treliças.

De (3.38) e (3.36), esse período médio é:

$$T_{med} = (1 - \alpha_0)^{-1} = 1 + \sec\frac{2\pi}{m+2} \tag{3.47}$$

Portanto T_{med} é igual a 3 para $m = 4$, e é irracional para todo $m \geq 6$ par. Por exemplo, $T_{med} = (1 + \sqrt{2})$ para $m = 6$, significando que a sequência X_6 típica tem uma célula de não imitação (11 ou 00) a cada 2.4142 células em média. A irracionalidade de T_{med} para todo $m \geq 6$ nos leva a conjectura que um centro periódico de minimização de esfera assintótica não existe nesse caso. Não existe uma razão fundamental do porque que um centro de esfera minimizante deve sempre aproximar assintoticamente um padrão periódico, esse parece ser um caso em que isso não acontece.

Vamos concluir essa seção com conjecturas sobre o comportamento de centros maximizantes e minimizantes para o caso de $d \geq 4$ finito e n grande, baseados na interpolação dos casos $d = 2$ e $d = \delta n$ assintótico, como foi feito para m ímpar na seção anterior. Desde que alguns dos centros maximizantes da esfera-2 do lema 3 são também qualificados pelo teorema 4 como centros assintóticamente maximizantes, podemos esperar que eles também maximizem esferas de qualquer raio finito. O centro (1001100...) pode também ser esperado como minimizante de esferas de qualquer raio em X_4, já que ele o faz para $r = 2$ e para raios assintóticos. Para $m \geq 6$ par, esperamos que o padrão minimizante degenere gradualmente para um comportamento caótico a medida que d cresce, começando do padrão zigzag interno para $d = 2$, e sempre obedecendo ao teorema 6.

3.6 Assimetrias assintóticas em espaços de Hamming restritos em DC

Nessa seção, discutiremos a determinação da distância típica $\delta_t(\vec{x}, m)$ de $\vec{x}$ em X_m para qualquer $\vec{x}$ e m. A equação (3.34) resolve esse problema trivialmente para m par, então a discussão será concentrada no caso m ímpar. Vamos considerar o polinômio gerador de erros

$$P_m(z; \vec{x}) = \sum_{d=0}^{n} S_m(\vec{x}, d) z^d = \left[\prod_{i=1}^{n} A_i(z; x_i)\right]_{\frac{m+3}{2}, \frac{m+3}{2}}, \tag{3.48}$$

Onde $A_i(z; x_i) = \|a_{ijk}(z; x_i)\|$ é a matriz de geração de erros a partir do dígito x_i na posição i da treliça. O elemento $a_{ijk}(z; x_i)$ é zero quando a transição jk é proibida ou

inacessível na posição i da treliça; é 1 quando a transição jk é rotulada com o dígito x_i; e é z quando é rotulada com $(1 - x_i)$. Numeramos os estados com 1,2,...$m+1$, em ordem de SDC decrescente.

Existem somente quatro matrizes $A_i(z; x_i)$, correspondendo a i par ou ímpar, e x_i igual a 1 ou 0. Podemos reduzir esse número para 2 se usarmos matrizes reduzidas baseadas na classe equivalente de estados $\{1, m+1\}, \{2, m\}, \{3, m-1\}$, e assim por diante. Em uma paridade de posições os estados acessíveis são $1, 3, 5, ..., m$ enquanto na outra eles são $(m+1), (m-1), (m-3), ..., 2$. Usando um desses conjuntos (eles são equivalentes) como a ordem dos estados reduzidos, a matriz reduzida para x_i quando ele é o dígito de imitação na posição i, aqui definida como a matriz de imitação, é:

$$A_m(z) = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 1 & z \\ 0 & 0 & 0 & \ldots & 1 & z & 0 \\ & \cdot & & & & \cdot & \\ & \cdot & & & & \cdot & \\ & \cdot & & & & \cdot & \\ & \cdot & & & & \cdot & \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ 0 & 1 & z & \ldots & 0 & 0 & 0 \\ 1 & z & 0 & \ldots & 0 & 0 & 0 \end{pmatrix}, (\frac{m+1}{2}) \times (\frac{m+1}{2}) \tag{3.49}$$

Seja $\vec{y}$ a sequência de tudo imitação em X_m. De (3.48) e (3.49), teremos:

$$P_m(z; \vec{y}) = \{[A_m(z)]^n\}_{kk}, \tag{3.50}$$

onde k é a parte inteira de $(m+5)/4$. A teoria de Perron-Frobenius [[6] cap.1] irá então resultar:

$$P_m(z; \vec{y}) = [\lambda_A(z)]^{n+\Omega(n)}, \tag{3.51}$$

Onde $\lambda_A(z)$ é o maior autovalor da matriz de imitação $A_m(z)$. Façamos $\delta_i(m)$ denotar a distância típica da sequência tudo imitação em X_m. Do lema 1, segue que uma sequência escolhida aleatoriamente em X_m tem probabilidade 1 de ter distância relativa $\delta_i(m)$ da sequência de tudo imitação $\vec{y}$. Portanto, $n.\delta_i(m)$ é a distância média assintótica a partir do centro $\vec{y}$ em X_m. Nós temos então de (3.48) e (3.51):

$$\delta_i(m) = \frac{< d >}{n} = \frac{1}{n}\frac{\sum dS(\vec{y}, d)}{|X_m|} = \lim_{n\to\infty} \frac{1}{n}\frac{P'_m(1; \vec{y})}{P_m(1; \vec{y})} = \frac{\lambda'_A(1)}{\lambda_A(1)} \tag{3.52}$$

para qualquer m ímpar.

Vamos agora considerar a sequência $\vec{x}$ com αn dígitos de imitação e $(1-\alpha)n$ dígitos de não imitação em X_m, com m ímpar. A distância entre $\vec{x}$ e uma sequência aleatoriamente escolhida em X_m é uma variável aleatória com distribuição:

$$prob(d) = \frac{S_m(\vec{x}, d)}{|X_m|} = prob\left[\sum_{i=1}^{\alpha n} f_i + \sum_{i=1}^{(1-\alpha)n} g_i\right], \tag{3.53}$$

onde as variáveis f_i e g_i representão incrementos de distância durante dígitos de imitação e de não imitação do centro, respectivamente. Tomando as esperanças em ambos os lados resulta:

$$\frac{< d >}{n} = \frac{1}{n}\sum_{i=1}^{\alpha n} < f_i > + \frac{1}{n}\sum_{i=1}^{(1-\alpha)n} < g_i > \tag{3.54}$$

No limite assintótico , as variáveis f_i tornam-se identicamente distribuidas , as variáveis g_i também, e podemos escrever:

$$\delta_t(\vec{x}, m) = \alpha < f_i > + (1-\alpha) < g_i > \tag{3.55}$$

Identificando essa equação com (3.52) para $\alpha = 1$, temos $< f_i >= \lambda'_A(1)/\lambda_A(1)$. De (3.23) e lema 1, obtemos que a distância típica da palavra tudo não imitação em X_m é $[1 - \delta_i(m)]$, e identificando esse valor com (3.52) e (3.55) para $\alpha = 0$ resulta $< g_i >= 1 - \lambda'_A(1)/\lambda_A(1)$. A substituição em (3.55) resulta na seguinte expressão para a distância típica de qualquer palavra $\vec{x} \in X_m$ em X_m:

$$\delta_t(\vec{x}, m) = 1 - \alpha + (2\alpha - 1)\frac{\lambda'_A(1)}{\lambda_A(1)} \tag{3.56}$$

A equação (3.56) mostra que distâncias típicas em X_m variam no intervalo $[\delta_i(m), 1 - \delta_i(m)]$, onde $\delta_i(m)$ é dado por (3.52) para m ímpar, e é igual a 1/2 para m par. É então claro que VSDC ímpares induzem uma assimetria no espaço de Hamming, caracterizada pelo desvio das distâncias relativas típicas de sequências restritas no espaço restrito, quando comparado com o valor universal 1/2 do espaço irrestrito.

Cálculos simples a partir de (3.52) resultam em $\delta_i(3) = \sqrt{5}/5 \simeq 0.44721$, $\delta_i(5) \simeq 0.483435$, e $\delta_i(7) \simeq 0.49264$. A assimetria portanto desaparece rapidamente quando m aumenta. O desvio máximo da distância típica irrestrita 1/2 ocorre sempre para sequências (...0101010..), e é no máximo $(1 - 2\sqrt{5}/5)100 \simeq 10.56\%$, em X_3. A figura 3.5 mostra a distribuição de distâncias a partir dessas duas sequências mais "atípicas" em X_3, assim como a partir da sequência centro discutida no exemplo 1, para a qual $\alpha = 1/3$.

Agora se considerarmos a distância típica a palavra tudo não imitação em X_m veremos que ela representa com probabilidade um a taxa de imitação de uma sequência escolhida ao acaso em X_m. Chamaremos a essa taxa de imitação de taxa típica de imitação em X_m $\alpha_t = 1 - \delta_i(m)$. Podemos então calcular a distância típica entre duas palavras escolhidas ao acaso em X_m denominada $\delta_{tt}(m)$. Aplicando $\alpha_t(m)$ a (3.56) resulta

$$\delta_{tt}(m) = 2\delta_i(m)(1 - \delta_i(m))$$

Para o caso $m = 3$ observamos uma distância típica entre palavras escolhidas ao acaso $\delta_{tt}(3) \simeq 0.496$.

3.7 Conclusões

Nesse capítulo destacamos como principais contribuições:

1. Identificação dos centros extremos de esferas no espaço m-DC restrito para os casos de raios 2 e raios assintoticamente grandes.
2. A partir dos centros anteriormente citados são calculados limitantes assintóticos superiores e inferiores na taxa do código restrito × distância mínima relativa.
3. Determinação de distâncias típicas em espaços m-DC restritos.

3.8 Apêndice A

$$S_3(\vec{x},d) = \sum_{M=1}^{d/2} \frac{(d/2-1)!}{(m-1)!(d/2-M-2)!} \frac{(n/3-d/2)!}{M!(n/3-d/2-M)!}(4/3)^M 3^{d/2}$$

Usamos a fórmula de Stirling para k grande:

$$k! \rightarrow k^k \exp(-k)\sqrt{2\pi k}$$

Obtemos

$$S_3(\vec{x},d) \rightarrow \sum_{M=1}^{d/2} \frac{(d/2-1)^{(d/2-1)}}{(M-1)^{(M-1)}(d/2-M-2)^{(d/2-M-2)}} \frac{(n/3-d/2)^{(n/3-d/2)}}{M^m(n/3-d/2-M)^{(n/3-d/2-M)}}(4/3)^M 3^{d/2}$$

fazendo $M = \mu n$ e $d = \delta n$

$$= \sum_{\mu=1/n}^{\delta/2} \frac{\delta/2-1/n)^{(\delta/2-1/n)n}}{(\mu-1/n)^{(\mu-1/n)n}(\delta/2-/mu-2)^{(\delta/2-\mu-2)n}} \frac{(1/3-\delta/2)^{(1/3-/delta/2)n}}{\mu^{\mu n}(1/3-\delta/2-mu)^{(1/3-\delta/2-\mu)n}}(4/3)^{\mu n} 3^{\delta n/2}$$

$$\dot{S}_3(\vec{x},\delta) = \lim_{n\rightarrow\infty} \frac{1}{n} \log_2 S_3(\vec{x},\delta n) =$$

Como temos uma soma de termos exponenciais, no limite acima haverá domínio do termo dado por:

$$\max_{0<\mu<\delta/2} [\delta/2 \log_2 \delta/2 + (1/3-\delta/2)\log_2(1/3-\delta/2) - 2\mu \log_2 \mu - (\delta/2-\mu-2)$$
$$\log_2(\delta/2-\mu-2) - (1/3-\delta/2-\mu)\log_2(1/3-\delta/2-\mu) + \mu \log_2 4/3 + \delta/2 \log_2 3]$$

Bibliografia

[1] Chien,T.M: "Upper bound on the efficiency of DC-constrained codes", BSTJ, pp.2267-2287, Nov. 1970.

[2] Ferreira,H.C:"Lower bounds on minimal Hamming distance achievable with runlenght constrained or DC-free block codes and the synthesis of (16,8) $d_{min} = 4$ DC-free block code", IEEE Trans. Magn., vol. MAG-20, pp. 881-883, 1984.

[3] Gilbert,E.N: "A comparison of signalling alphabets", BSTJ, vol. 31, pp. 504-522, 1952.

[4] Varshamov,R.R: " Estimate of the number of signals in error-correcting codes", Dokl.Akad.Nauk SSSR, vol. 117, pp.739-741, 1957.

[5] Bellman,R.: "Introduction to Matrix Analysis". McGraw-Hill, 1960.

[6] Seneta,E: "Non negative matrices and Markov Chains ". New York, Springer Verlag, 1981.

Capítulo 4

ALGUNS MÉTODOS CONSTRUTIVOS PARA CLCEs

4.1 Introdução

No capítulo 2 obtivemos alguns CLCEs via busca computacional. Porém o uso prático daqueles códigos é bastante limitado, pois um pequeno aumento na taxa do CLCE se dá as custas de um aumento exponencial no tamanho do dicionário de codificação / decodificação. Por exemplo, o CLCE(22,13,4) com VSDC=4 necessita de memória de codificação 8K x 22 e memória de decodificação de 4M x 13. Além disso encontramos dificuldades tecnológicas para implementar memórias grandes em sistemas de alta velocidade.

Nesse capítulo abordamos algumas técnicas que tentam conferir maior estruturação ao CLCE e com isso resolver os problemas acima mencionados. A primeira idéia envolvida na estruturação algébrica é o uso de palavras de códigos corretores de erro lineares (CCEL) para formar o CLCE. Por isso, discutiremos primeiro o compromisso linearidade x balanceamento, para depois passar às técnicas construtivas propriamente ditas.

4.2 Linearidade × Balanceamento

Abordaremos o compromisso linearidade x balanceamento através do questionamento da eficácia dos diversos caminhos que levam ao balanceamento de um CCEL.

4.2.1 Primeira construção

Um primeiro caminho que imaginamos seguir é: Tentar achar um CCEL $= \{\vec{c}_0, \vec{c}_1, .., \vec{c}_m\}$ cujas palavras $\vec{c}_i$ são balanceadas a menos da palavra tudo zero, $\vec{c}_0 = 0^{(n)}$:

$$\text{Linearidade} + \text{Balanceamento} \Rightarrow W(\vec{c}_i) = n/2 = d_{min}, \forall \vec{c}_i \neq \vec{c}_0 \text{ , usando o}$$

"Singleton Bound" [1] segue $n/2 = d_{min} \leq n - k + 1$ ou $r \leq 1/2 + 1/n$. Concluimos que:

- É necessária uma grande expansão de faixa para esse procedimento.
- A taxa do código tenderá a se anular para n grande segundo o limitante superior de Plotkin esquematizado na figura 4.1 [2].

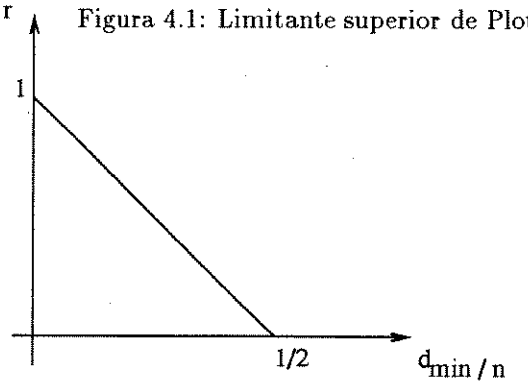


Figura 4.1: Limitante superior de Plotkin

4.2.2 Segunda construção

Um segundo caminho que podemos tentar: construir um código de bloco balanceado como um dos "cosets"[1] de um CCEL. Entende-se por coset de um CCEL um código formado por palavras resultantes da soma módulo 2 dos elementos (palavras) do CCEL com uma determinada palavra de translado $\vec{T}$, que não pertença ao CCEL. Abaixo construimos o arranjo padrão para o CCEL [1]:

$\vec{c}_0$	$\vec{c}_1$	$\vec{c}_2$	. . . .	$\vec{c}_{2^k-1}$
$\vec{e}_1$	$\vec{c}_1 \oplus \vec{e}_1$	$\vec{c}_2 \oplus \vec{e}_1$	. . . .	$\vec{c}_{2^k-1} \oplus \vec{e}_1$
$\vec{e}_2$	$\vec{c}_1 \oplus \vec{e}_2$	$\vec{c}_2 \oplus \vec{e}_2$	. . . .	$\vec{c}_{2^k-1} \oplus \vec{e}_2$
.	.	.	.	.
.	.	.	.	.
$\vec{e}_{2^{n-k}-1}$	...	...	. . . .	$\vec{c}_{2^k-1} \oplus \vec{e}_{2^{n-k}-1}$

Repare que o arranjo padrão divide o espaço X^n em todos os possíveis "cosets" do CCEL ($\vec{T} = \vec{e}_i$) . Assim, para que um dos cosets seja balanceado, o lider $\vec{e}_i$

[1] Coset pode ser traduzido como classe lateral

deve ser também balanceado, ou seja, $\vec{e}_i$ representa erro de $n/2$ posições que é corrigível. Supondo d_{min} par, a capacidade de correção é $t = (d_{min} - 2)/2$ portanto $n/2 = W(\vec{e}_i) \leq t + 1 \quad \Longrightarrow \quad n/2 \leq (d_{min} - 2)/2 + 1$ ou $d_{min} = n$.
Concluimos que esse procedimento só é possível, no caso d_{min}, par para os códigos de repetição, ou seja , não é aplicável. No caso d_{min} ímpar os argumentos também nos levam a conclusão de existência de códigos com número de bits de informação inferiores ou iguais a dois.

4.2.3 Terceira construção

Outro procedimento simples é tentar construir um código balanceado não linear apartir do entrelaçamento da palavra $\vec{c}_i \in$ CCEL, com a sua inversa $\bar{\vec{c}}_i$. Por exemplo:

$$\vec{c}_i = 00000 \Rightarrow \vec{c}_i^{'} = 0101010101$$

Repare que:

- Esse método nos leva a um CLCE(2n,k,2d) partindo de um CCEL(n,k,d).
- O ganho assintótico suave do CCEL para canais AWGN - PSK (kd_{min}/n) é mantido no CLCE.
- O codificador pode ser implementado pela concatenação do codificador CCEL e um codificador de linha do tipo Manchester [16]. Porém o decodificador deve atuar no bloco como um todo , o que pode ser feito com a mesma complexidade do decodificador do CCEL original.
- Continuamos a trabalhar com uma grande expansão de faixa, ou equivalentemente taxa baixa ($r \leq 1/2$).

4.2.4 Quarta construção

Podemos construir um CLCE usando as palavras quase balanceadas de um CCEL para formar os alfabetos de um CL. Essas palavras quase balanceadas são as que têm peso em torno de $n/2$ num intervalo aberto de mais ou menos ε. Avaliemos a eficiência desse procedimento através da estimativa do número de palavras quase balanceadas de um CCEL. Das identidades de Mac'Williams [15] podemos calcular os primeiros momentos da distribuição de pesos (A_i) do CCEL cujo código dual tem distância mínima $d^{'}$:

Soma $\sum_{i=0}^{n} A_i = 2^k$
Momento 1 $\sum_{i=0}^{n} iA_i = 2^k n/2$
Momento 2 $\sum_{i=0}^{n}(i - n/2)^2 A_i = 2^k n/4$ para $d^{'} > 2$

Usando a desigualdade de Chebychev [8] para a distribuição $P(i) = A_i/2^k$ dada por

$$P(|I - m_I| \geq \varepsilon) \leq \sigma_I^2/\varepsilon^2$$

onde,

m_I = média da distribuição

σ_I^2 = variância da distribuição

segue

$$\sum_{|i-n/2|<\varepsilon} A_i \geq 2^k(1-\frac{n}{4\varepsilon^2})$$

para $d' > 2$.

Com $r = k/n > 0$ e n grande, os pesos das palavras do CCEL tendem rapidamente a se concentrar em torno de n/2, conforme visto no capítulo 3 quando falamos de distância típica.

Para melhorar a eficiência do método, podemos trabalhar convenientemente com ε para obter balanceamento completando os pesos de palavras "quase-balanceadas"ou usando mais de um estado tesrminal de SDC. Por exemplo:

I- $\varepsilon = 2 \Rightarrow$ uso palavras de peso $n/2-1$, $n/2$ e $n/2+1$. Podemos colocar sufixos nessas palavras tornando-as balanceadas através da regra:

$\vec{c}_i \in CCEL$, $W(\vec{c}_i) = n/2 \Rightarrow$ completo com sufixo 10.

, $W(\vec{c}_i) = n/2-1 \Rightarrow$ completo com sufixo 11.

, $W(\vec{c}_i) = n/2+1 \Rightarrow$ completo com sufixo 00.

II- $\varepsilon = 5 \Rightarrow$ consideremos o código de Golay estendido[18] $(24, 2^{12}, 8)$. A sua distribuição de pesos é:

w	0	8	12	16	24
A_w	1	759	2576	759	1

Tomando as palavras balanceadas exceto $1^{(12)}0^{(12)}$ e seu complemento temos o CLCE $(24, 2574, 8)$ e VSDC=16. Se considerarmos dois estados terminais de SDC(+4 e -4) e portanto dois alfabetos: o alfabeto progressivo formado pelas palavras de peso 12 e 16 , e o alfabeto regressivo formado pelas palavras de peso 12 e 8. Assim obtemos o CLCE bialfabético (24,3334,8) e VSDC=24. Se permitirmos o uso da palavra $1^{(12)}0^{(12)}$ e seu complemento temos o CLCE bialfabético (24,3336,8) e VSDC=32.

Embora esse método de construção de CLCEs atinja assintóticamente a taxa do CCEL original ($r \rightarrow \frac{k-1}{n}$), notamos que somente a restrição de balanceamento não evita a ocorrência indesejável de VSDC grande. Por isso veremos a seguir novos métodos onde correlatamos a sequência de palavras de um CCEL , de modo a obter a VSDC na linha razoavelmente restrita.

4.3 Códigos "coset" balanceados

4.3.1 Introdução

Esse método foi introduzido por Deng,H.R. & Herro,M.A. [13] e se aplica bem aos canais de transmissão de alta velocidade, onde uma baixa complexidade de codifi-

cação/decodificação é exigida.

A idéia fundamental envolvida é a "partição do subespaço vetorial" gerado por um CCEL(n,k+J) em 2^k subconjuntos $\{B_0, B_1, ..., B_{2^k-1}\}$ de 2^J palavras (elementos) cada. A cada k_upla de informação associamos um subconjunto B_i, porém escolhemos apenas um elemento para transmitir. O critério da escolha dessa palavra é baseado no valor do estado de SDC (soma digital corrida) ao final da transmissão do bloco anterior e é feita de modo a confinar os valores de SDC num intervalo de variação finito. Vide figura 4.2 :

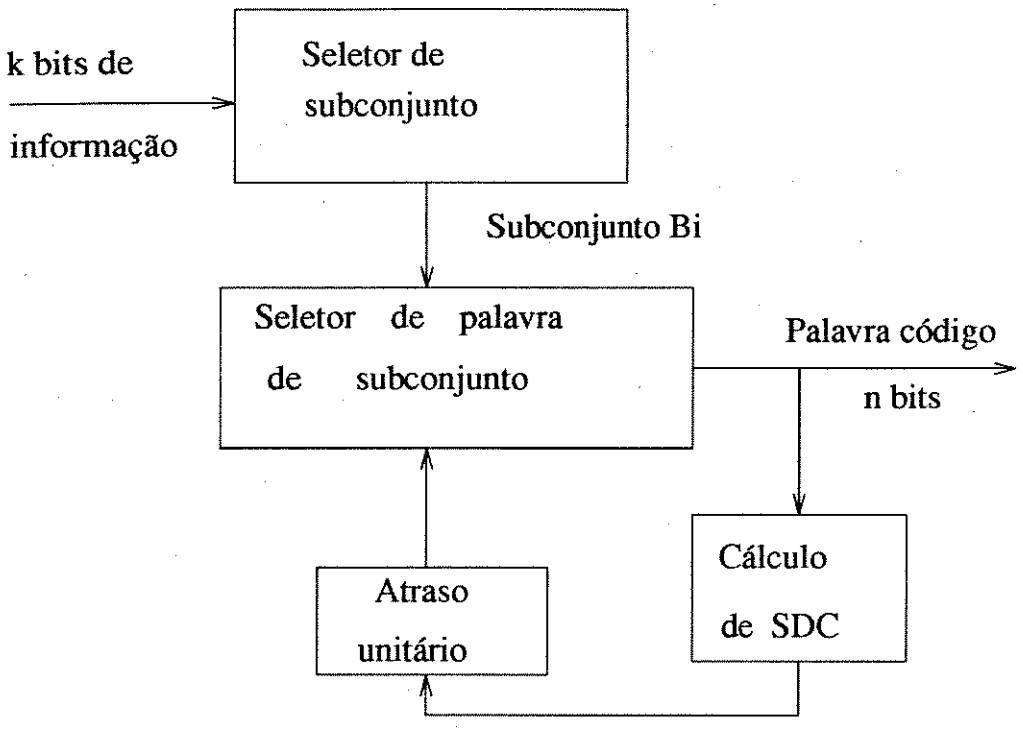


Figura 4.2: Diagrama esquemático do codificador "coset" balanceado

Repare que apenas palavras código do CCEL (chamado de CCEL pai) são usadas, assim o código "coset" balanceado preserva a capacidade de correção do CCEL pai e as estruturas de codificação/decodificação conhecidas para códigos de bloco lineares podem ser usadas mediante pequenas alterações.

A propriedade de linearidade permite-nos realizar a partição do CCEL pai, através da construção do arranjo dos 2^{k+J} elementos do CCEL em 2^J "cosets", conforme o esquema abaixo [2]:

[2] Esse arranjo é diferente do arranjo padrão usado em decodificação de CCELs [1] porém procuramos manter a nomenclatura usual

$\vec{c}_0$	$\vec{c}_1$	. . .	. . .	$\vec{c}_{2^k-1}$
$\vec{e}_1$	$\vec{c}_1 \oplus \vec{e}_1$	. . .	. . .	$\vec{c}_{2^k-1} \oplus \vec{e}_1$
$\vec{e}_2$	$\vec{c}_1 \oplus \vec{e}_2$	. . .	. . .	$\vec{c}_{2^k-1} \oplus \vec{e}_2$
.	.	.	.	.
.	.	.	.	.
$\vec{e}_{2^J-1}$	$\vec{c}_1 \oplus \vec{e}_{2^J-1}$	. . .	. . .	$\vec{c}_{2^k-1} \oplus \vec{e}_{2^J-1}$

Às linhas do arranjo chamaremos "cosets" ; às colunas chamaremos $B_0, B_1, ..., B_{2^k-1}$ e aos $\vec{e}_i$s , i=0,...,$2^J - 1$, chamaremos líderes de "coset".

Repare que ao escolhermos os líderes de "coset" como elementos formadores de um subespaço de CCEL, podemos representá-los como combinação dos J primeiros vetores linha ($\vec{g}_l$) da matriz geradora do CCEL (G). Esse procedimento permite representar qualquer palavra $\vec{v} \in$ CCEL como:

$$\vec{v} = \left\{(0^{(J)}, u^{(k)}) * G\right\} \oplus \sum_{l=1}^{J} \alpha_l \vec{g}_l, \quad \alpha_l = 0, 1. \tag{4.1}$$

$0^{(J)}$ é a J_upla de elementos nulos e $u^{(k)}$ é a k_upla de informação na entrada do codificador.

Nota-se que o primeiro termo a direita da equação 4.1 corresponde à escolha do subconjunto B_i enquanto que o segundo termo a direita seleciona um dos elementos de B_i para transmitir, ou seja, é o equacionamento das operações dos blocos do codificador (fig. 4.2).

Um dos objetivos que devemos levar em conta ao escolhermos os líderes de coset é maximizar a capacidade do termo a direita da equação (4.1) de confinar estados de SDC na saída do codificador. Por isso o autor [13] propõe uso de CCEL que apresentem palavras do tipo :

$$\vec{g}_1 = (1^{(w_1)}, 0^{(w_2)}, 0^{(w_3)}, ..., 0^{(w_J)})$$
$$\vec{g}_2 = (0^{(w_1)}, 1^{(w_2)}, 0^{(w_3)}, ..., 0^{(w_J)})$$
$$.....$$
$$\vec{g}_J = (0^{(w_1)}, 0^{(w_2)}, ..., 0^{(w_{J-1})}, 1^{(w_J)})$$

com $\sum_i w_i = n$.

Os autores [13] provam que CCELs pais escolhidos dentre a família dos BCH binários apresentam palavras da forma requerida, bastando escolher:

$n = 2^b - 1 = m.J$, m e J ímpares.

$m \geq 2t + 1$

$w_i = m$

Um exemplo simples e bastante usado é trabalhar com CCELs que possuam a palavra tudo 1 ($1^{(n)}$), também chamados de CCELs transparentes. Assim, J=1 e $\vec{g}_1 = 1^{(n)}$.

UNICAMP BIBLIOTECA

4.3.2 Algorítmo de codificação

Seja D'_t o estado de SDC ao final da palavra código transmitida no instante t. Seja d_l a disparidade do l_ésimo segmento $\vec{v}_l$ da palavra código $\vec{V}$ (que pode ser representada como $\vec{V} = (\vec{v}_1, \vec{v}_2, .., \vec{v}_J)$ e calculada conforme $\vec{V} = (0^{(J)}, u^{(k)}) * G$. Se $\vec{g}_l$ é adicionado a $\vec{V}$, a polaridade de d_l na nova palavra código mudará; assim podemos escrever o algorítmo para confinar SDC:

1. $\vec{V} \leftarrow (0^{(J)}, u^{(k)}) * G, \quad l \leftarrow 1, \quad D'_t \leftarrow D'_{t-1}$
2. Se $D'_t.d_l \leq 0$

   então $D'_t \leftarrow D'_t + d_l$, $l \leftarrow l + 1$, vá para passo 4.
   senão vá para passo 3.
3. $\vec{V} \leftarrow \vec{V} + \vec{g}_l$, $D'_t \leftarrow D'_t - d_l$, $l \leftarrow l + 1$, vá para 4
4. Se $l \leq J$ vá para 2 , senão codifique o próximo bloco de informação.

4.3.3 Algorítmo de decodificação

Seja $\vec{R}$ a versão detetada no receptor para a palavra transmitida :

1. Decodifique $\vec{R}$ com base em esquemas tradicionais de decodificação para CCEL, obtendo $(\hat{a}_1, \hat{a}_2, ..., \hat{a}_J, \hat{u}^{(k)})$.
2. Retire a informação útil decodificada , ou seja, os k últimos bits da palavra decodificada anteriormente.
3. decodifique a próxima palavra recebida.

4.3.4 Exemplos

Exemplo 1

Trabalhemos com todo o espaço X^n; podemos considera-lo como um CCEL trivial com $d_{min} = 1$,ou seja, sem capacidade de corrreção. Usando $\vec{g}_1 = 1^{(n)}$ resulta o seguinte algorítmo de codificação:

1. $\vec{V} \leftarrow (0^{(1)}, u^{(k)})$
2. $D'_t \leftarrow D'_{t-1}$
3. Se $D'_t.d_t \leq 0$

   então transmito $\vec{V}$, senão transmito $\bar{\vec{V}}$

Repare que esse exemplo nos leva a códigos apenas de linha (CLs) que são implementáveis sem auxílio de tabelas e por isso se prestam a uso em linhas de transmissão de alta velocidade. De fato são bastante semelhantes aos CLs PF-mB(m+1)B propostos em [3].

Exemplo 2

Trabalhemos com o CCEL pai Hamming (7,4,3) com J=2 [3], gerado por:

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

Queremos codificar $\vec{u}$=(0 0) no instante t, supondo $D'_{t-1} = 1$:

- $\vec{V}$=(0000)*G=(0000000), $l \leftarrow 1$, $D'_t \leftarrow 1$;
- $d_l = d_1 = -4$, $D'_t.d_t = -4 < 0$, $D'_t \leftarrow 1 - 4 = -3$, $l \leftarrow l + 1 = 2$;
- $d_l = d_2 = -3$, $D'_t.d_2 = 9 > 0$;
- $\vec{V}$=(0000000)+$\vec{g}_2$= (0000111), $l \leftarrow l + 1 = 3$;
- $l > J$ então já temos a versão final codificada de $\vec{v}$

Esse CLCE apresenta os seguintes parâmetros:

– Capacidade de correção de 1 erro.

– VSDC= 6.

– Taxa efetiva $r = 2/7 = 0.286$.

O CLCE pode ser interpretado na forma clássica de códigos de linha , em três alfabetos de codificação (e três estados terminais de SDC) conforme expõe a tabela a seguir:(o que significa que nem todas as palavras do CCEL são usadas)

Palavra fonte	SDC=-1	SDC=0	SDC=1
00	1111000	0000111	0000111
01	0110101	0110010	0110010
10	1010011	1010011	1010100
11	1100110	1100110	1100001

O espectro de potência é ilustrado na figura 4.3:

Repare que o CLCE exemplificado nos oferece bom confinamento de SDC (VSDC=6) as custas de uma taxa efetiva de código excessivamente baixa (r=0.28). O'Reilly,J.J. & Popplewel,O. [5] trabalham em técnicas complementares que apresentamos a seguir para melhorar a performance dos códigos "Coset balanceados" com J=1.

[3] Exemplo retirado de [4]

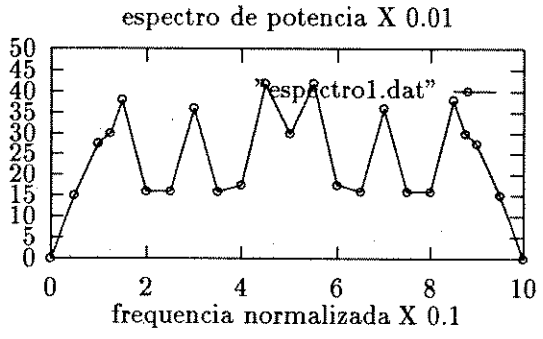


Figura 4.3: espectro de potência do CLCE coset balanceado do exemplo 2

4.3.5 Melhoria no balanceamento para essa classe de códigos com J=1.

I

Se ao invés de trabalharmos com o CCEL pai diretamente, usarmos a sua versão transladada pela soma com uma palavra externa ao mesmo (chamada de vetor modificador), podemos eliminar a presença das palavras código $0^{(n)}$ e $1^{(n)}$ do dicionário original. Agora as palavras do CLCE não são as mesmas que as do CCEL pai, porém a distância mínima e consequentemente a capacidade de correção de erros permanecem inalteradas.

II

Usando um CCEL(n,k,d) sistemático transparente é possível formar um CLCE$(2n, 2k-1, d)$ que mescle codificação algébrica com codificação por consulta a dicionário para substituir palavras de disparidade elevada:

Algorítmo de codificação

1. Seja $\vec{u}^{(2k-1)}$ a palavra fonte a ser codificada.
2. Divida $\vec{u}$ em duas partes $\vec{u}_2^{(k-1)}$ e $\vec{u}_1^{(k)}$, formando as duas palavras fonte auxiliares $\vec{U}_2' = (0, \vec{u}_2)$ e $\vec{U}_1' = (\vec{u}_1)$
3. Se $\vec{U}_1'$ e/ou $\vec{U}_2'$ é $0^{(k)}$, ou $\vec{U}_1'$ é $1^{(k)}$ então codifique usando o dicionário obtendo $\vec{V}''$. Senão :
   - Codifique $\vec{V}_1' = \vec{U}_1' * G$, $\vec{V}_2' = \vec{U}_2' * G$, e $\vec{V}' = (\vec{V}_2', \vec{V}_1')$
   - Seja D_{t-1}' o estado de SDC ao final da palavra de linha anterior; aplique a seguinte regra para obter $\vec{V}''$

     $D_{t-1}' = 0$:

     disparidade$(\vec{V}') \leq 0$, $\vec{V}'' = \vec{V}'$

     disparidade$(\vec{V}') > 0$, $\vec{V}'' = \vec{\bar{V}}'$

$D'_{t-1} \neq 0$:
disparidade$(\vec{V}').D'_{t-1} \leq 0$, $\vec{V}'' = \vec{V}'$
disparidade $(\vec{V}').D'_{t-1} > 0$, $\vec{V}'' = \bar{\vec{V}}'$

4. transmitir $\vec{V}''$

Note que a concatenação de palavras de um CCEL (transparente) duas a duas forma um conjunto normalmente rico em elementos balanceados, isso poderá suprir a demanda requerida pelo dicionário se a seguinte condição for satisfeita:

Número de palavras balanceadas não produzidas pelo algoritmo de codificação $\geq$ número de palavras balanceadas requeridas pelo dicionário.

Assim resulta o seguinte critério que o CCEL pai deve satisfazer [5]:

$$n_0^2/2 + \sum_{i=1}^{n} n_i^2 \geq 2^{k+1} - 1$$

onde n_i = número de palavras do CCEL com disparidade i.

Algorítmo de decodificação

1. Dividir o vetor recebido $\vec{R}^{(2n)}$ em $\vec{R}_1^{(n)}$ e $\vec{R}_2^{(n)}$.
2. Corrigir erros de $\vec{R}_1$ e $\vec{R}_2$ pelos métodos tradicionais de CCELs obtendo $\hat{\vec{V}}_2''$ e $\hat{\vec{V}}_1''$.
3. Se o n_ésimo bit de $\hat{\vec{V}}_2''$ for um e a disparidade do par $(\hat{\vec{V}}_2'', \hat{\vec{V}}_1'')$ zero, decodifique pelo dicionário para obter a palavra de informação estimada $\hat{\vec{U}}$.
   Senão:
   Decodifique pelo CCEL os vetores $\hat{\vec{V}}_2''$ e $\underline{\hat{\vec{V}}_1'' \text{ para}}$ $\hat{\vec{U}}_2''$ e $\hat{\vec{U}}_1''$.
   Se o k_ésimo bit de $\hat{\vec{U}}_2''$ for 1 então $\hat{\vec{U}}' = (\hat{\vec{U}}_2'', \hat{\vec{U}}_1'')$.
   Senão $\hat{\vec{U}}' = (\hat{\vec{U}}_2'', \hat{\vec{U}}_1'')$.
4. Finalmente ler a informação "útil" $\hat{\vec{U}}$ em $\hat{\vec{U}}' = (0, \hat{\vec{U}})$

4.3.6 Tabela de alguns CLCEs coset balanceados e seus parâmetros

CCEL pai	dicionário ?	modificador(octal)	Freq. de corte	VSDC $\leq$	taxa
Hamming(7,4,3)	não	254	.0237	12	.43
Hamming(7,4,3)	sim	0	.0345	8	.5
BCH(15,11,3)	não	25252	.0084	36	.43
BCH(15,7,5)	sim	0	.0054	34	.43

4.3.7 Conclusões

Os CLCEs coset balanceados são bastante eficientes e de complexidade de implementação semelhante a dos CCELs. Porém deixam a desejar quando queremos trabalhar com comprimentos de bloco relativamente pequenos, alta taxa efetiva e VSDC baixo. Isso nos levou a sugerir a classe de CLCE fortemente confinada em VSDC descrita a seguir.

4.4 Códigos de bloco "fortemente" restritos em VSDC com capacidade de correção de 1 erro

4.4.1 Introdução

Esse novo método construtivo se propõe a extender a distância mínima "natural" de 2, das palavras balanceadas de um CL monoalfabético restrito em VSDC (o qual chamaremos CL base), para 4. Com isso passamos a ter a capacidade de correção de 1 erro (hard).

Veremos a seguir que o método permite trabalhar com taxa de CLCE tão próxima quanto quisermos da taxa do CL base, assim nos aproximamos dos melhores códigos baseados apenas em dicionários (capítulo 2) com a vantagem da redução de complexidade na codificação/decodificação.

4.4.2 Conceitos envolvidos

Para conseguirmos o aumento em d_{min} no CL base nos utilizamos de conceitos de "Mapeamento por partição de conjuntos" [6] e "Modulação codificada de bloco" [7]:

- Mapeamento por partição de conjuntos – Advém da partição sucessiva das palavras do CL base, as quais chamaremos pontos da constelação n_dimensional de trabalho. Esses níveis de partição devem gerar subconjuntos cujas distâncias mínimas de Hamming são crescentes. Formamos assim uma árvore cujos nós representam os subconjuntos e cujos braços (valem 0 ou 1) indicam o caminho do mapeamento dos pontos da constelação em relação à partição . Tomemos como exemplo a constelação 8_dimensional gerada pela restrição VSDC=4. Temos

então $|CL| = 54$ pontos os quais particionamos sucessivamente com objetivo de encontrar subconjuntos de $d_{min} \geq 4$, conforme ilustrado na figura 4.4.

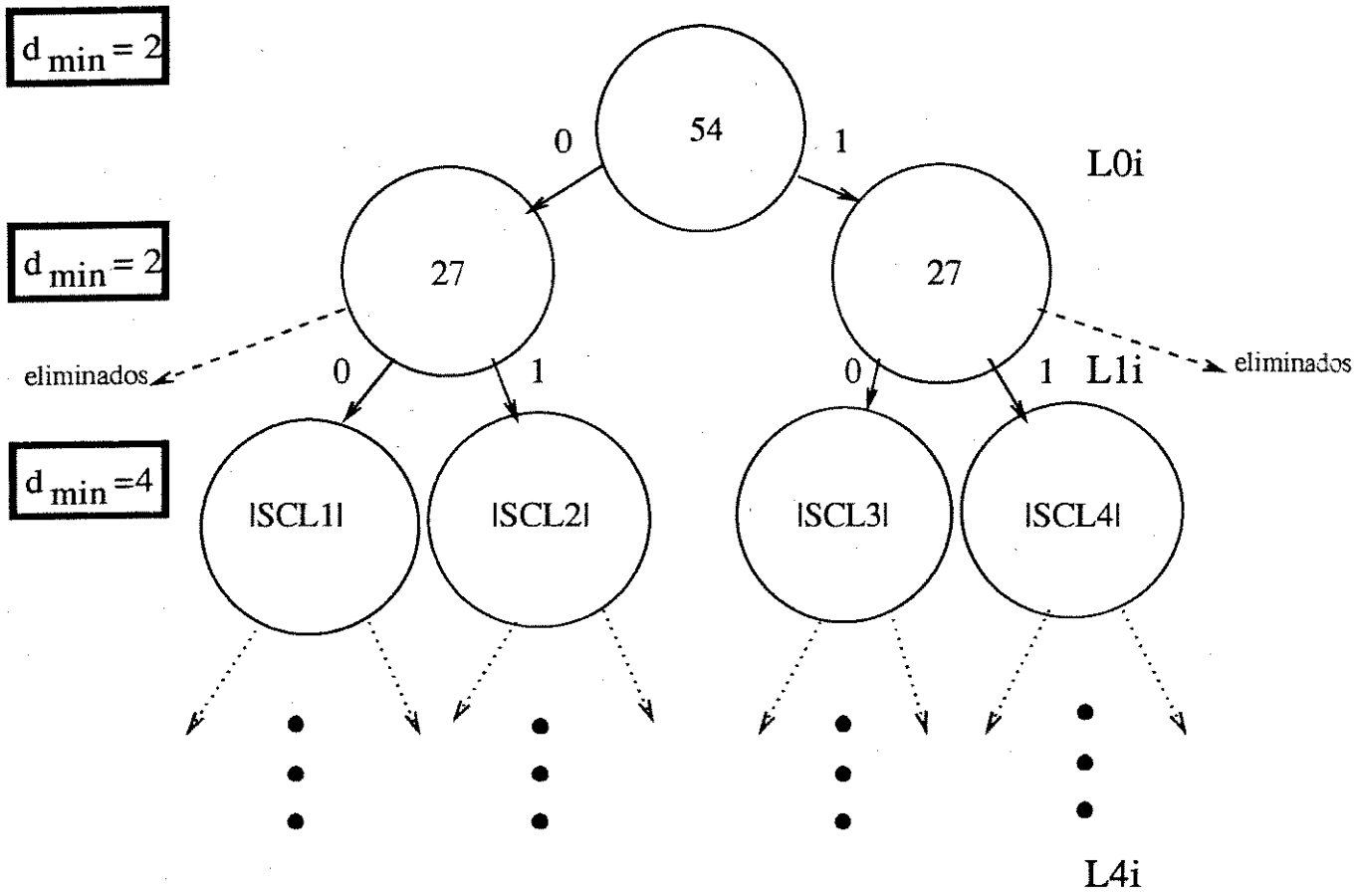


Figura 4.4: Partição da constelação binária 8_dimensional, VSDC=4

Note que de acordo com os resultados de busca computacional do capítulo 2, sabemos que o melhor CLCE para n=8 , VSDC=4 e $d_{min} = 4$ tem 12 palavras código; isso nos leva a especular sobre a existência da partição ilustrada acima [4] com $|SCL_i| \geq 8$. De fato, por simples inspeção nos melhores CLCEs (capítulo 2) selecionamos 4 códigos disjuntos SCL_1 , SCL_2 , SCL_3 e SCL_4 com $|SCL_1| = |SCL_2| = |SCL_3| = 10$ e $|SCL_4| = 12$. Chamaremos esses subconjuntos de "subcódigos de linha" (SCL) [5]. Ao mapeamento feito através de palavras binárias com k bits denominaremos "rótulos": $\ell(\vec{c}_i) = \vec{L}_i = (L_{0i}, L_{1i}, L_{2i}, ...L_{k-1i})$, $\vec{c}_i \in CL$.

- O uso da modulação codificada de bloco consiste na montagem da "matriz de palavras código" concebida em [7] e devidamente encaixada no nosso problema. Essa matriz tem número de linhas igual ao número de níveis da partição

[4] Repare que nesse nível de partição vários pontos da constelação são descartados

[5] A designação de subcódigos é ligeiramente conflitante com a usada por Forney[14] definida para grupos e subgrupos

; as linhas correspondentes ao nível de partição de $d_{min} = 2$ são compostas de palavras código de um CCEL de paridade simples($d_{min} = 2$) enquanto que as linhas correspondentes a níveis de partição de $d_{min} \geq 4$ não são codificadas ($d_{min} = 1$) e as colunas são rótulos para a constelação em questão. Dessa maneira a distância mínima entre os vetores formados pelo conjunto de pontos mapeados pela matriz apresentam $d_{min} = 4$. Para o exemplo dado formamos a seguinte "matriz de palavras código" onde o uso de 4 colunas significa juntar 4 blocos do CL base para formar um novo bloco de CLCE com capacidade de correção de 1 erro:

I	I	I	P
I	I	I	P
I	I	I	I
I	I	I	I
I	I	I	I

onde P = bit paridade e I = bit de informação.

A seguir, apresentaremos com mais detalhes os passos que compõem os processos de codificação/decodificação. Assim ficarão mais claros os conceitos apresentados anteriormente.

4.4.3 Transmissão/Codificação

- Preenchemos a matriz de palavras código com bits de informação.
- calculamos os bits de paridade.
- Transmitimos coluna a coluna os pontos da constelação (palavras do CL base) correspondentes ao mapeamento inverso $\vec{C}(\vec{L}_i)$ onde $\vec{L}_i$ é a i_ésima coluna da matriz de palavras código.
- Ao bloco final formado pela transmissão de todas as colunas da matriz denominaremos vetor CLCE ($\vec{V}$).

Observações:

- No exemplo em questão existe o caso de chegarmos a mais de um ponto da constelação partindo de um mesmo rótulo. Isso possibilita a introdução de "canais de serviço estatísticos" via código, pois a escolha de qual palavra iremos transmitir pode ser associada a outro bit de informação.

- A taxa efetiva do código do exemplo é dada por (número de bits de informação da matriz)/(número total de bits do vetor CLCE) = $18/32 = 0.5625$.
- O número de canais de serviço estatísticos do código do exemplo é dado por: (Número de pontos na constelação que efetivamente podem ser usados) $-2^k = 42 - 2^5 = 10$ canais. Cada qual com taxa média de $\frac{1}{32\times 8}$ × Taxa de linha (supondo dígitos de informação estatisticamente independentes e igualmente distribuidos) .

4.4.4 Recepção/decodificação(Hard)

- Sincronizamos o "quadro" de recepção reconhecendo o começo de cada vetor CLCE recebido. Isso pode ser feito mediante realimentação de informação de deteção de erros da saída do decodificador.
- Verificamos se cada uma das palavras (4 no exemplo em questão) que compõem o vetor CLCE apresenta violação em relação ao CL base, ou seja, pertencem ou não ao CL base (no exemplo dado temos 42 palavras possíveis que pertencem ao CL base). No modo tradicional de decodificação por consulta a dicionário usa-se uma memória de decodificação para fazer essa operação.
- Paralelamente a verificação, fazemos o mapeamento das palavras CL para " rótulos decodificados " através de consulta a memória de decodificação.
- Se ocorrerem mais de uma violação dentro do vetor CLCE, não será possível a correção Hard. Logo indicamos a deteção desses erros no vetor final decodificado.
- Se não ocorreram violações, os bits dos rótulos decodificados (saída da memória de decodificação) , correspondentes aos bits de informação na " matriz de palavras código " , dão o vetor final decodificado.
- Se ocorreu uma violação, calcula-se o caminho dentro da árvore de partições que nos leva ao subcódigo de $d_{min} = 4$ do qual tentaremos corrigir o erro. Esse cálculo é feito facilmente através da soma mod 2 dos bits que compõem os rótulos decodificados sem violação:

  No exemplo em questão

$$L_{ki} = \sum_{j\neq i}^{\oplus} L_{kj} \qquad k = 0, 1$$

  i= índice da palavra CL recebida com violação.
  j= índice das 3 palavras CL restantes recebidas sem violação.

Esses 2 bits nos levam a um dos subcódigos de linha SCL1, SCL2, SCL3 ou SCL4 , cada qual representado por uma memória de correção. Caso mais de

um erro tenha ocorrido na palavra CL, a correção pode não ser possível e a presença dos erros podem ser apenas detetadas.

O diagrama em blocos do sistema codificação e decodificação pode ser visto na figura4.5.

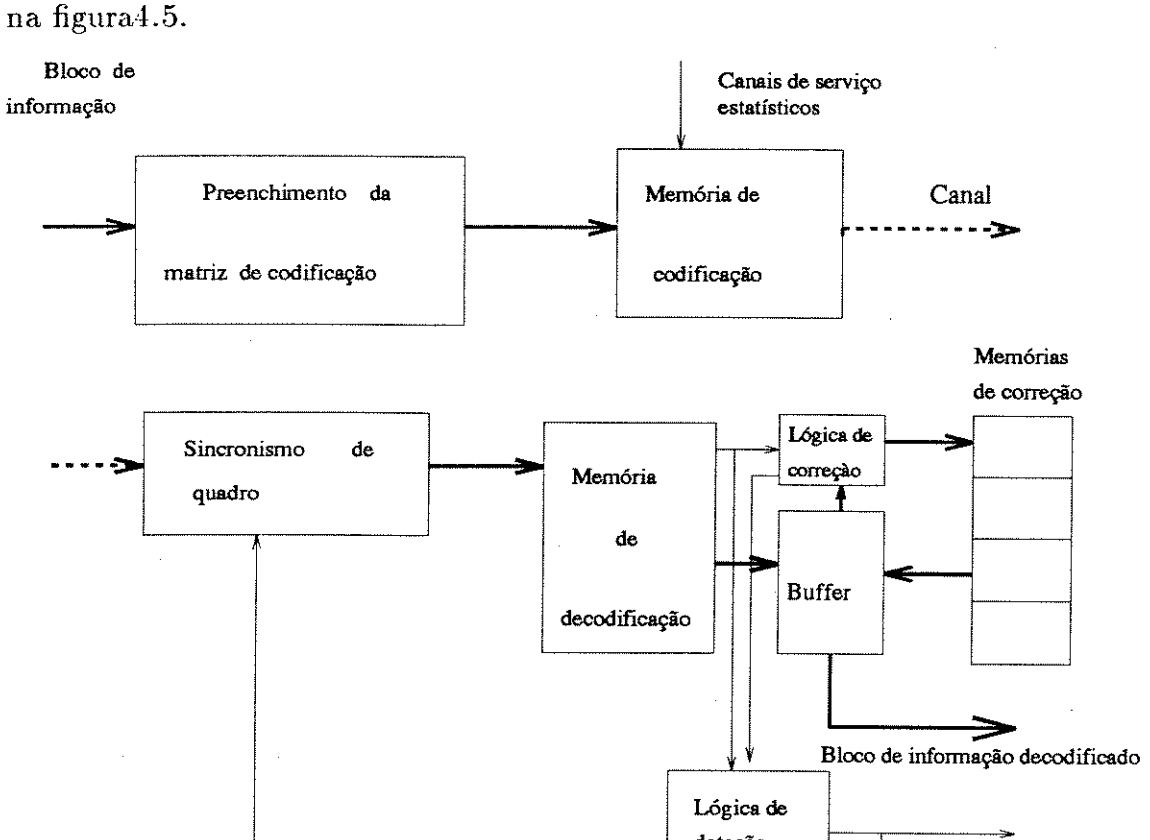


Figura 4.5: Diagrama em blocos do sistema de codificação/decodificação do CLCE fortemente restrito

4.4.5 Busca de um código de maior taxa

Na procura de melhores taxas partimos para a tentativa de particionar a constelação determinada por n=12, VSDC=6 com $|CL| = 792$. Nesse caso o melhor CLCE (capítulo 2) é o CLCE(12,6,4). Consequentemente tentaremos encontrar a árvore de partição para essa constelação cuja distância mínima de 4 aconteça a partir da terceira ramificação, ou solucionar o problema equivalente: encontrar 8 subcódigos disjuntos com pelo menos 64 pontos cada. Para isso implementaremos um método empírico–computacional ao qual denominaremos " Busca aleatória de partição "[6].

[6] O termo "Aleatória" não corresponde ao sentido probabilístico, mas aos vários caminhos tentados sem critérios teóricos

Esse método é uma tentativa de achar os 8 subcódigos disjuntos partindo de diversas condições iniciais dentro de uma sequência de busca fixa. Descrevemos o algorítmo a seguir.

Um possível algorítmo para particionamento

1. Calcular os pontos da constelação apartir das restrições dadas e armazená-los no vetor $\mathcal{CL}[*]$.
2. Escolher o ponto inicial de busca $\mathcal{CL}[j] \in \mathcal{CL}[*]$.
3. Executar o algorítmo que acha um subcódigo de linha com d_{min}=4 e 64 pontos apartir dos elementos de $\mathcal{CL}[*]$. Esse é o mesmo algorítmo usado nas buscas do capítulo 2, e é resumido a seguir:
   (a) O vetor subcódigo $\mathcal{SCL}[*]$ inicialmente vazio, recebe um elemento igual a $\mathcal{CL}[j]$.
   (b) Percorrer $\mathcal{CL}[*]$ a procura do próximo ponto que satisfaça $d(\mathcal{CL}[i], \mathcal{SCL}[l]) \geq 4$, $1 \leq l \leq |\mathcal{SCL}[*]|$.
   (c) Se encontrarmos i que satisfaça as condições de distância , incorporamo-no ao conjunto $\mathcal{SCL}[*]$ enquanto o número de pontos de $\mathcal{SCL}[*]$ for menor que 65.
4. Expurgar os elementos do subcódigo (achados no item anterior) do conjunto $\mathcal{CL}[*]$. Retornar a 2 enquanto: não se formarem 8 subcódigos e o número de pontos do último subcódigo obtido for igual a 64. Caso contrário ir para 5.
5. Se a saída do loop não se deu devido a insuficiência de pontos no último subcódigo, obtivemos uma partição válida. Senão voltar para 2 com o vetor $\mathcal{CL}[*]$ completado com toda a constelação original e novo ponto inicial de busca $\mathcal{CL}[j]$.

Aplicamos o algorítmo descrito para a condição VSDC= 6 e o melhor resultado forneceu apenas 7 subcódigos de 64 elementos (note que o algorítmo não assegura a inexistência da partição desejada). Partimos para o particionamento da constelação gerada por n=12 e VSDC= 8 com $|CL| = 900$. Aplicamos o algorítmo e conseguimos a partição desejada (nesse caso todos os pontos iniciais de busca nos levaram a partições). Os resultados anteriores nos levam ao seguinte questionamento:

> É possível, a partir de dados tirados da constelação, prever a existência ou não da partição que queremos?

Esse problema, embora não tenhamos conseguido resolver, é interessante de ser discutido para nos dar uma visão da dificuldade teórica envolvida:

- Cabe observar que esse tipo de constelação (formada apartir das restrições VSDC) não tem estrutura algébrica definida (não formam grupos simples) e com isso não é possível tratá-las pelas teorias de partição de reticulados [14].

- Uma segunda observação é que , ao contrário dos códigos de bloco corretores lineares, a distribuição de distâncias depende do ponto de refêrencia na constelação significando irregularidades no espaço de trabalho.
- A primeira idéia que temos é nos utilizar de parâmetros (calculáveis por computador) da constelação analisada que nos forneçam algum esclarecimento da questão. Por exemplo: à quantidade de pares não ordenados entre pontos da constelação que distam de 2 denominamos NP2 e à quantidade de pares a distância ≥ 4 denominamos NP4. A priori essas quantidades poderiam nos levar a uma indicação para o problema, já que ao efetuarmos a "partição ótima" nosso objetivo é a eliminação do número máximo de pares a distância 2, conforme veinos no diagrama 4.6. Porém ao idealizarmos a partição em que eliminamos

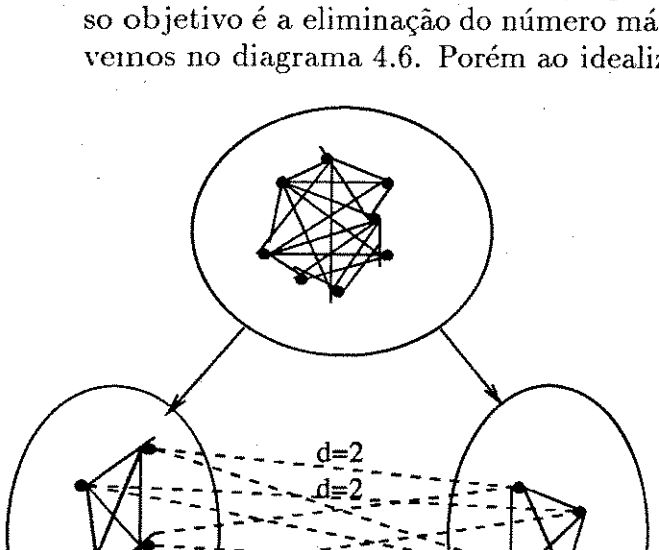


Figura 4.6: Diagrama de uma partição mostrando o rompimento desejado de ligações de distância 2

primeiro todos os pares de distância 2 não chegamos a conclusões úteis, pois normalmente NP2 <<< NP4. Isso nos leva a notar que não é possível romper uma ligação de distância 2 sem romper várias ligações de distância maior que 2. Concluimos que uma linha de pesquisa para o problema é determinar um número médio máximo de ligações de $d \geq 4$ rompidas por ligação de $d = 2$ rompida.

Com fins de ilustração podemos comparar as quantidades NP4/NP2 para as constelações de n=12 analisadas anteriormente:

VSDC=6 $\rightarrow$ NP4/NP2= 23.5 $\rightarrow$ 7 subcódigos.
VSDC=8 $\rightarrow$ NP4/NP2= 24.5 $\rightarrow$ 8 subcódigos.

Seguem as tabelas para os códigos citados (com palavras código em notação hexadecimal).

n=8 e VSDC=4:

SCL1											
D4	B1	A6	9A	8D	72	65	59	4E	2B		
SCL2											
D2	C5	B4	A3	8E	71	5C	4B	3A	2D		
SCL3											
D1	C6	B2	9C	8B	74	63	4D	39	2E		
SCL4											
CC	C3	AA	A5	99	96	69	66	5A	55	3C	33

n=12 e VSDC=8:

SCL1									
0B7	CF	F9	12F	157	17A	19B	1BC	1E3	238
250	267	29E	2AD	2D3	2EA	2F4	335	34B	36C
387	3B2	3D8	43D	45B	46E	4AB	4D5	4F2	51E
533	54D	574	5A5	5C6	5E8	60F	636	669	699
6CC	72A	751	794	83E	86B	875	89D	8D6	8EC
939	94E	9A6	9C5	9F0	A17	A5A	A8B	AB1	B0D
B23	B54	BA8	BC2	C57	C5C	C8E	C93	cB4	CC9
SCL2									
0AF	0D7	0FA	137	14F	179	19D	1E5	23D	25B
26E	2B3	2CD	31E	32B	355	372	388	3C3	3A6
43B	45D	467	49E	4B5	4CB	4EC	52D	553	56A
587	5B2	5D4	617	671	6A9	6C6	6D8	719	734
74C	78A	85E	86D	873	89B	8B6	92E	974	9A3
9C6	9C9	A0F	A3A	A95	AAC	ACA	AE1	B13	B25
B58	C3C	C8D	CAA	CD1	D0B	D15	D61	D98	DA4

SCL3									
0BB	0DD	0E7	13D	15B	16E	18F	1B6	1E9	22F
257	279	2B5	2CB	2EC	2F2	333	34D	374	399
3AA	3C6	437	44F	47A	4AD	4D3	4F4	52B	555
59A	5B1	5CC	5E2	61B	63C	663	687	70E	725
752	768	7C1	876	897	8AE	8DA	8F1	91E	927
978	9C3	9D4	A1D	A4E	A65	AA3	AB8	BAB	B2C
B51	B62	B85	B92	BC8	C39	C6C	C8B	C9C	CB2

SCL4									
0BD	0DB	0EE	13B	15D	167	197	1F1	237	24F
27A	2AB	2D5	32D	353	38E	3B4	3C9	3E2	42F
457	479	4B3	4CD	535	54B	56C	572	599	5A6
61D	665	696	6AC	6C3	6F0	707	71A	754	7A1
B7C	88F	8BA	8E3	936	95A	969	99C	9A5	A1B
A2E	A56	A71	ACC	B15	B38	B64	B83	BD0	C1E
C66	C95	CA9	CCA	D0D	D13	D2A	DB0	DC1	E23

SCL5									
0BE	0EB	0F5	15E	16D	173	1A7	1B9	276	28F
2D9	317	32E	378	39A	3C5	47C	497	4CE	50F
536	559	59C	5AA	5C3	5E4	61E	627	639	64B
655	6B2	6E1	746	789	7D0	82F	857	879	8B3
BCD	8E6	91B	935	96A	98E	9D1	A3C	A63	A96
AA5	AAA	AF0	B29	B32	B4C	C1D	C3A	C65	CAC
CD2	D23	D54	D85	DC8	E13	E58	EC4	F0A	F24

SCL6									
0DE	0ED	0F3	13E	16B	175	1C7	1D9	27C	297
2AE	2B9	30F	356	39C	3A3	3CA	3E4	476	48F
4BA	517	539	54E	5AC	5D2	5E1	62B	635	647
659	6D4	6E2	726	770	785	837	84F	87A	8AB
8BC	8D5	91D	953	966	996	9B1	9CC	A1E	A2D
AB2	AC3	AD8	B2A	B34	B45	B89	C1B	C2E	C63
CA5	CC6	CE8	D32	d49	d83	E38	E4A	E64	E8C

SCL7									
0F6	17C	19E	1AB	1B5	1CD	1D3	23E	26B	275
29B	2A7	2CE	2F8	31D	347	35A	3AC	3D4	3E1
45E	46D	473	49D	4AE	4C7	51B	527	5B8	5CA
666	6B1	6C9	6D2	716	729	783	83B	85D	867
8AD	8CB	9aF	956	971	999	9B2	9E4	A53	A6C
A9C	AC5	AE2	B26	B49	B8A	C17	C6A	C74	C9A
CA3	CCC	D1C	D43	D86	DD0	E0B	E25	E32	EA8

SCL8									
0FC	176	1AD	1B3	1CB	1D5	25E	26D	273	29D
2B6	2C7	31B	327	33C	36A	3CC	3D2	43E	46B
475	49B	4A7	4D6	51D	547	55A	58E	5B4	678
6AA	6D1	6E4	731	749	798	83D	85B	86E	89E
8E5	8F2	917	92B	94D	9B8	A55	A93	AA9	B0E
B43	B70	B94	BA2	C0F	C33	CC3	CD8	D25	D51
D62	D89	D92	DC4	E16	E19	E2C	E61	E85	EB0

Finalmente podemos exemplificar o CLCE com VSDC=8 e n=12 através de uma matriz de palavras código de 6 colunas:

I	I	I	I	I	P
I	I	I	I	I	P
I	I	I	I	I	P
I	I	I	I	I	I
I	I	I	I	I	I
I	I	I	I	I	I
I	I	I	I	I	I
I	I	I	I	I	I
I	I	I	I	I	I

Temos então os seguintes parâmetros para o CLCE:

- r= 0.71
- 48 canais de serviço a taxa média de $\frac{1}{12\times512}$ taxa de linha.

4.4.6 Conclusões

- Os CLCEs obtidos por esse método podem se aproximar rapidamente da taxa do seu CL base, bastando aumentar o número de colunas da matriz de palavras código. Geramos assim um compromisso Taxa de código X Atraso na codificação/decodificação (complexidade).
- Esse CLCEs se aplicam particularmente aonde um confinamento estrito (determinístico) da VSDC é requisitado, pois sabemos que um bom confinamento em média (estatístico) [8] também produzirá um bom nulo espectral .

4.5 CLCEs Q_ários (Q > 2)

Embora essa tese se concentre em CLCEs binários, pudemos constatar que a expansão de faixa (1/r) requerida na transmissão de CLCEs binários muitas vezes

inviabiliza a sua aplicação em sistemas de faixa limitada. Por isso citaremos suscintamente algumas linhas de pesquisa que lidam com a construção desses CLCEs.

4.5.1 Método de Calderbanc e Mazo

Calderbank,L. e Mazo [9] usaram métodos de balanceamento por modificação de códigos de treliça aplicados a subreticulados Λ do Z^n que têm partições em cosets que possuem a propriedade: Todo ponto de Λ está no mesmo coset que seu negativo. Assim, ambos os pontos podem ser usados para representar o mesmo bit de informação sem afetar a estrutura do código de treliça original. A seleção entre os pontos é feita de modo a confinar os estados de SDC na linha.

4.5.2 CLCEs ternários

Ferreira,H.C. et alle [10] propõem CLCEs ternários com capacidade de correção de 1 erro achados por métodos de busca computacional semelhantes aos usados por nós no capítulo 2 para o caso binário.

4.6 Outros métodos

4.6.1 Baseados em códigos de controle de erros não lineares

Como por exemplo aqueles gerados pela matriz de Hadamard [11]

4.6.2 Baseados em códigos de treliça binários

Nesse caso poderíamos trabalhar diretamente sobre o diagrama de estados do codificador, alocando valores aos braços (transições de estado), que observem as restrições:

- O peso para qualquer caminho fechado sobre o diagrama (loop) é metade da quantidade total de bits do loop $\Leftrightarrow$ balanceamento.
- Gerar códigos não catastróficos.

Vide ilustração na figura 4.7.

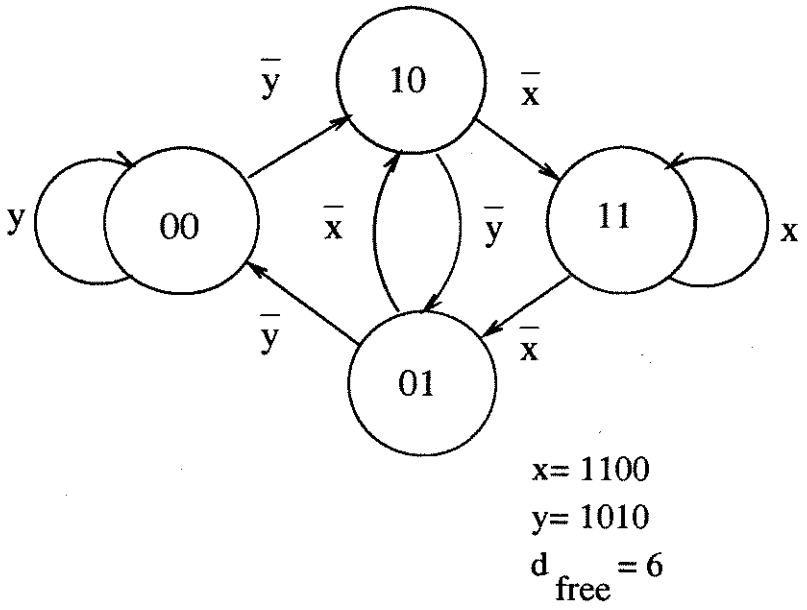


Figura 4.7: Código de treliça balanceado de 4 estados e taxa 1/4

Outra técnica descrita em [12] transforma um código linear convolucional numa treliça balanceada preservando d_{free}. A transformação envolve um mapeamento do conjunto de bits que formam os braços do código convolucional em outro conjunto de bits balanceados que preservam a distância de Hamming, ou seja, a distância entre qualquer par de braços no código convolucional original é menor ou igual a distância entre os respectivos novos braços obtidos pelo mapeamento.

4.7 Conclusões

Nesse capítulo são abordados vários métodos construtivos para CLCEs com estruturação algébrica obtidas a partir de códigos de controle de erros lineares. Cabe citar como principal contribuição a classe dos códigos fortemente restritos em VSDC com capacidade de correção de 1 erro.

Bibliografia

[1] Lin,S. & Costello,D.J.:"Error control coding: Fundamentals and applications",Prentice Hall, 1983.

[2] Peterson,W.W. & Weldon,E.J.J.:"Error correcting codes" ,MIT Press, 1972.

[3] Krzymlen,W.A.:"Transmission performance analysis of a new class of line codes for optical fiber systems", IEEE Trans. on Comm. , abril 1989.

[4] O'Reilly,J.J. & Popplewell,A.:"Spectral characteristics of a class of DC free error-correcting transmission codes", Electronics letters, julho 1988.

[5] O'Reilly,J.J. e Popplewell,A.:"Class of disparity reducing transmission codes with embedded error protection", IEE Proceedings , abril 1990.

[6] Ungerboeck,G.:"Channel coding with multilevel/phase signals", IEEE trans. on inform. theory, janeiro 1982.

[7] Cusack,E.L.:"Error control codes for QAM signalling", Electronics Letters , janeiro 1984.

[8] Justensen,J.:"Information rates and power spectra of digital codes", IEEE trans. on inform. theory, vol IT-28,pp.457-472 1982.

[9] Calderbank,A.R. et alle :"Baseband trellis codes with spectral null at zero", IEEE trans. on inform. theory,pp.425-434 1988.

[10] Ferreira,H.C. et alle :"On ternary error correcting line codes", IEEE trans. on comm., maio 1989.

[11] Litsyn,S.N. e Barg,A.M.:"DC-constrained codes from Hadamard matrices", IEEE trans. on inf. theory, maio 1991.

[12] Ferreira,H.C. et alle:"Hamming distance preserving mapping and trellis codes with constrained binary symbols",IEEE trans. on inf. theory, setembro 1989.

[13] Deng,R.H. & Herro,M.A.:"DC-free coset codes", IEEE Trans. on Inf. Theory,vol.34, julho 1988.

[14] Forney,G.D.J.:"Coset codes - Part I: Introduction and geometrical classification",IEEE Trans. on Inf. Theory, setembro 1988.

[15] Mac Williams,F.J. & Sloane,J.A.:"The theory of error correcting codes".North Holand, New York,1977.

[16] Cattermole,K.N.:"Codes for digital line transmission", Electron. Lett.,vol.11,pp548-550,1975.

[17] Proakis,J.G:"Digital communications". Mc Graw-Hill, Singapure, 1989.

[18] Blaum,m. et Alle: " Error correcting codes with bounded running digital sum", IEEE Trans. on Inf. Theory , janeiro 1993.

Capítulo 5

Exemplo de aplicação de CLCEs

5.1 Introduçao

Esse último capítulo objetiva expor um exemplo de sistema de comunicações no qual introduzimos um CLCE. Apresentamos as principais características do sistema de comunicações óticas de deteção direta e analisamos o desempenho do CLCE dentro do mesmo, através de cálculo de parâmetros como: ganho de potência, espectro do código, etc.

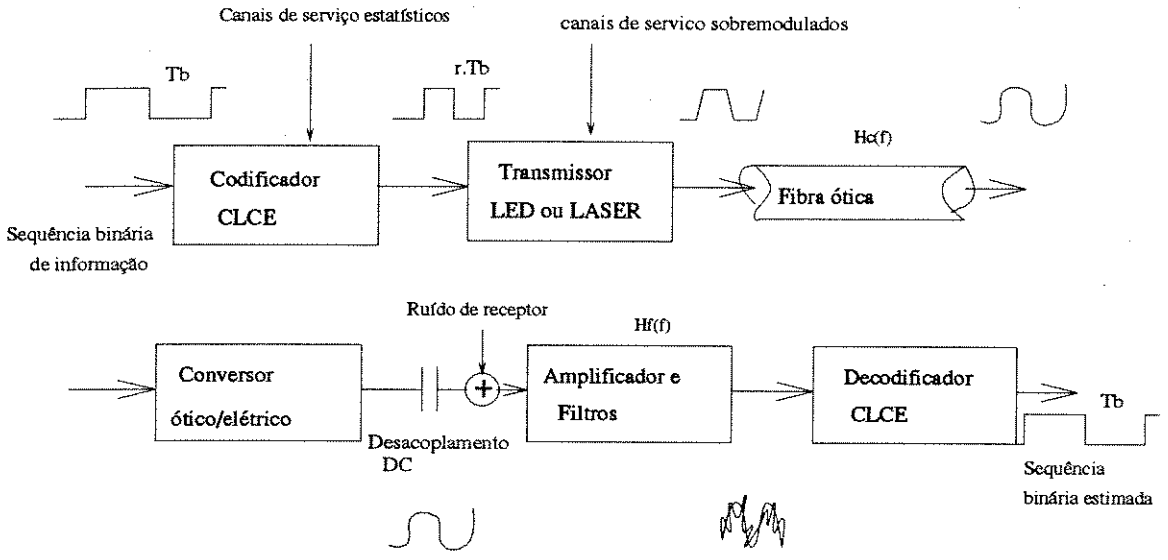


Figura 5.1: Diagrama em blocos de um sistema de comunicações óticas de deteção direta

O sistema de comunicações óticas proposto está esquematizado na figura 5.1 e é composto pelos blocos:

(I) CODIFICADOR CLCE: Representa uma implementação de um dos tipos de CLCEs descritos ao longo da tese.

(II) CONVERSOR ELETRO-ÓTICO ou FONTE ÓTICA: Representa a implemen-

tação de um dos dois componentes de estado sólido "LED" ou "LASER" , que convertem sinais de corrente elétrica em sinais de potência ótica.

(III) FIBRA ÓTICA MONOMODO: É o meio pelo qual o sinal ótico é transmitido.

(IV) CONVERSOR ÓTICO-ELÉTRICO ou DETETOR ÓTICO: Representa um dos fotodiodos de estado sólido PIN ou APD (fotodiodo de avalanche) que convertem sinais de potência ótica em sinais de corrente elétrica.

(V) AMPLIFICADORES E FILTROS: Processam o sinal de corrente para ser amostrado pelo decodificador.

(VI) DECODIFICADOR CLCE: Representa a implementação de amostrador + decodificador para o CLCE usado. Pode ser de "decisão abrupta" quando trabalha com dois níveis de quantização por amostra; ou pode ser de decisão suave quando trabalha com mais de dois níveis de quantização por amostra.

Se retirássemos o codificador CLCE do sistema apresentado, surgiriam os seguintes problemas que degradam a taxa de erro do enlace:

(A) Flutuação na linha de base ou "base line wander"[1]: O acoplamento AC entre os blocos detetor e amplificador/filtro é um filtro passa alta que suprime o conteúdo espectral da informação nas baixas frequências. Isso se reflete no sinal temporal como uma superposição de "nível DC aleatório" que fecha o olho binário na tomada de decisão.

(B) O canal de serviço sobremodulado com conteúdo espectral em baixas frequências interfere com o sinal de assinante que tem parte de energia não desprezível na mesma banda.

Os principais mecanismos de degradação do sinal transmitido no sistema são: atenuação de potência ótica, distorção dos pulsos óticos por dispersão e ruído introduzido no receptor.

A atenuação na fibra ótica é função fundamentalmente do comprimento de onda central da fonte ótica e é comum trabalhar-se em janelas espectrais onde encontra-se (no estado da arte) vales de atenuação na característica "atenuação X comprimento de onda". Como exemplos comuns temos as janelas 1,3μm (0,5 dB/Km) e janela 1,55μm(0,2 dB/Km)

A fibra ótica monomodo é aquela que permite a propagação de apenas um modo espacial,ou seja, o modo HE11 é a solução da equação de onda eletromagnética aplicada ao modelo de guia de onda cilíndrico [2]. De modo simples podemos dizer que a dispersão numa fibra ótica monomodo é o efeito físico da variação natural da velocidade de propagação da luz com o seu comprimento de onda. Assim, se excitarmos uma ponta da fibra com um impulso de potência ótica com uma fonte ótica real, veremos na outra ponta um pulso alargado. Logo, dado o formato do espectro da fonte $P(\lambda)$ e as características de dispersão da fibra $\tau(\lambda)$, fica caracterizada a resposta em frequência do bloco linear "fibra ótica" $H_c(f)$.

No sistema exemplo consideramos que todas as formas de ruído são gerados no receptor ótico, a saber:

1. Ruído quântico de deteção ótica.
2. Ruído de multiplicação no APD.
3. Ruído térmico de receptor ótico.
4. Ruído do elemento ativo do amplificador.

De modo mais geral, um sistema de comunicaçoes óticas de deteção direta apresenta outros tipos de ruído como: de fonte ótica, ruído de partição modal, ruído modal etc[3].Porém ao considerarmos fontes do tipo LED ou laser monomodo e transmitindo em taxa adequada, podemos modelar apenas o ruído dominante procedente do receptor.

5.2 Modelamento dos elementos sistêmicos

5.2.1 Fonte ótica

Tem o espectro de densidade de potência ótica normalizado modelado por uma gaussiana

$$P(\lambda) = \frac{1}{\sqrt{2\pi}\sigma} \exp[\frac{-(\lambda - \lambda_c)^2}{2\sigma^2}]$$

Quando modulada por um degrau de subida de corrente, a fonte apresenta saída ótica com resposta de primeira ordem com tempo de subida Ts. Quando modulada por um degrau de descida de corrente, a fonte apresenta saída ótica com resposta de primeira ordem e tempo de descida Td. Esse tratamento ignora portanto as oscilações de relaxação do laser, consideradas irrelevantes para a discussão aqui desenvolvida.

5.2.2 Fibra ótica monomodo de comprimento L

É caracterizada pela função de transferência $H_c(f)$ que é calculada a partir das características de atenuação A(L,λ) e de dispersão $\tau(\lambda)$. Como exemplo citamos uma fibra monomodo do tipo "depressed cladding" com dispersão nula em $\lambda_0 = 1,310\mu m$:

$$\tau(\lambda) \simeq a + b\lambda^2 + c\lambda^{-2}$$

$$A(L, \lambda_0) = L \times 0,5 dB/Km \quad (L\ em\ Km)$$

Calcula-se a função transferência [7] segundo

$$H_c(f) = \int_\lambda P(\lambda) A(L, \lambda) \exp[-jw(\tau(\lambda) - < \tau >]d\lambda$$

onde$< \tau >= \int_\lambda P(\lambda)\tau(\lambda)d\lambda$

5.2.3 Conversores ótico elétricos

Supomos que tanto o PIN quanto o APD apresentam tempos de subida e descida desprezíveis em relação aos períodos de bit envolvidos. Têm como parâmetros de interesse:

- A Responsividade $R = \eta q/h\nu$ nos dá a relação entre potência ótica incidente e corrente elétrica gerada (em A/W).
  onde η é a eficiência quântica, q é a carga elementar , h a constante de Plank e ν a frequência da portadora ótica.

- Ganho de multiplicação por avalanche M . No caso de PIN , M=1.

- Ruído quântico de multiplicação: O processo de geração de elétrons primários num semicondutor iluminado é de natureza estatística e o número de elétrons (N) é bem descrito pela distribuição de Poisson

$$p(N) = \exp(-\bar{P})\bar{P}^N/N!$$

  onde $\bar{P}$ é o número médio de elétrons devido a potência incidente no intervalo de tempo analisado dado por:

$$\bar{P} = P(t)\frac{\eta}{h\nu}\triangle T$$

  O fato de não se poder predizer o número exato de elétron-lacunas gerados pela conversão potência corrente, representa um tipo de ruído chamado de "ruído quântico". O processo poissoniano de deteção ocorre nos dois tipos de detetores PIN e APD, porém no caso do APD esses elétrons primários poissonianos sofrem uma multiplicação ao entrar numa região de campo elétrico elevado,ou seja, cada elétron primário dá origem a M elétrons em média na saída do dispositivo. O processo de multiplicação também é estatístico e a distribuição do cascateamento desses dois processos torna-se bastante complicada [5][4]. No nosso modelamento para ruído não necessitamos de máxima exatidão, por isso adotaremos a aproximação proposta por Smith & Garret[6]:

$$S_D = 2q < i_0 >< M^2 >$$

  onde S_D é a densidade espectral de corrente de ruído na saída do detetor quando a fotocorrente média no período de bit é $< i_0 >$. Nota-se que temos um processo não estacionário com ruído dependente do padrão de bit incidente.O ganho médio quadrático por avalanche é $< M^2 > \simeq M^{2+x}$ onde $0 \leq x \leq 1$, é um parâmetro que depende do material e tipo de APD. Adotamos também a aproximação Gaussiana quando supomos que o processo em questão é Gaussiano.

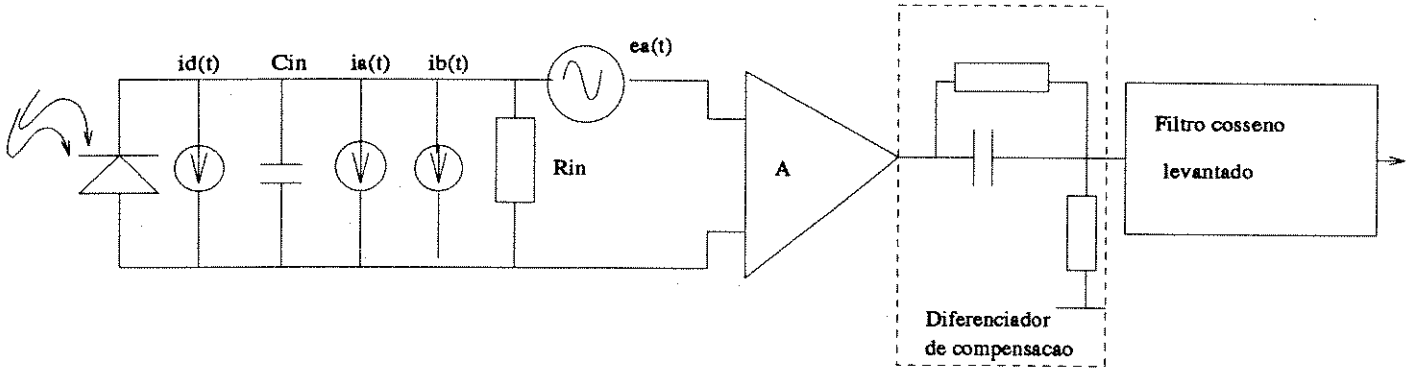


Figura 5.2: Front-end tipo alta impedância (HZ)

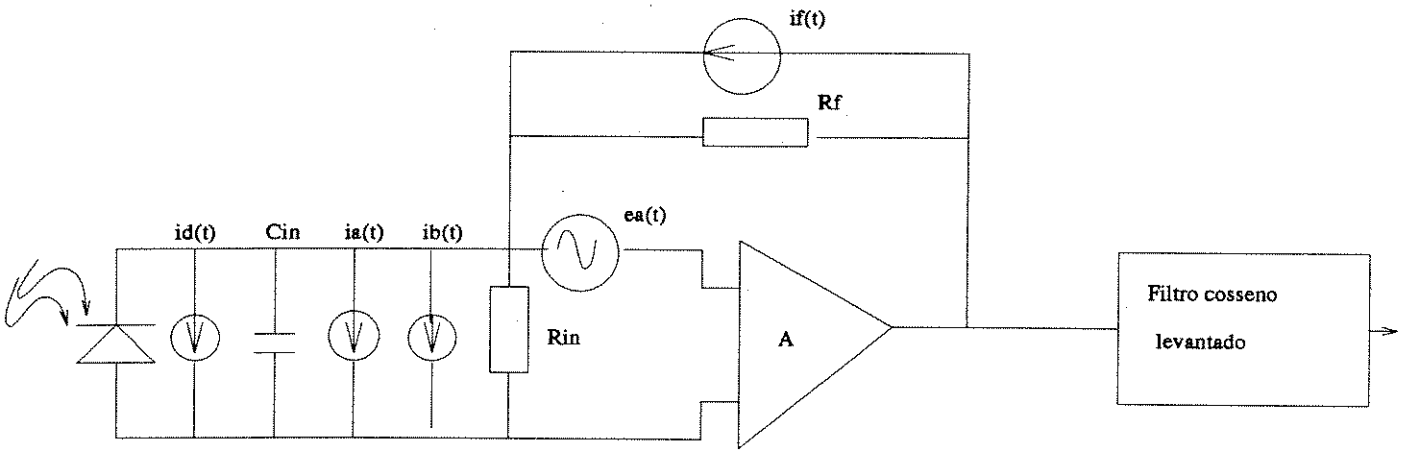


Figura 5.3: Front-end tipo transimpedância (TZ)

5.2.4 Amplificadores/filtros

Existem dois tipos básicos de preamplificadores (ou front-ends) , cujos modelos para análise de ruído estão descritos nas figuras 5.2 e 5.3: HZ é a sigla para preamplificador de alta impedância e TZ é a sigla para preamplificador de transimpedância. O significado dos elementos dos circuitos das figuras mencionadas estão listados a seguir:

- A é um amplificador de transimpedância ideal.
- $id(t)$ é a fonte de ruído quântica devido ao detetor.
- $ib(t)$ é a fonte de ruído térmico devido ao resistor de polarização na entrada Rb.
- $if(t)$ é a fonte de ruído térmico devido ao resistor de realimentação R_f.
- $ia(t)$ é a fonte de ruído shunt equivalente devido ao elemento ativo do amplificador A.

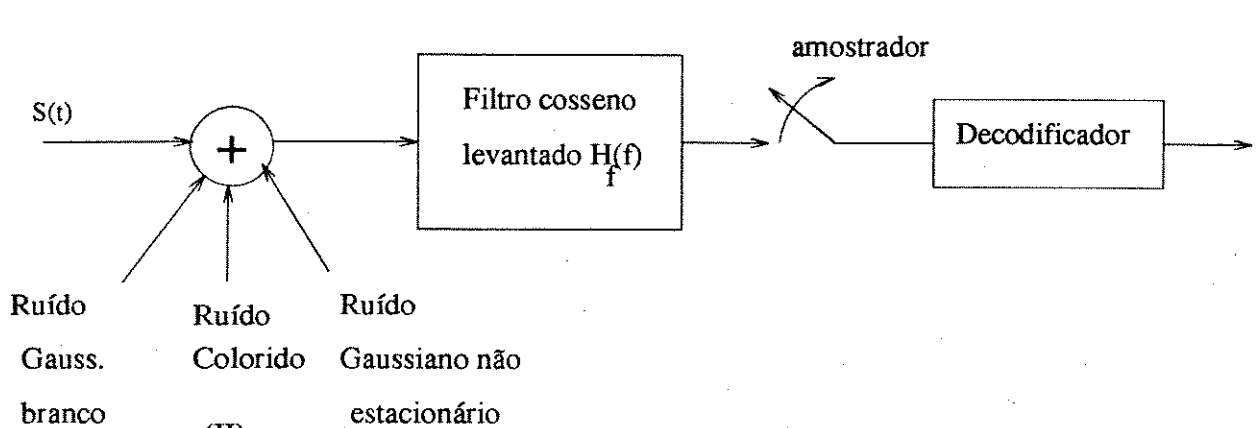


Figura 5.4: Esquema de ruídos de receptor

- $ea(t)$ é a fonte de ruído série equivalente devido ao canal do elemento ativo de A.

5.3 Equacionamento dos ruídos de receptor

Os amplificadores das figuras 5.3 e 5.2 são projetados de modo que suas respostas em frequência sejam planas dentro da faixa em que o filtro cosseno levantado deve atuar,ou seja, o filtro molda o espectro do receptor. Assim podemos dizer que olhando pela entrada do filtro existem três tipos de ruído distintos conforme mostra a figura 5.4:

I Gaussiano branco - devido aos ruídos térmicos das resistências de entrada.

II Gaussiano colorido - devido a $ea(t)$.

III Gaussiano não estacionário (localmente estacionário) - devido ao fotodetetor.

Chamemos de S_I,S_{II} e S_{III} as densidades espectrais de corrente de ruídos I, II e III. Por exemplo para o preamplificador de alta impedância a FET de $1\mu m$ de canal:

$$S_I = 4KT/R_{in}$$

onde R_{in} é a resistência equivalente de entrada.

$$S_{II} = 1,589KT/gm[(1/R_{in})^2 + (2\pi f C_{in})^2]$$

onde gm é a transcondutância do elemento ativo e C_{in} a capacitância equivalente de entrada.

Para o amplificador de transimpedância prático o ruído térmico do resistor de realimentação é dominante:

$$S_{II} \simeq 0$$

$$S_I \simeq 4KT/R_f$$

5.4 Teoria simplificada para cálculo de ganhos por introdução do CLCE no sistema

5.4.1 Introdução

Definamos o ganho de um CLCE em relação a um CL como 10 log(P_ótica CL)/ (P_ótica CLCE) dada uma taxa de erro , onde P_ótico refere-se a potência de pico acoplada pela fonte à fibra e que permite ao sistema trabalhar numa dada taxa de erro aceitável.(Ex: 10^{-9} para 34 Mbit/s). Baseado no modelo da figura 5.4 pode-se equacionar a potência de ruído nos instantes de amostragem σ^2_{eq}

$$\sigma^2_{eq} = \int_B [S_I(f) + S_{II}(f) + S_{III}(f)]H_f(f)df(A^2) \tag{5.1}$$

onde A é o ganho de amplificação conforme mostram as figuras 5.2 e 5.3.

5.4.2 Aproximações

A - Com o uso de filtro de Nyquist(cosseno levantado) e considerando os ruídos tipo I e III, resulta em amostras recebidas independentes entre si ou $p(\vec{r}|\vec{v}) = p(r_1|v_1)p(r_2|v_2)...p(r_n|v_m)$, onde $\vec{v}$ é o sinal códificado e $\vec{r}$ o sinal amostrado recebido. Porém para o ruído colorido II isso não é verdadeiro, contudo a pequena diferença que faz quando do uso de APD nos leva a supor a independência.

B - Se desconsiderarmos ,em princípio , o efeito da dispersão; teremos uma sequência de pulsos NRZ atravessando um filtro de Nyquist. Portanto o espectro final de pulso não é de Nyquist e teremos interferência entre símbolos (I.E.S.) nos instantes de amostragem. O fechamento de olho certamente depende do tipo de código usado e portanto é um fator a se considerar no ganho de codificação. Porém , faz-se uma outra simplificação supondo olhos com mesma abertura para os códigos comparados (CL e CLCE) para que possamos ter expressões algébricas simples para os ganhos. A abertura de olho e dispersão serão levadas em conta na simulação computacional ao final do capítulo.

C - Suporemos que estamos trabalhando com sistemas limitados por atenuação ,ou seja, a redução do período de bit ao trocarmos o CL pelo CLCE não coloca o sistema em regime de limitação por dispersão.

D - Considera-se razão de extinção nula, ou seja, potência ótica de fonte nula para pulso zero.

E - Com as hipóteses anteriores, podemos usar um decodificador suave de máxima "verossimilhança" [8] no qual tentamos achar a palavra código $\vec{v}_j$ transmitida que oferece

$$\max_{\vec{v}_j} p(\vec{r}|\vec{v}_j)$$

Pela independência entre amostras

$$p(\vec{r}|\vec{v}_j) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp \frac{-(r_i - v_{ji})^2}{2\sigma_{ji}^2} \tag{5.2}$$

Em vistas da equação 5.2, os efeitos da atenuação podem ser incorporados ao ruído sem perda de generalidade, assim mantemos $v_{ji} \in \{0,1\}$ e σ_{ji} é o desvio padrão do ruído total não estacionário devido ao bit i da palavra v_j incorporando o efeito de normalização em relação à atenuação.

Usando como métrica $\ln p(\vec{r}|\vec{v}_j)$: queremos $\vec{v}_j$ que maximize a métrica

$$-\sum \frac{r_i^2 - 2r_i v_{ji} + v_{ji}^2}{2\sigma_{ji}^2}$$

Fazendo a mudanças de variáveis $r_i = (\grave{r}_i + 1)/2$ e $v_{ji} = (\grave{v}_{ji} + 1)/2$, resulta a métrica

$$-\sum \frac{\grave{r}_i{}^2 - 2\grave{r}_i\grave{v}_{ji} + \grave{v}_{ji}{}^2}{2\grave{\sigma}_{ji}{}^2}$$

onde $\grave{\sigma}_{ji} = 2\sigma_{ji}$ e $\grave{v}_{ji} \in \{-1,1\}$. O fato de a maioria das palavras do CLCE ser balanceada nos sugere o uso da métrica simplificada (que não nos leva ao decodificador de máxima verossimilhança, porém não perde muito em termos de probabilidade de erro):

$$\mu(\vec{v}_j) = \sum \frac{-\grave{r}_i\grave{v}_{ji}}{\grave{\sigma}_{ji}{}^2} \tag{5.3}$$

5.4.3 Cálculo da probabilidade de erro para o decodificador suave

Aproximemos a probabilidade de erro de bloco (Pe) pelo "Union Bound"[8]. Assim, calcula-se as probabilidades dos eventos de erro dois a dois: $p(\mu_0 < \mu_1 | v_o transmitido) = p(\mu_0 - \mu_1 < 0 | v_o transmitido) = p(\sum -\grave{r}_i(\grave{v}_{0i}/\grave{\sigma}_{0i}{}^2 - \grave{v}_{1i}/\grave{\sigma}_{1i}{}^2) < 0)$.

Mas desprezando a I.E.S temos $r_i = v_{0i} + X_{0i}$ onde X_{0i} é a amostra de ruído gaussiano de média zero e variança σ_{0i}^2 . Fazendo também a mudança de variáveis $\grave{X}_{0i} = 2X_{0i}$ e $\grave{v}_{0i} = 2v_{0i} - 1$ resulta:

$$p(\sum(\grave{v}_{0i} + \grave{X}_{0i})(\grave{v}_{0i}/\grave{\sigma}_{0i}^2 - \grave{v}_{1i}/\grave{\sigma}_{1i}{}^2) > 0) = \tag{5.4}$$

$$p(\underbrace{\sum \grave{X}_{0i}(\grave{v}_{0i}/\grave{\sigma}_{0i}{}^2 - \grave{v}_{1i}/\grave{\sigma}_{1i}{}^2)}_{A} > \underbrace{\sum \grave{v}_{0i}(\grave{v}_{0i}/\grave{\sigma}_{0i}{}^2 - \grave{v}_{1i}/\grave{\sigma}_{1i}{}^2)}_{B}) \tag{5.5}$$

O termo em A é uma variável aleatória gaussiana de média nula e variança aproximada (supondo balanceamento nas posições que contribuem para a distância de Hamming entre as palavras : d_{min}) $(1/\sigma_{ton}^2 + 1/\sigma_{toff}^2)^2 \times 4(d_{min}\sigma_{ton}^2/2 + d_{min}\sigma_{toff}^2/2)$. O termo em B é o limiar de decisão que para o pior caso de probabilidade de erro é dado por $(1/\sigma_{ton}^2 + 1/\sigma_{toff}^2)d_{min}$. Onde σ_{ton}^2 e σ_{toff}^2 são as potências dos ruídos totais para recepção de nível um ou zero respectivamente. Nota-se que σ_t^2 está normalizado pelo fator de atenuação α , da fibra ,da fonte e receptor ótico, portanto desnormalizamos essa variança fazendo $\sigma_t = \alpha\sigma_{eq}$:

Assim podemos escrever o teto aproximado sobre a probabilidade de erro média de palavra no decodificador suave:

$$PER \leq (2^k - 1)Q(\sqrt{\frac{d_{min}}{2\alpha^2(\sigma^2_{eqon} + \sigma^2_{eqoff})}}) \tag{5.6}$$

onde $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^\infty \exp(-x^2/2)dx$

5.4.4 Cálculo do ganho

Supomos P_1 como a potência ótica de pico recebida pelo detetor do sistema não codificado ($d_{min} = 1$) para uma taxa de erro suficientemente baixa. A amostra no receptor $r \sim P_1$ e $\alpha = c/P_1$, onde c é uma constante de conversão corrente amostrada na saída do filtro cosseno levantado(equivalente de entrada) - potência ótica recebida, assim

$$PER \leq 2^{k_1} Q(\sqrt{\frac{P_1^2}{2c^2(\sigma^2_{eqon} + \sigma^2_{eqoff})}})$$

Chamemos de P_2 a potência ótica de pico recebida no sistema codificado($d_{min} > 1$).Ao impormos a mesma taxa de erro baixa nos dois sistemas e desprezando o fator 2^k para códigos com k pequenos, resulta:

$$(\frac{P_1}{P_2})^2 = d_{min} \frac{\sigma^2_{1eqoff} + \sigma^2_{1eqon}}{\sigma^2_{2eqoff} + \sigma^2_{2eqon}}$$

Nesse ponto podemos esclarecer que se o sistema sofresse ação apenas de ruído gaussiano branco: $\sigma^2_{1eqon} = \sigma^2_{1eqoff} \sim N_0/Tb_1$, , $\sigma^2_{2eqon} = \sigma^2_{2eqoff} \sim N_0/Tb_2$, onde Tb é o período de bit e N_0 a densidade de ruído branco, teríamos por ganho:

$$G = P_1/P_2 = \sqrt{d_{min} r_2/r_1}$$

que é a metade do ganho em dB para sistemas PSK coerentes codificados. Mas de modo geral

$$G = (d_{min} \frac{\sigma^2_{1eqoff} + \sigma^2_{1eqon}}{\sigma^2_{2eqoff} + \sigma^2_{2eqon}})^{1/2} = (d_{min}\theta)^{1/2} \tag{5.7}$$

onde definimos $\theta \equiv (\frac{\sigma^2_{1eqoff} + \sigma^2_{1eqon}}{\sigma^2_{2eqoff} + \sigma^2_{2eqon}})$ ≡relação de ruídos e $r_r \equiv r_2/r_1 \equiv$ taxa relativa.

5.4.5 Ganhos a partir de simulação

Por meio de simulação semi-analítica[3] é possível levar em conta os efeitos de I.E.S. Na simulação calculamos, para uma sequência pseudo aleatória , a probabilidade de erro média. Para isso utilizamos na equação (5.4) valores de amostras recebidas na simulação. Nesse contexto o decodificador não é de máxima verissimilhança porém é eficiente. Os resultados da simulação e sua implementação serão discutidos mais adiante.

5.4.6 Ganho para Front-End de alta impedância(HZ) a PIN-FET(canal de $1\mu m$)

No caso de PIN desprezamos o ruído quântico o que nos leva ao ruído estacionário :

$$\sigma_{eq}^2 = \int_0^{1/2+\beta} (4KT/R_{in}) + 1,589KT/gm[(1/R_{in})^2 + (\frac{2\pi C_{in}}{Tb})^2 y^2] H_f^2(y/Tb)Tb^{-1}dy$$

onde β é o "Roll-off" do filtro cosseno levantado.

Calculemos ,por facilidade, o caso $\beta = 0$:

$$\sigma_{eq}^2 = KT/Tb\{2/R_{in} + 0,794/gmR_{in}^{-2} + 1,589/gm(2\pi c_{in}/Tb)^2 2^{-4}\}$$

Na prática o termo em R_{in}^2 pode ser desprezado, resultando:

$$\sigma_{eq}^2 \simeq KT/Tb\{2/R_{in} + 5,4 \times 10^{-2}/gm(2\pi c_{in}/Tb)^2\}$$

E finalmente, mantendo a mesma resistência de entrada para os dois codificadores(melhor caso), resulta

$$\theta = \sigma_{1eq}^2/\sigma_{2eq}^2 = r_r\{\frac{2/R_{in} + 5,4 \times 10^{-2}/gm(2\pi c_{in}/Tb)^2}{2/R_{in} + 5,4 \times 10^{-2}/gm(2\pi c_{in}/Tb)^2 \times r_r^{-2}}\}$$

Nota-se que o ganho de sensitividade do sistema codificado compõe-se de dois fatores:

1. - Ganho devido a distância mínima $G_1 = \sqrt{d_{min}}$
2. - Perda devido a excesso de ruido de B e B^3 : $G_2 = \frac{1}{\sqrt{\theta}}$

$$G = G_1/G_2 = \sqrt{d_{min}\theta}$$

5.4.7 Ganho para Front-End PIN-Transimpedância

Por limitações tecnológicas a transimpedância do receptor otimizado não pode permanecer a mesma quando aumentamos a taxa de linha em consequência da codificação CLCE [9]. Assim supomos que $R_{f(CLCE)} = R_{f0(CL)} \times r_r$.
Então temos as seguintes expressões para os ruídos:

$$\sigma_{eq1}^2 = 4KT/R_{f0} \times B$$

$$\sigma_{eq2}^2 = \frac{4KTB}{r_r^2 R_{f0}}$$

Assim, $G = \sqrt{d_{min}r_r^2}$

5.4.8 Ganho de codificação para FRONT-END a APD

$$G = (d_{min}\frac{\sigma^2_{1eqon} + \sigma^2_{1eqoff}}{\sigma^2_{2eqon} + \sigma^2_{2eqoff}})^{1/2} = (d_{min}\theta)^{1/2} \tag{5.8}$$

Nesse caso o ruído não é estacionário e pode-se dividi-lo:

$$\sigma^2_{eq} = \sigma^2_T + \sigma^2_{SH}$$

onde,σ^2_T é a potência de ruído estacionário e σ^2_{SH} é a potência de ruído shot não estacionário.

$\sigma^2_{eqon}(M)$ é o Ruído total equivalente na entrada do receptor para amostra de bit 1 (e é função do ganho de avalanche M):

$$1/M^2(\sigma^2_T + 2qid_sB_{eq}) + 2q(< ion > +id_b)M^xB_{eq} \tag{5.9}$$

$\sigma^2_{eqoff}(M)$ = é o Ruído total equivalente na entrada do receptor para amostra de bit 0 :

$$1/M^2(\sigma^2_T + 2qid_sB_{eq}) + 2q(< ioff > +id_b)M^xB_{eq} \tag{5.10}$$

onde,

M é o Ganho de avalanche

q é a carga elementar

x é o parametro de ruído do APD

id_s é a corrente de escuro superficial

id_b é a corrente de escuro volumétrica

B_{eq} é a Banda equivalente de ruído ($1/2T_b$ para o filtro de Nyquist cosseno levantado)

Trabalhemos por conveniência com taxa de extinsão nula ou equivalentemente $< i_{off} >= 0$. Sabemos que existe um ponto de ganho ótimo (M_{ot}) no qual temos a melhor sensitividade para o receptor; equaciona-se esse ponto[2]:

$$q(< i_{on} > +id_b)M^x_{ot}B_{eq} = \frac{\sigma^2_T + 2qid_sB_{eq}}{M^2_{ot}} \tag{5.11}$$

$$\sigma^2_{eqon}(M_{ot}) = (2/x+1)(\frac{\sigma^2_T + 2qid_sB_{eq}}{M^2_{ot}}) \tag{5.12}$$

das equações (5.12) e (5.10), resulta:

$$\sigma^2_{eqon}(M_{ot}) + \sigma^2_{eqoff}(M_{ot}) = (2/x+2)(\frac{\sigma^2_T + 2qid_sB_{eq}}{M^2_{ot}}) + 2qid_bM^x_{ot}B_{eq} \tag{5.13}$$

$$\theta = \frac{(2/x+2)/M^2_{1ot}(\sigma^2_{1T} + 2qi_{ds}B_{1eq}) + 2qid_bM^x_{1ot}B_{1eq}}{(2/x+2)/M^2_{2ot}(\sigma^2_{2T} + 2qi_{ds}B_{1eq}r_r^{-1}) + 2qid_bM^x_{2ot}B_{1eq}r_r^{-1}} \tag{5.14}$$

das equações (5.14) e (5.11) e sabendo-se que $< i_{2on} >= G < i_{1on} >= (d_{min}\theta)^{1/2} < i_{1on} >$ onde $< i_{1on} >= R \times s_{ens}$(responsividade $\times$ sensitividade do receptor para sistema com CL e dada uma taxa de erro), resulta:

$$< i_{2on} >= (d_{min}\theta)^{1/2} R \times s_{ens} \tag{5.15}$$

$$\theta = \frac{(2/x+2)(\frac{\sigma_{1T}^2+2qid_sB_{1eq}}{xq(Rs_{ens}+id_b)B_{1eq}})^{\frac{-2}{x+2}}(\sigma_{1T}^2+2qid_sB_{1eq})+2qid_bB_{1eq}(\frac{\sigma_{1T}^2+2qid_sB_{1eq}}{xq(Rs_{ens}+id_b)B_{1eq}})^x}{(2/x+2)(\frac{\sigma_{2T}^2+2qid_sB_{1eq}r_r^{-1}}{xq(Rs_{ens}(d_{min}\theta)^{1/2}+id_b)B_{1eq}r_r^{-1}})^{\frac{-2}{x+2}}(\sigma_{2T}^2+2qid_sB_{1eq}r_r^{-1})+2qid_bB_{1eq}r_r^{-1}(\frac{\sigma_{2T}^2+2qid_sB_{1eq}r_r^{-1}}{xq(Rs_{ens}+id_b)B_{1eq}r_r^{-1}})^x} \tag{5.16}$$

Nota-se que para cada taxa de erro de operação no sistema CL temos os parâmetros necessários para resolver a equação transcedental (5.16) em θ e determinar o ganho para uso de CLCE.

Para sistemas que usem APDs muito bons, podemos supor $id_b \simeq id_s \simeq 0$ e da equação (5.16) resulta:

$$\theta = r_r^{-2\frac{(2+x)}{(2-x)}} \times d_{min} \times s^{2\frac{(x+4)}{(2-x)}} \tag{5.17}$$

onde $s = \sigma_{1T}^2/\sigma_{2T}^2$ cujo valor depende do tipo de pré HZ ou TZ. Nota-se que supondo o APD ideal com $x \rightarrow 0$ além de $id_b \simeq id_s \simeq 0$, resulta:

$$\theta \rightarrow r_r^{-2} d_{min} s^4 \tag{5.18}$$

$$G \rightarrow d_{min} s^2 / r_r \tag{5.19}$$

Observa-se que se o ruído estacionário fosse de densidade igual e branca para o CL e o CLCE , o APD ideal dobraria o ganho de codificação em relaçao ao receptor sem APD e atingiria o ganho suave para sistemas coerentes PSK AWGN($r_r d_{min}$). Isso clarifica que o fenômeno da dependência do ruído do APD diretamente com a potência ótica incidente reforça o ganho por codificação de controle de erros.

5.4.9 Perdas de ganho de codificação devido ao efeito de dispersão

Quando o comprimento da fibra é suficientemente longo a dispersão impõe penalidade de sensitividade ao sistema da figura 5.1 devido ao aumento de interferência entre símbolos no amostrador por alargamento dos pulsos óticos. Gimlett e Cheung [7] analisam os sistemas a LED /fibra monomodo tipo "depressed cladding" e as penalidades devido ao fechamento de olho de amostragem devido a dispersão. Eles nos dão os seguintes resultados para regime de baixa dispersão e receptor a PIN: P_D

é a Penalidade de sensitividade por dispersão $\simeq 22B^2\sigma_t^2$ (em dB) ,onde

σ_t^2 = Largura rms da resposta impulsiva da fibra.

B = Taxa de transmissão.

Como penalidade no ganho de codificação (com decodificação) abrupta, podemos considerar[1]

$$P_G = P_D \times (r_r^{-2} - 1) \tag{5.20}$$

A equação (5.20) indica que sob regime de dispersão, o ganho de codificação (abrupto) é facilmente perdido pois a penalidade varia quadraticamente com a taxa relativa de codificação entre o CL e o CLCE.

5.5 Ganho de codificação em sistemas com decodificação abrupta

Na decodificação abrupta(Hard decision), os pulsos são quantizados em apenas dois níveis, ou seja, a decisão é feita bit a bit. O resultado, derivado do cálculo das capacidades para canais AWGN [8], compara decodificação abrupta e suave e indica a diferença assintótica de 2dB do ganho suave para o abrupto. Assim, no nosso sistema podemos aproximar uma diferença referencial de 1dB considerado o fato de estarmos transformando potência em corrente no fotodetetor.Valores mais exatos para esse ganho foram obtidos por simulação, como mostramos adiante.

5.6 Simulação computacional

5.6.1 Introdução

Usando o modelamento exposto ao longo do capítulo , simulamos em micro computador tipo PC AT sistemas de comunicações óticas que empregassem CLs e CLCEs[10]. Procurou-se fazer a comparação de desempenho entre um CL 3B-4B de faixa mínima [11] e um CLCE coset balanceado 4B-15B com $d_{min} = 7$ e derivado do CCEL BCH(15,5,7) de matriz geradora

$$G = \left| \begin{array}{cccccccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{array} \right|$$

cuja escolha deveu-se principalmente a facilidade de implementação na simulação.

5.6.2 O sistema de simulação

Seguimos a filosofia de simulação de sistemas de comunicações óticas exposta por Iramina, Silva e Ribeiro [12],ou seja, sistema implementado em linguagem "C" de forma

[1] A penalidade denota a perda do ganho de codificação em dB devido a dispersão

modular. Cada bloco sistêmico é representado em um arquivo ou módulo, por exemplo: módulo codificador 4B-15B (figura 5.5), módulo decodificador suave(figura5.6), módulo decodificador abrupto (figura 5.7) , etc.

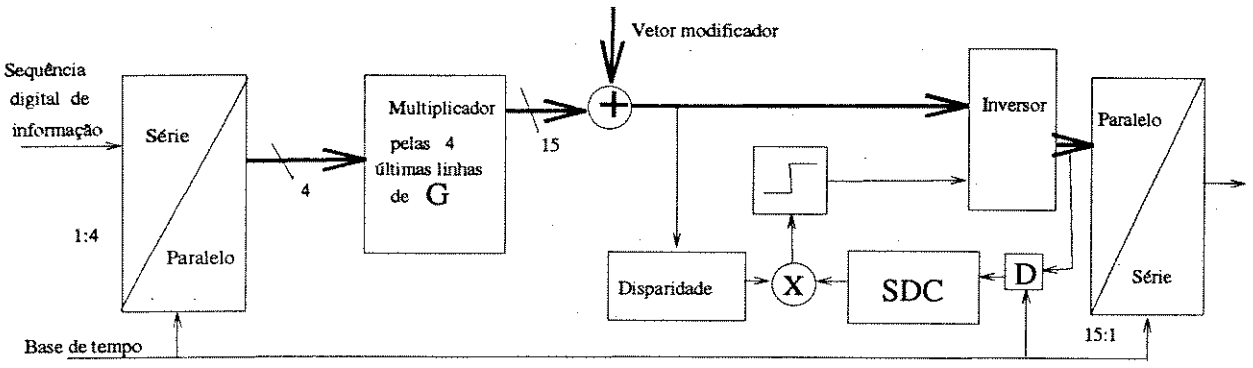


Figura 5.5: Esquema do módulo codificador 4B-15B

A principal característica do sistema é a padronização das interfaces de comunicação entre os módulos com objetivo de permitir o desenvolvimento sistemático de biblioteca de modelos. A montagem dos módulos é feita num programa principal onde são definidos parâmetros e fluxo de dados no sistema.

5.6.3 Simulações

Nas simulações fixou-se a taxa de tributário em 140Mbit/s que é usual em equipamentos comerciais.Usou-se como fonte LEDs ultra rápidos.[7]

5.6.4 Determinação do espectro dos códigos e suas frequências de corte inferior

A opção pelo cálculo espectral "exato" através do algorítmo de Cariolaro e Tronca[13] foi substituida pela "simulação propriamente dita"; por ser de menor complexidade de implementação haja vista a grande quantidade (30) de estados do codificador. Além disso, um parâmetro importante que procuramos é a frequência de corte inferior e sua estimativa é independente da exatidão nos detalhes espectrais. O diagrama da figura 5.8 mostra a configuração do sistema de simulação. O espectro resultante para o CLCE 4B-15B sem vetor modificador[2] apresenta-se na figura 5.9. O número de amostras processadas foi de 1.474.560 e a frequência de corte estimada em 1,15MHz. O espectro para o mesmo CLCE com vetor modificador 2AAA(hexa) apresenta-se na figura 5.10. O número de amostras processadas foi de 1.064.960 e a frequência de corte estimada em 5,7MHz. O espectro do código 3B-4B de faixa mínima apresenta-se na figura 5.11. O número de amostras processadas foi de 819.200 e a frequência de corte estimada em 10MHz.

[2] Uso de vetor modificador conforme ítem 4.3.5 do capítulo 4

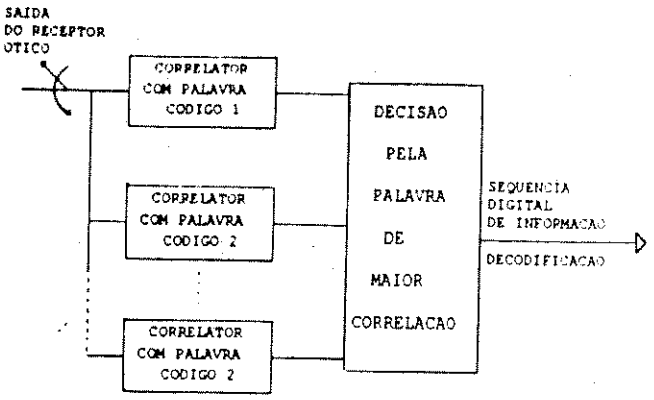


Figura 5.6: Esquema do módulo decodificador suave 4B-15B

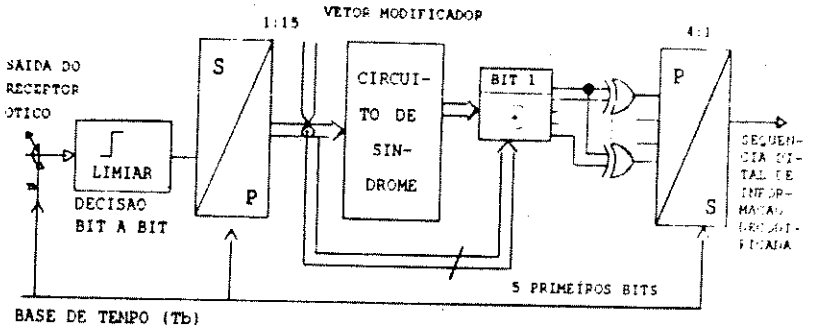


Figura 5.7: Esquema do módulo decodificador abrupto 4B-15B

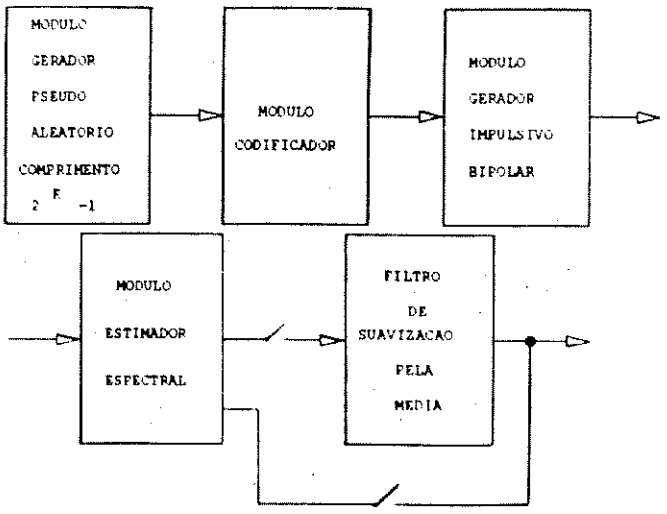


Figura 5.8: Diagrama em bloco do sistema de estimação espectral

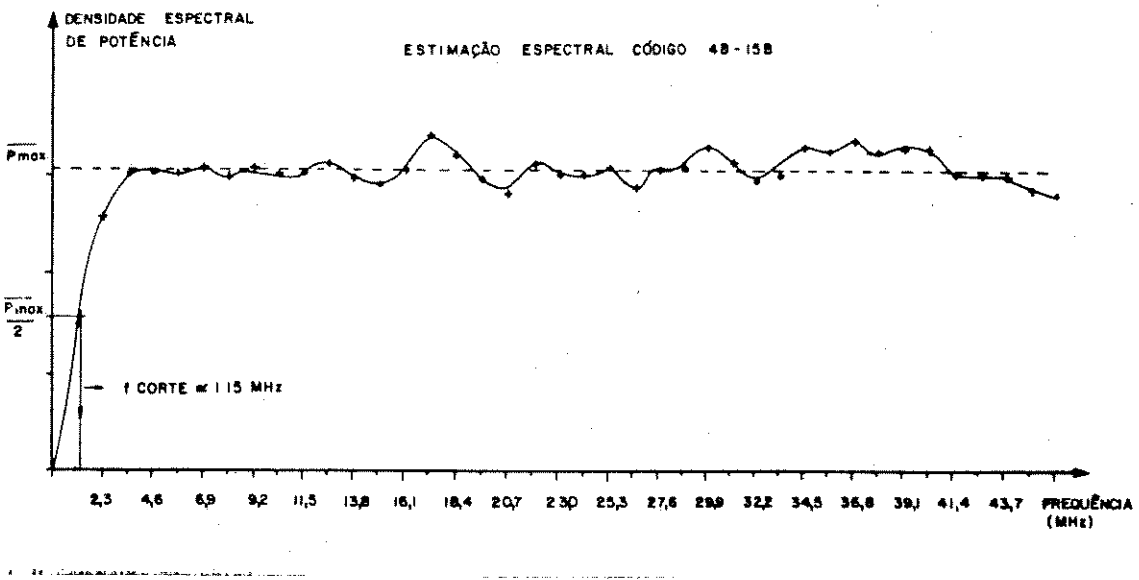


Figura 5.9: espectro do código 4B-15B sem modificador

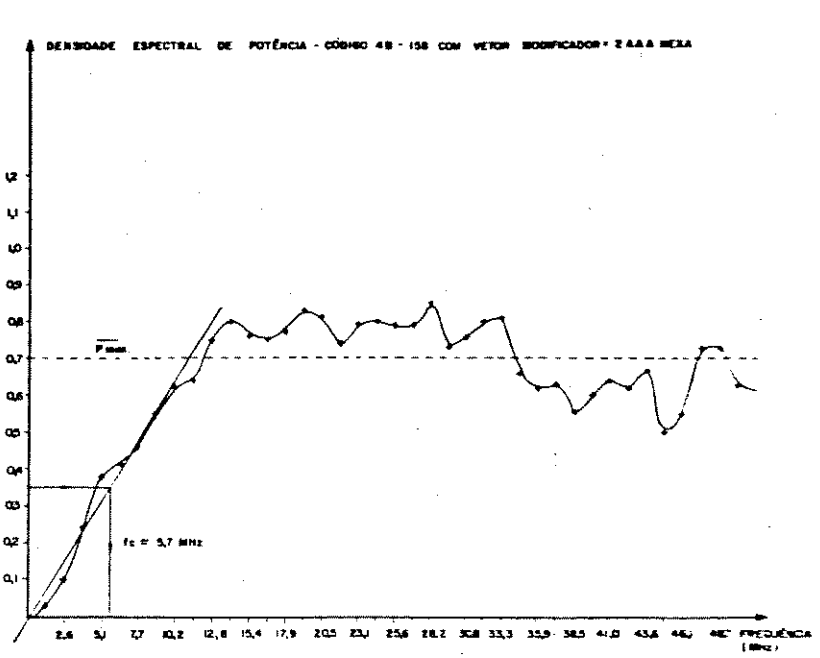


Figura 5.10: espectro do código 4B-15B com modificador

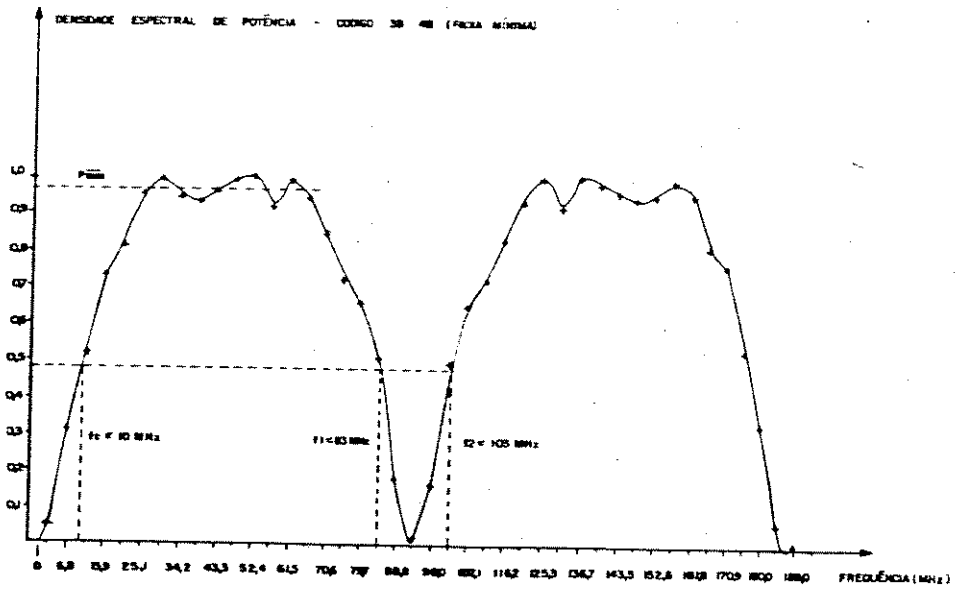


Figura 5.11: espectro do código 3B-4B de faixa mínima

5.6.5 Curvas de taxa de erro X potência acoplada com comprimento de fibra nulo

O simulador foi configurado conforme mostra a figura 5.12 e tem por principais parâmetros: comprimento de fibra(L), tempos de subida e descida do LED (Ts,Td) (modelamento de primeira ordem), largura espectral rms do LED (σ) , comprimento de onda central do LED(λ_c), atenuação na fibra em dB/Km (α), coeficientes de modelamento de atraso de dispersão material da fibra, potência acoplada na fibra pela fonte (P_acoplada),período de bit (Tb), comprimento de sequência pseudo aleatória, "Roll-off" do filtro cosseno levantado(β),limiar de decisão na decodificação abrupta (LIM) , parâmetros do receptor, etc. A seguir expomos os resultados da simulação

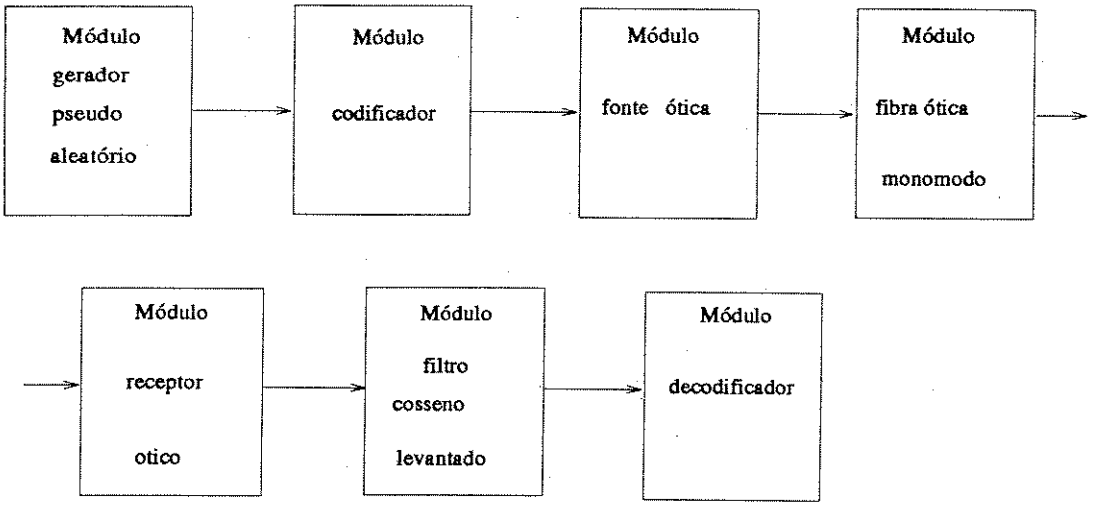


Figura 5.12: Configuração do simulador do sistema de comunicações óticas

para sistemas com os parâmetros comuns: $\lambda_c = 1,3\mu m$, β=0,5, temperatura=300K, Ts=Td=0; aplicados aos diversos tipos de receptores óticos:

- Receptor a PIN com densidade espectral de de ruído igual para os dois códigos comparados: Essa hipótese, de se manter fixa a densidade espectral, é irreal para os receptores óticos básicos conforme visto na teoria ao longo do capítulo. Porém serve como base otimista quanto aos ganhos de codificação. Os parâmetros de receptor usados foram:LIM=0,5, eficiência quântica do PIN (η)=0,8, Resistência equivalente de ruído= 21 $K\Omega$. As curvas resultantes apresentam-se na figura 5.13.

- Receptor PIN-Transimpedância: Os parâmetros do receptor são: LIM=0,5, η=0,8, Resistência de transimpedância 3B-4B= 21KΩ. As curvas resultantes apresentam-se na figura 5.14

- Receptor PIN-Alta impedância: Os parâmetros do receptor são: LIM=0,5, η=0,8,resistência de alta impedância = 300 KΩ, transcondutância do elemento ativo(FET de canal 1μm)= 0,04 $\mho$, capacitância total de entrada= 2pF. As curvas resultantes apresentam-se na figura 5.15.

- Receptor APD InGaAs - Transimpedância: Os parâmetros de receptor são: LIM=0,4, $\eta = 0{,}7$, $Id_s = 20\eta$A, $Id_b = 3\eta$A, transimpedância 3B-4B=21KΩ, ganho APD 3B-4B=8, ganho APD 4B-15B=16 (os ganhos de APD são ótimos para taxa de erro 10^{-9}). Os resultados apresentam-se na figura 5.16.

- Receptor APD InGaAs - Alta impedância : Os parâmetros de receptor são: LIM=0,4 , parâmetros de APD idem ao anterior, resistência de alta impedância= 300KΩ, ganho ótimo de APD 3B-4B = 5, ganho ótimo de APD 4B-15B=11. Os resultados apresentam-se na figura 5.17.

- Receptor APD Ge - transimpedância: Os parâmetros do receptor são: LIM=0,4, transimpedância 3B-4B=21KΩ, parâmetro de excesso de ruído $x = 0{,}7$, $Id_s = 100\eta$A, $Id_b = 5\eta$A, ganho ótimo APD 3B-4B=8, ganho ótimo APD 4B-15B=10. Os resultados apresentam-se na figura 5.18.

- APD Ge- Alta impedância: Os parâmetros do receptor são: LIM=0,4 , parametros de APD idem ao anterior, resistência de alta impedância= 300KΩ, transcondutância do elemento ativo = 0,04 Ʊ,ganho ótimo de APD 3B-4B = 5, capacitância total de entrada= 2pF. Os resultados apresentam-se na figura 5.19.

5.6.6 Curvas comprimento de fibra X Potência ótica acoplada para taxa de erro 10^{-9}

O receptor do sistema simulado foi do tipo PIN com mesma densidade espectral para os dois códigos, dada pela resistência equivalente de ruído de 21KΩ. A fibra modelada é de"dispersão deslocada" operante em $1{,}5\mu m$. Os demais parâmetros são: α=0,2 dB/Km, σ=60ηm, λ_c=1,5μm,Ts=Td=1ηs, $\eta = 0{,}7$, comprimento da sequência pseudo aleatória = $2^7 - 1$, precisão na fase do relógio = 22°, temperatura = 300 K. Os resultados apresentam-se na figura 5.20.

5.7 Conclusões

5.7.1 Resultados da teoria

A teoria prevê pequenos ganhos de CLCEs(ou até perdas) para esse tipo de sistema. Vários fatores contribuem para a degradação do ganho de codificação se compararmos com sistemas usuais a CCEL como radio PSK (AWGN), a saber:

a - Excesso de ruído de B^3 para FRONT-END alta impedância.

b - Queda pela metade do ganho em dB pelo efeito de transformação potência ótica-corrente elétrica.

c - Densidade de ruído aumenta linearmente com a banda para receptores transimpedância.

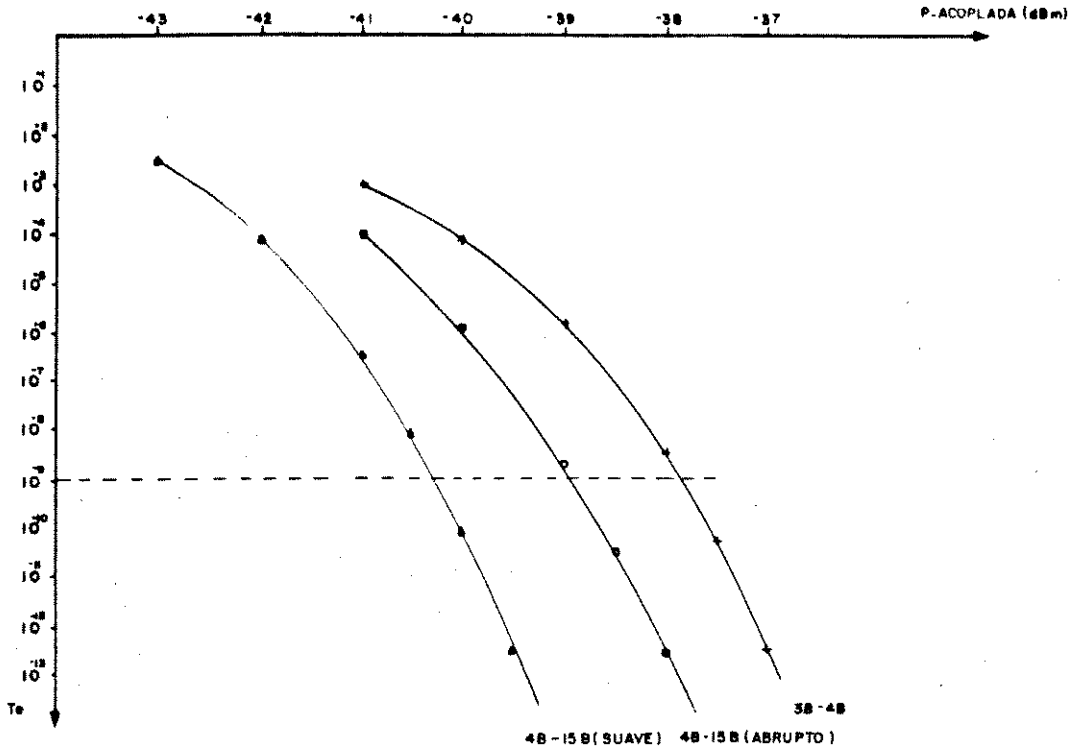


Figura 5.13: Receptor PIN com densidade espectral de ruido fixa

Porém o componente APD favorece ao ganho de codificação, sobretudo quando ele tem baixas correntes de escuro e parâmetro de excesso de ruído pequeno.

5.7.2 Resultados da simulação

- Os resultados da teoria são confirmados pela simulação que corrige a teoria ao acrescentar o efeito da I.E.S. devido a característica da função transferência total (não Nyquist).
- Embora saibamos que o estado da arte da tecnologia ofereça a melhor solução para sistemas limitados por atenuação com o uso de amplificadores óticos.Para efeito de engenharia , se adotarmos a solução CLCE então deve-se usar CLCEs com taxa relativa o mais próximo possível de 1.
- Os resultados mostrados pela figura 5.20, embora baseados em hipótese não real de densidade espectral, poderiam estar indicando melhorias em sistemas não equalizantes limitados por dispersão, quando empregarmos CLCEs mais eficientes; porém a decodificação suave não é tecnologicamente viável em sistemas de alta velocidade.

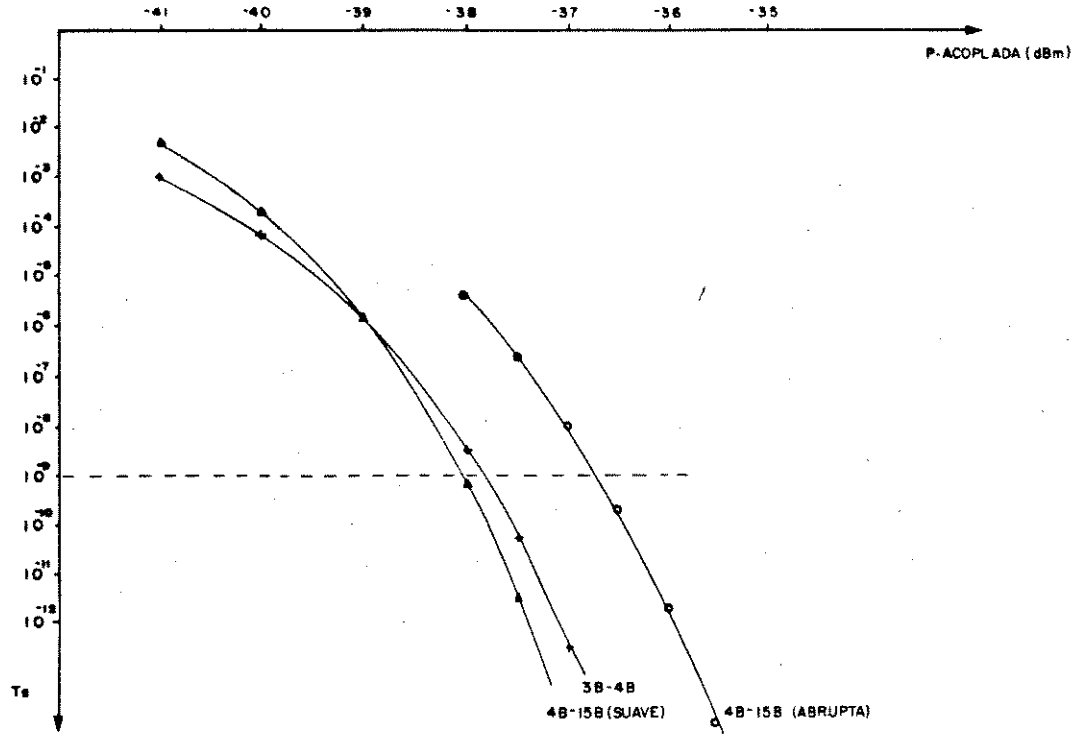


Figura 5.14: Receptor PIN - Transimpedância

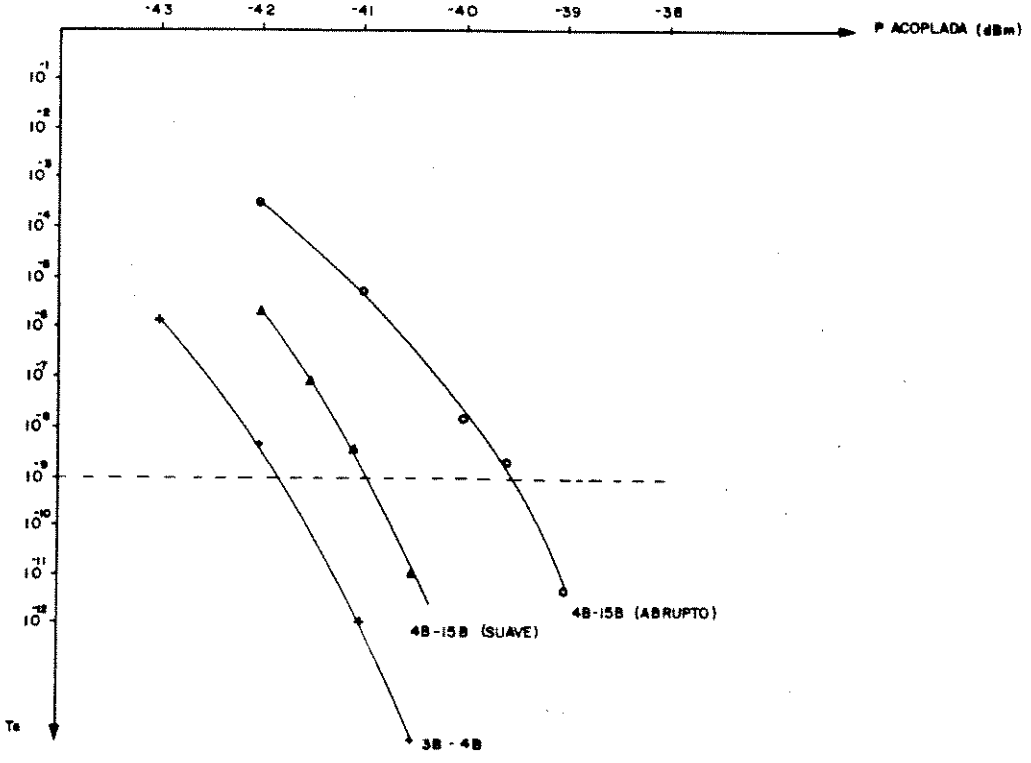


Figura 5.15: Receptor PIN - Alta impedância

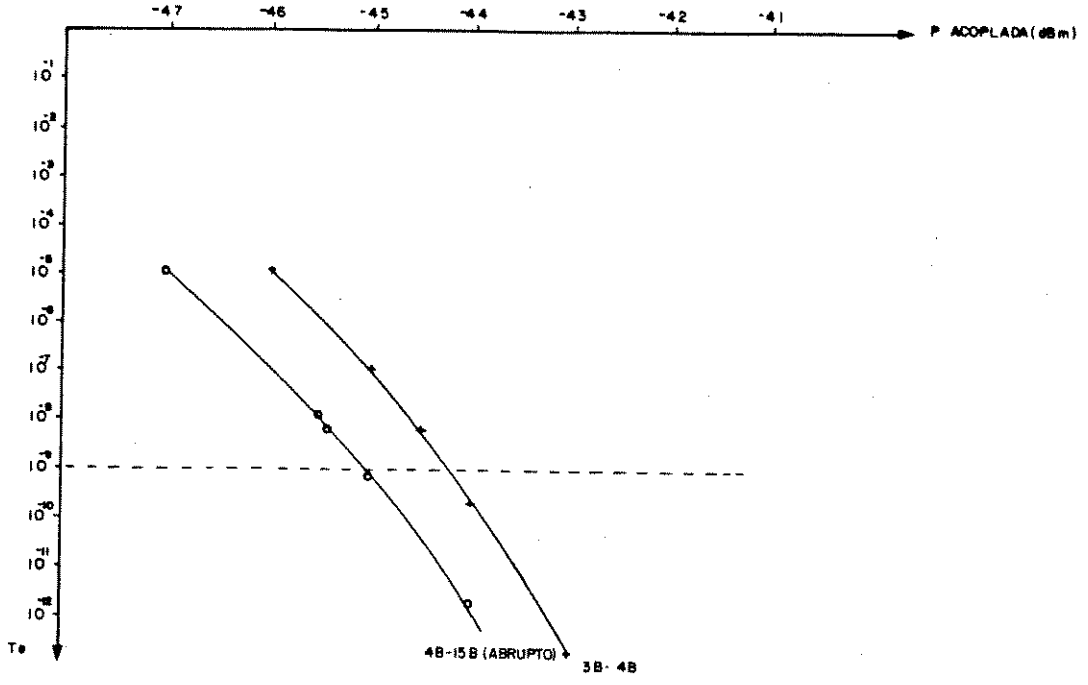


Figura 5.16: Receptor APD InGaAs - Transimpedância

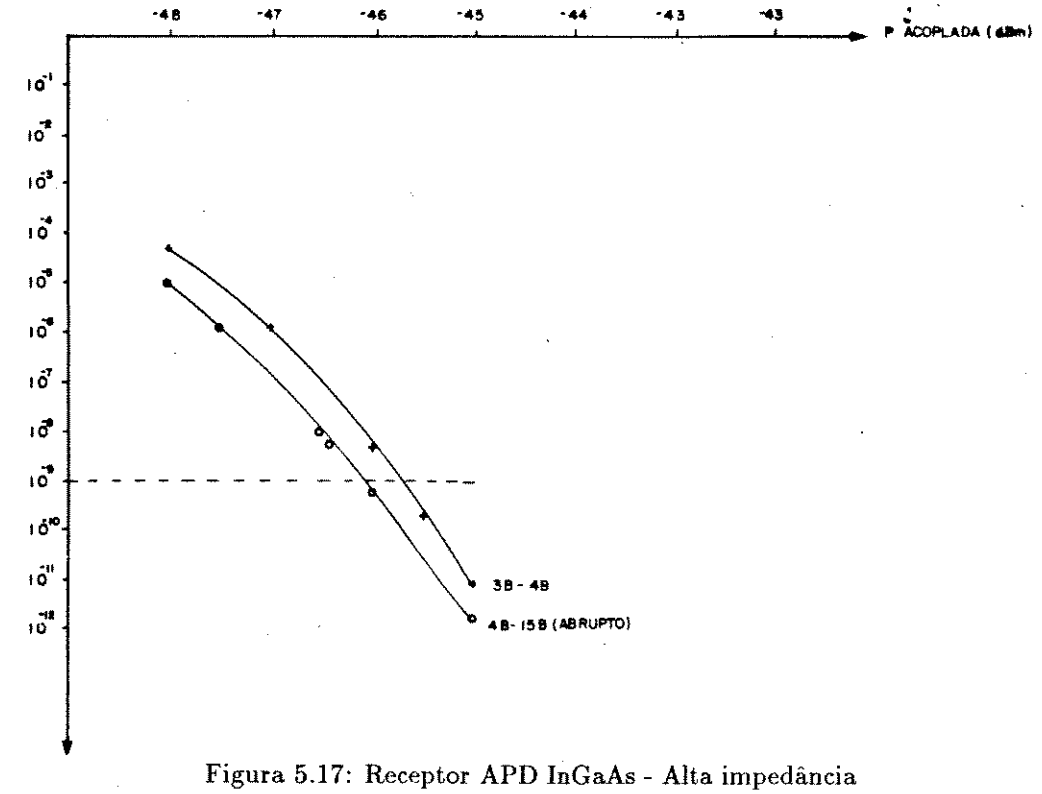


Figura 5.17: Receptor APD InGaAs - Alta impedância

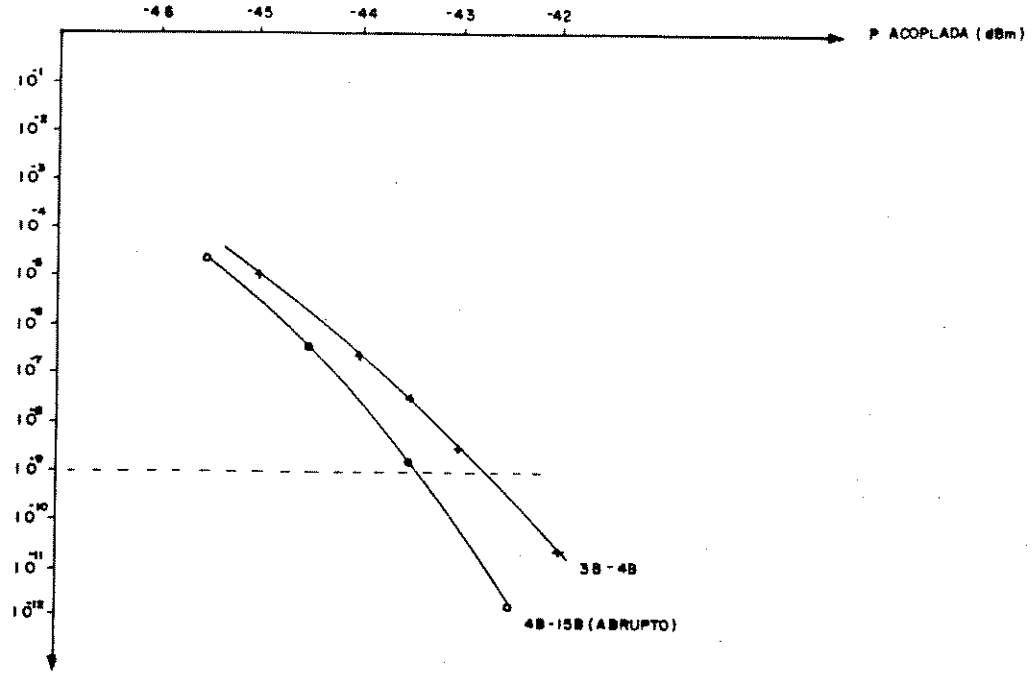


Figura 5.18: Receptor APD Ge - Transimpedância

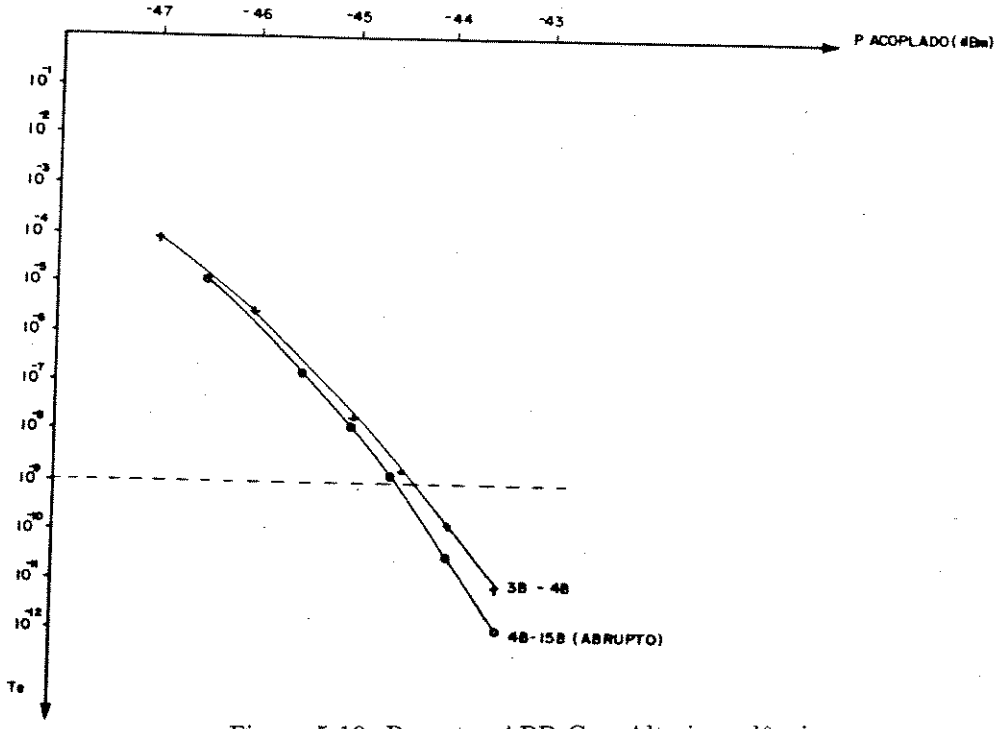


Figura 5.19: Receptor APD Ge - Alta impedância

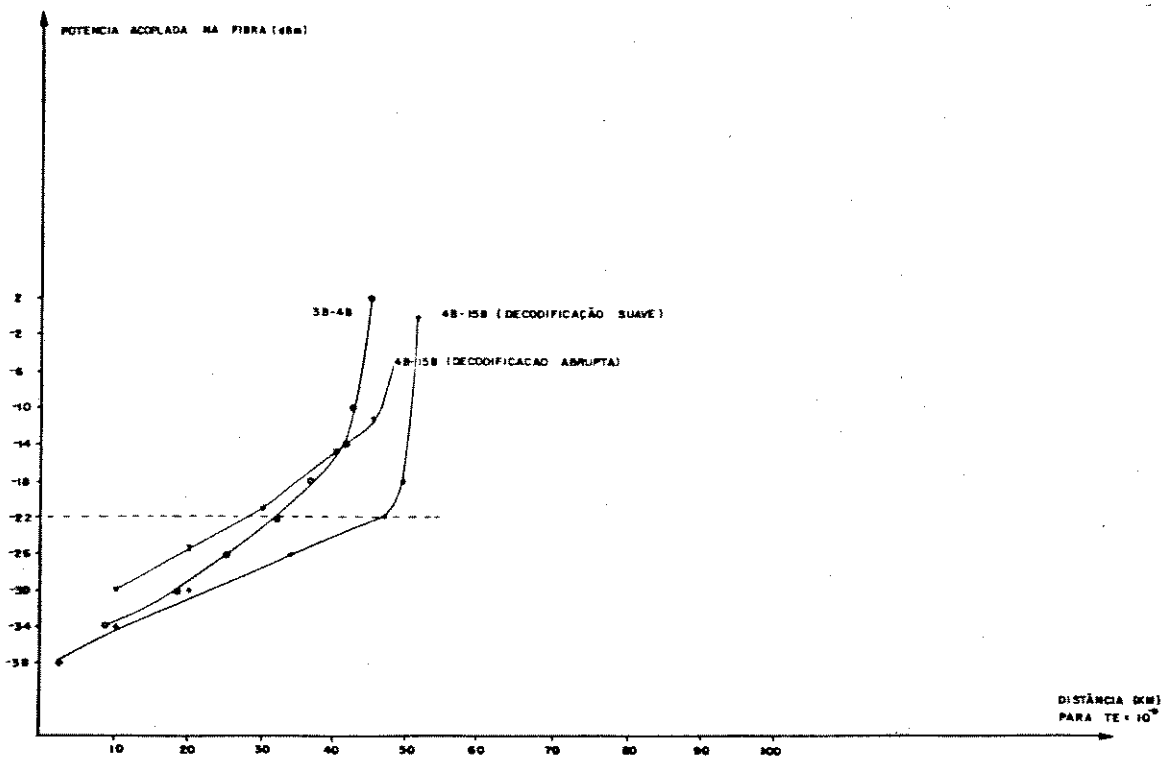


Figura 5.20: Comprimento de fibra X Potência ótica aclopada para PER=10^{-9}

Bibliografia

[1] Personick,S.D.:"Optical fiber transmission systems". Plenum Press, New York, 1981.

[2] Keiser,G.:"Optical fiber communications". Mc Graw-Hill,1983.

[3] Duff,D.G.:"Computer-aided design of digital lightwave systems", IEEE Select. Areas Commun.; vol. SAC-2, pp.171-185, janeiro 1984.

[4] Personick,S.D.:"Statistics of a general class of avalanche detectors with applications to optical communication", BSTJ,vol.50,dezembro 1971.

[5] McIntyre,R.J.:"Multiplication noise in uniform avalanche diode", IEEE trans. Elec. Dev., ED-13, janeiro 1966.

[6] Smith.D.R. & Garret,I.:"A simplified approach to digital optical receiver design", Opt. Quantum Electron., vol.10,pp.211-221,1978.

[7] Gimlett,J.L. & Cheung, N.K.:"Dispersion penalty analysis for LED/single-mode fiber transmission systems", Journal of Lightwave Tec.,vol LT-4,setembro 1986.

[8] Proakis,J.G:"Digital communications". Mc Graw-Hill, Singapure, 1989.

[9] Hullet,J.L. & Muoi,T.V.:"A feedback receive amplifier for optical transmission systems", IEEE Trans. on Comm., outubro 1976.

[10] Nisenbaum,E. & Waldman,H.:"Análise de desempenho de um CLCE em sistemas de comunicações óticas", Anais do 9o SBT,pp.3.3.1-3.3.5 .

[11] Alcântara,H.J.M:"Códigos de bloco binários de faixa mínima para comunicações via fibra ótica", FEE/Unicamp, julho 1988.

[12] Iramina,M.M;Dilva,J.A;Ribeiro,R.F.G.S:"Simulação de sistemas de comunicações óticas", CPqD/Telebras.

[13] Cariolaro,G.L. & Tronca,G.P:"Spectra of block coded digital signals", IEEE Trans. on communications, vol.com-22, outubro 1974.

Capítulo 6

Considerações finais

6.1 Conclusões

O trabalho de busca exaustiva computacional descrito no capítulo 2, gerou códigos que mostram as influências simultâneas da restrição VSDC e de distância mínima de Hamming no excesso de velocidade de CLCEs com comprimentos de bloco pequenos ($n < 30$). Os códigos "úteis" assim gerados não são implementáveis devido à dificuldade de decodificação(memória grande de decodificação). Porém usando os mesmos algorítmos em conjunto com a idéia de modulação codificada de blocos, no capítulo 4, chegamos à construção de CLCEs "fortemente" restritos em VSDC com capacidade de correção de 1 erro que usam memórias possíveis de se implementar. No mesmo capítulo citamos métodos algorítmicos/algébricos baseados em CCELs; esses métodos são bem práticos embora gerem , na maior parte das vezes, restrições VSDC frouxas. No capítulo 3 a análise de centros de esferas de Hamming no espaço DC restrito levou-nos a identificar:

1- Os padrões maximizantes e minimizantes para o caso de esfera de raio 2.

2- Os padrões assintoticamente maximizantes e minimizantes para m ímpar.

3- Os padrões assintoticamente maximizantes para m par.

Quanto ao centro assintóticamente minimizante para $m \geq 6$ par, cabe conjecturar que seja um padrão caótico(aperiódico). Nesse mesmo capítulo chegou-se a expressões para pisos e tetos na taxa de CLCE $\times$ distância mínima relativa por meio de análise combinatória de erros a partir dos centros maximizantes e minimizantes. Em particular obtivemos a expressão geral para pisos no caso de m par.

A análise de desempenho de CLCEs em sistemas de comunicações óticas de deteção direta (Cap. 5) , levou-nos a concluir a importância de se trabalhar com alta taxa de codificação para se obter ganhos.

Finalmente, quanto a comparação dos CLCEs e sistemas que usam concatenação , podemos nos ater a um exemplo para clarear os compromissos existentes entre: taxa de codificação, comprimento de bloco e complexidade de implementação. Usando a classe dos CLCEs fortemente restritos em VSDC (cap. 4) obtemos um

código (usando a partição da figura 4.4) com VSDC=4, $d_{min} = 4$, $n = 40$ e $r = 0,575$. Usando concatenação de um CL $(6,4)$, VSDC=4, e um CCEL Reed-Solomon [1] com símbolos em $GF(2^4)$ e capacidade de correção de 1 símbolo, formamos o código $n = 90$, $d_{min} = 3$ e $r \simeq 0,58$. Nesse exemplo vemos que embora o comprimento de bloco do código concatenado seja bem maior que o do CLCE, a sua implementação acaba sendo mais simples. Além disso, há um tipo diferente de correção de erros entre os dois sistemas: o CLCE corrige apenas 1 bit errado por bloco de 40 bits, enquanto que o sistema concatenado corrige 1 símbolo errado num bloco de 15 símbolos (onde o símbolo em questão corresponde à palavra do CL).

6.2 Sugestões para trabalhos futuros e comentários finais

Um trabalho na mesma linha de pesquisa seguido na tese poderia ser feito para códigos com restrições simultâneas de distância mínima e de comprimento de símbolos corridos (RLL). Nesse caso poderiamos tentar fazer uso das técnicas construtivas de "state splitting" [2] em conjunto com a restrição de distância mínima. Além disso, os resultados de centros maximizantes/minimizantes e distâncias típicas podem ser extendidos facilmente para restrições tipo nulo espectral na frequência de Nyquist ou até , mais trabalhosamente, para outros nulos espectrais. Por exemplo, se invertermos um rótulo sim outro não na treliça X_3 caimos no caso do código EMM(even mark modulation)[3] que tem nulo na frequência de Nyquist, sendo que essa transformação preserva a distância de Hamming.

Finalmente um trabalho de pesquisa em CLCEs próprios para correção de erros em "rajadas" seria de grande utilidade nas áreas de gravação ótica e magnética.

Bibliografia

[1] Lin,S. & Costello,D.J.:"Error control coding: Fundamentals and applications",Prentice Hall, 1983.

[2] Marcus,B.H.;Sigel,P.H.; Wolf,J.K.:" Finite-state modulation codes for data storage", IEEE J. on Selected Areas in communications, vol. 10, janeiro 1992.

[3] Karabed,R. & Siegel,P.H: "Matched spectral-null codes for partial-response channels", IEEE Trans. on Inf. Theory, vol.27, maio 1991.

6.3 Apêndice B - programas de busca exaustiva computacional

```
(* PROGRAMA PARA GERACAO DE CODIGOS COM DISTANCIA MINIMA E VARIACAO DA
   SOMA DIGITAL CORRIDA *)

PROGRAM CODRESTR(INPUT,OUTPUT);

(* DECLARACAO VARIAVEIS GLOBAIS *)

TYPE  VETOR=ARRAY[UINT(0)..UINT(65000)] OF UNSIGNED;

VAR
    SEMENTE,PRATUAL,PRFUTURO:VETOR;
    ARVENDINV,ARVALOR:ARRAY[UINT(0)..UINT(530000)] OF UNSIGNED;
    (*CODIGO:ARRAY[UINT(1)..UINT(10),UINT(1)..UINT(1000)] OF UNSIGNED;*)
    SEMF,L,I,J,K,W,M,PIATUAL,PFATUAL,PIFUTURO,PFFUTURO,NIVEL,
    NIVELA,AUXEND,CNRA,CNRF,PCODI,NUMCOD,
    N,DMIN,NUMCAM:UNSIGNED;
    DMAX,D:INTEGER;
    EXPANDIU:BOOLEAN;
    PF:TEXT;

(* FUNCAO QUE CALCULA DISTANCIA ENTRE 2 PALAVRAS DE COMPRIMENTO N*)

FUNCTION DIST(P1,P2,N:UNSIGNED):UNSIGNED;
VAR I,D,
    J:UNSIGNED;


BEGIN
 D:=0;
 J:=1;
 P1:= UXOR(P1, P2);
 FOR I:=1 TO N DO
  BEGIN

    IF  UAND (P1,J)<>0  THEN D:=D+1;
    J:=J*2;
   END;
  DIST:=D;
 END;




(* FUNCAO QUE CALCULA DISPARIDADE DE UMA PALAVRA,POS.=TRUE *)

FUNCTION DISP(N,I:UNSIGNED):BOOLEAN;
VAR K,J:UNSIGNED;

BEGIN
 DISP:=FALSE;
(* K:=0;
 FOR J:=0 TO 15 DO
  IF (I SHL J)<0 THEN K:=K+1;
 IF (K>=(N+1)DIV 2) THEN DISP:=TRUE
END;*)
IF DIST(I,0,N)>=((N+1) DIV 2) THEN DISP:=TRUE;
END;

(* FUNCAO QUE DETERMINA SE A PALAVRA ESTA DENTRO DOS LIMITES DE
   BALANCEAMENTO *)

FUNCTION LIMITADA(I:UNSIGNED):BOOLEAN;
VAR
   BAL:BOOLEAN;
```

```
   DISPC:INTEGER; (* DISPARIDADE CORRIDA *)
   J,K :UNSIGNED;

BEGIN
 DISPC:=0;
 BAL:=TRUE;
 J:=1;
 K:=1;
 REPEAT
  IF UAND(J,I) <>0 THEN DISPC:=DISPC+1
                   ELSE DISPC:=DISPC-1;
  IF (DISPC>D) OR (DISPC < -DMAX-D) THEN BAL:=FALSE;
  J:=J*2;
  K:=K+1;
 UNTIL (NOT BAL) OR (K>N);
 IF BAL THEN
            IF NOT((DISPC<=0 ) AND (DISPC >= -DMAX)) THEN BAL:=FALSE;
 LIMITADA:=BAL;
END;



(* FUNCAO QUE CALCULA O FATORIAL DE UM INTEIRO *)

FUNCTION FACT(N:UNSIGNED):REAL;

BEGIN
 IF N=0 THEN FACT:=1
        ELSE FACT:=N*FACT(N-1);
END;



(* SUBROTINA GERADORA DE PALAVRAS INICIAIS DE BUSCA=SEMENTES DE DISPARIDADE
   POSITIVAS
   OBS: N <=15 PARA O MICROPC *)

PROCEDURE GERASEMENTES(N:UNSIGNED);
VAR J,I:UNSIGNED;
    K:REAL;

BEGIN
 SEMF:=0;
 J:=1;
 K:= EXP(N*LN(2));
 FOR I:=0 TO ROUND(K-1) DO
  IF LIMITADA(I) THEN
                  BEGIN
                   SEMENTE[J]:=I;
                   SEMF:=SEMF+1;
                  (*  WRITELN(SEMENTE[J]);*)
                   J:=J+1;
                  END;
END;


(* PROGRAMA PRINCIPAL *)
BEGIN
 N:=5;
 D:=1;
 DMAX:=1;
 IF LIMITADA(UINT(%X'00000011')) THEN WRITELN('BALANCEADA')
                else WRITELN('NAO BALANCEADA');
(* INICIALIZACAO DE VARIAVEIS *)
```

```
NIVELA:=0;
NUMCOD:=0;
OPEN(PF,'ARQCONT.SAIDA');
REWRITE(PF);

(* ENTRADA DE DADOS *)


WRITELN(' ENTRE COM TAMANHO DO BLOCO ');
READLN(N);
WRITELN(PF,'TAMANHO DE BLOCO= ',N:6);
WRITELN(' ENTRE COM DISTANCIA MINIMA ');
READLN(DMIN);
WRITELN(PF,'DISTANCIA MINIMA= ',DMIN:6);
WRITELN('ENTRE COM EXCESSO DE VSDC');
READLN(D);
WRITELN(PF,'EXCESSO DE VSDC= ',D:6);
WRITELN('ENTRE COM MAXIMA DISPARIDADE TERMINAL');
READLN(DMAX);
WRITELN(PF,'MAXIMA DISPARIDADE TERMINAL= ',DMAX:6);
WRITELN(PF);
WRITELN(PF);


(* GERACAO DE SEMENTES *)
GERASEMENTES(N);
WRITELN('NUMERO DE SEMENTES GERADAS= ',SEMF);

(* LOOP PRINCIPAL PARA GERACAO DE CODIGOS APARTIR DE CADA SEMENTE *)
FOR L:=1 TO SEMF DO
 BEGIN
 NIVEL:=1;
  (* PREENCHIMENTO DO NIVEL 1 DA ARVORE *)
  PIATUAL:=1;
  PFATUAL:=0;
  FOR K:=L+1 TO SEMF DO
   IF DIST(SEMENTE[L],SEMENTE[K],N)>=DMIN THEN
      BEGIN
       PFATUAL:=PFATUAL+1;
       ARVALOR[PFATUAL]:=SEMENTE[K];
       ARVENDINV[PFATUAL]:=0;
      END;
  IF PFATUAL >= PIATUAL THEN  (* SE FORAM ENCONTRADAS PALAVRAS NO NIVEL 1 *)
   BEGIN
      NIVEL:=2;
      CNRA:=2;  (* INICIALIZACAO DO CONTADOR DE NUMERO DE GRUPOS DE RAMOS *)
      PRATUAL[1]:=1;
      PRATUAL[2]:=PFATUAL+1;
     (*INICIALIZACAO DE APONTADORES PARA GRUPOS DE RAMOS *)
      (* LOOP DE PREENCHIMENTO DE ARVORE *)
      REPEAT
        PIFUTURO:=PFATUAL+1;
        PFFUTURO:=PFATUAL;
        CNRF:=1;
        PRFUTURO[1]:=PFFUTURO+1;
        FOR I:=1 TO CNRA-1 DO
         FOR J:=PRATUAL[I] TO ((PRATUAL[I+1])-1) DO
         BEGIN
          EXPANDIU:=FALSE;
          FOR W:=J+1 TO PRATUAL[I+1]-1 DO
          IF DIST(ARVALOR[W],ARVALOR[J],N) >= DMIN THEN
           BEGIN
            PFFUTURO:=PFFUTURO+1;
```

```
            EXPANDIU:=TRUE;
            ARVALOR[PFFUTURO]:=ARVALOR[W];
            ARVENDINV[PFFUTURO]:=J;
           END;
           IF EXPANDIU THEN
            BEGIN
             CNRF:=CNRF+1;   (* PREENCHIMENTO DE VETOR DE CONTROLE DE GRUPOS
                                                    DE RAMOS CRIADOS *)
             PRFUTURO[CNRF]:=PFFUTURO+1;
            END;
          END;
        IF PIFUTURO <= PFFUTURO THEN
         BEGIN
          PIATUAL:=PIFUTURO;
          PFATUAL:=PFFUTURO;
          FOR M:=1 TO CNRF DO
           PRATUAL[M]:=PRFUTURO[M];
          CNRA:=CNRF;
          NIVEL:=NIVEL+1;
         END;
      UNTIL PIFUTURO > PFFUTURO;
     (* PREENCHIMENTO DOS MAIORES CODIGOS ENCONTRADOS NO ARQUIVO DE SAIDA*)

      NUMCAM:= PFATUAL-PIATUAL +1;
      WRITELN(PF,'SEMENTE= ',L:6,' NUMERO DE CODIGOS= ',NUMCAM:6);
      WRITELN(PF,'NIVEL= ',NIVEL:6);
      WRITELN(PF);
      WRITELN(PF);
      IF NIVEL >= NIVELA THEN
       BEGIN
        NIVELA:=NIVEL;
        PIATUAL:=PIATUAL-1;
        J:=1;
        FOR I:=1 TO NUMCAM DO
         BEGIN
          WRITELN(PF);
          PIATUAL:=PIATUAL+1;
          AUXEND:=PIATUAL;

          for j:=1 to nivela do
          begin
           IF AUXEND<>0 THEN WRITE(PF,ARVALOR[AUXEND]:6,' ')
                        ELSE WRITE (PF,SEMENTE[L]:6,' ');
            AUXEND:=ARVENDINV[AUXEND]
           END;
          WRITELN(PF);
          END;
        END
        ELSE WRITELN(PF,'NAO VAI SER IMPRESSO POIS O CODIGO EH PEQUENO');
        writeln(PF);
  END;
 END;
CLOSE(PF);
END.
```

```
PROGRAM CODRESUNICO(INPUT,OUTPUT);

(* DECLARACAO DE VARIAVEIS GLOBAIS *)

TYPE  VETOR=ARRAY[UINT(1)..UINT(20000)] OF UNSIGNED;
VAR
    SEMENTE:ARRAY[UINT(1)..UINT(1300000)] OF UNSIGNED ;
    SEMIN:ARRAY[UINT(1)..UINT(10)] OF UNSIGNED;
    CODIGO:VETOR;
    SEMF,L,I,J,K,W,M,PIATUAL,PFATUAL,PIFUTURO,PFFUTURO,NIVEL,
    NIVELA,AUXEND,CNRA,CNRF,PCODI,NUMCOD,
    N,DMIN,NUMCAM,CONTSEMIN:UNSIGNED;
    EXPANDIU,MEMOVER,COLETA:BOOLEAN;
    PF:TEXT;
    D,DMAX,AUX:INTEGER;

(* FUNCAO QUE CALCULA DISTANCIA ENTRE 2 PALAVRAS DE COMPRIMENTO N*)

FUNCTION DIST(P1,P2:UNSIGNED):UNSIGNED;
VAR I,D,
    J:UNSIGNED;


BEGIN
 D:=0;
 J:=1;
 P1:= UXOR(P1, P2);
 FOR I:=1 TO N DO
  BEGIN

    IF UAND(P1,J)<>0  THEN D:=D+1;
    J:=J*2;
   END;
  DIST:=D;
 END;


(* FUNCAO QUE DETERMINA SE A PALAVRA ESTA DENTRO DOS LIMITES DE
   BALANCEAMRNTO *)

FUNCTION LIMITADA(I:UNSIGNED):BOOLEAN;
VAR
   BAL:BOOLEAN;
   DISPC:INTEGER; (* DISPARIDADE CORRIDA *)
   J,K:UNSIGNED;

BEGIN
 DISPC:=0;
 BAL:=TRUE;
 J:=1;
 K:=1;
 REPEAT
  IF UAND(J,I) <> 0 THEN DISPC:=DISPC+1
                    ELSE DISPC:=DISPC-1;
  IF (DISPC >D) OR (DISPC <-DMAX-D) THEN BAL:=FALSE;
  J:=J*2;
  K:=K+1;
 UNTIL (NOT BAL) OR (K>N);
 IF BAL THEN
          IF NOT((DISPC <= 0 ) AND (DISPC >= -DMAX)) THEN BAL:=FALSE;
 LIMITADA:=BAL;
END;



(* FUNCAO QUE CALCULA DISPARIDADE DE UMA PALAVRA,POS.=TRUE *)
```

```
FUNCTION DISP(I:UNSIGNED):BOOLEAN;
VAR K,J:UNSIGNED;

BEGIN
 DISP:=FALSE;
IF DIST(I,0)>=((N+1) DIV 2) THEN DISP:=TRUE;
END;


(* FUNCAO QUE CALCULA O FATORIAL DE UM INTEIRO *)

FUNCTION FACT(N:UNSIGNED):REAL;

BEGIN
 IF N=0 THEN FACT:=1
        ELSE FACT:=N*FACT(N-1);
END;




(* SUBROTINA GERADORA DE PALAVRAS INICIAIS DE BUSCA=SEMENTES DE DISPARIDADE
   POSITIVAS
   OBS: N <=15 PARA O MICROPC *)

PROCEDURE GERASEMENTES(N:UNSIGNED);
VAR J,I:UNSIGNED;
    K:REAL;

BEGIN
 SEMF:=0;
 J:=1;
 K:= EXP(N*LN(2));
 FOR I:=0 TO ROUND(K-1) DO
  IF LIMITADA(I) THEN
                  BEGIN
                   SEMENTE[J]:=I;
                   SEMF:=SEMF+1;
                  (* WRITELN(SEMENTE[J]);*)
                   J:=J+1;
                  END;
END;


(* PROGRAMA PRINCIPAL *)

BEGIN
(*OPEN(PF,'ARQUNICO.SAIDA');
REWRITE(PF);
WRITELN(PF);*)

(* ENTRADA DE DADOS *)


(*WRITELN(' ENTRE COM TAMANHO DO BLOCO ');
READLN(N);
WRITELN(' ENTRE COM DISTANCIA MINIMA ');
READLN(DMIN);*)

DMIN:=4;
D:=5;
DMAX:=0;
FOR AUX:=11 TO  12  DO
BEGIN
  N:=2*AUX;
```

```
 (* GERACAO DE SEMENTES *)

 OPEN(PF,'ARQUNICO2.SAIDA');
 REWRITE(PF);
 GERASEMENTES(N);
 WRITELN(PF,'N= ',N:6);
 WRITELN(PF,'SEMF= ',SEMF:6);
 WRITELN(PF,'D= ',D:6);
 WRITELN(PF,'DMAX= ',DMAX:6);
 WRITELN(PF,'DMIN= ',DMIN:6);
 FOR W:=1 TO 10 DO
     SEMIN[W]:=0;
 SEMIN[1]:=SEMENTE[1];
 FOR CONTSEMIN:=1 TO 2 DO
 BEGIN
  IF SEMIN[CONTSEMIN]<>0 THEN
  BEGIN
   COLETA:=TRUE;
   NIVEL:=1;
   CODIGO[1]:=SEMIN[CONTSEMIN];

   (* LOOP PRINCIPAL PARA GERACAO DE CODIGOS APARTIR DE CADA SEMENTE *)

   FOR L:=1 TO SEMF DO
    BEGIN
     EXPANDIU:=TRUE;
     FOR J:=1 TO NIVEL DO
      IF DIST(SEMENTE[L],CODIGO[J])< DMIN THEN EXPANDIU:=FALSE;
     IF EXPANDIU THEN BEGIN
                        NIVEL:=NIVEL+1;
                        CODIGO[NIVEL]:=SEMENTE[L]
                       END
                       ELSE IF COLETA THEN
                          BEGIN
                            FOR W:=1 TO CONTSEMIN DO
                              IF SEMENTE[L]= SEMIN[W] THEN COLETA:=FALSE;
                            IF COLETA THEN BEGIN
                                           SEMIN[W+1]:=SEMENTE[L];
                                           COLETA:=FALSE
                                           END
                                      ELSE COLETA:=TRUE;
                          END;
    END;
    WRITELN(PF,'SEMENTE= ',SEMIN[CONTSEMIN]:7);
    WRITELN(PF,'NIVEL= ',NIVEL:6);
    WRITELN(PF);
 END;
END;
CLOSE(PF);
  (* FOR J:=1 TO NIVEL DO

  WRITELN(CODIGO[J]);*)
END;
(*CLOSE(PF);*)
END.
```

```
 program codalf(output);

(* declaracao de variaveis globais *)

type  vetor = array[  1..  20000] of integer32  ;
var
    semente1:array[  1..  130000] of integer32   ;
    semente2:array[  1..  130000] of integer32   ;
    semente3:array[  1..  130000] of integer32   ;
    semin:array[  1..  10] of integer32  ;
    codigo:vetor;
    semf1,semf2,semf3 ,l,i,j,k,w,m,piatual,pfatual,pifuturo,pffuturo,nivel3,
    nivela,auxend,cnra,cnrf,pcodi,numcod,nivel1,nivel2,
    n,dmin,numcam,contsemin:integer32  ;
    expandiu,memover,coleta:boolean;
    pf:text;
    d,dmax,aux:integer32  ;

(* funcao que calcula distancia entre 2 palavras de comprimento n*)

function dist(p1,p2:integer32  ):integer32  ;
var i,d,
    j:integer32  ;


begin
 d:=0;
 j:=1;
 p1:= xor(p1, p2);
 for i:=1 to n do
  begin

    if land(p1,j)<>0  then d:=d+1;
    j:=j*2;
   end;
  dist:=d;
 end;


(* funcao que determina se a palavra esta dentro dos limites de
   balanceamrnto - alfabeto regressivo *)

function limitada(i,d,dmax:integer32  ):boolean;
var
   bal:boolean;
   dispc:integer32  ; (* disparidade corrida *)
   j,k:integer32  ;

begin
 dispc:=0;
 bal:=true;
 j:=1;
 k:=1;
 repeat
  if land(j,i) <> 0 then dispc:=dispc+1
                    else dispc:=dispc-1;
  if (dispc >d) or (dispc <-dmax-d) then bal:=false;
  j:=j*2;
  k:=k+1;
 until (not bal) or (k>n);
(* descarta alfabeto progressivo *)
 if bal then
          if  (dispc > 0) then bal:=false;
 limitada:=bal;
end;
```

```
(* funcao que calcula disparidade de uma palavra,pos.=true *)

function disp(i:integer32  ):boolean;
var k,j:integer32  ;

begin
 disp:=false;
if dist(i,0)>=((n+1) div 2) then disp:=true;
end;

(* funcao que deteta palavra de disparidade nula *)

function dispnul(i:integer32):boolean;

begin
dispnul:=false;
if  (n mod 2) = 0 then
      if dist(i,0)= (n div 2) then dispnul:=true;
end;

(* funcao que calcula o fatorial de um inteiro *)

function fact(n:integer32  ):real;

begin
 if n=0 then fact:=1
        else fact:=n*fact(n-1);
end;



(* subrotina geradora de palavras iniciais de busca=sementes de disparidade
   positivas
   obs: n <=15 para o micropc *)

procedure gerasementes(n:integer32  );
var j,i:integer32  ;
    k:real;

begin
 semf1  :=0;
 j:=1;
 k:= exp(n*ln(2));
 for i:=0 to round(k-1) do
  if limitada(i,d,dmax) then
                  begin
                   semente1[j]:=i;
                   semf1  :=semf1  +1;
                   j:=j+1;
                  end;
end;


(* programa principal *)

begin

(* entrada de dados *)

open(pf,'arqalf.saida','new');
rewrite(pf);
for w:= 2 to 11 do
```

```
begin
dmin:=4;
d:=1;
dmax:=0;
n:=2*w;

 (* criacao da palavra auxiliar tudo 1 *)
aux:=1;
for k:=1 to n-1 do
  aux:= lshft(aux,1)+1;



 (* geracao de sementes *)


 gerasementes(n);
 writeln(pf,'n= ',n:6);
 writeln(pf,'semf1  = ',semf1  :6);
 writeln(pf,'d= ',d:6);
 writeln(pf,'dmax= ',dmax:6);
 writeln(pf,'dmin= ',dmin:6);

  (* uniao ordenada dos alfabetos progressivos e regressivos no vetor semente3[] *)
semf3:=0;
for j:=1 to semf1 do

   (* colocar primeiro as palavras de disparidade 0 cuja complementar esteja
num mesmo alfabeto e tenha vsdc=2 *)

  if  dispnul(semente1[j]) then
         begin
                  coleta:=false;
                if limitada(semente1[j],1,0) then coleta:=true;
                if semf3<>0 then
                 for k:=1 to semf3 do
                  if semente1[j]=semente3[k] then coleta:=false;
                if coleta then
                              begin
                      semf3:=semf3+2;
                        semente3[semf3-1]:=semente1[j];
                        semente3[semf3]:=xor(aux,semente1[j]);
                               end;
          end  ;
writeln('semf3=',semf3);

for j:=1 to semf1 do

   (* colocar primeiro as palavras de disparidade 0 cuja complementar esteja
num mesmo alfabeto *)

  if  dispnul(semente1[j]) then
         begin
                  coleta:=false;
                if limitada(semente1[j],d,0) then coleta:=true;
                if semf3<>0 then
                 for k:=1 to semf3 do
                  if semente1[j]=semente3[k] then coleta:=false;
                if coleta then
                              begin
                      semf3:=semf3+2;
                        semente3[semf3-1]:=semente1[j];
                        semente3[semf3]:=xor(aux,semente1[j]);
                               end;
          end  ;
```

```
writeln('semf3=',semf3);


for j:=1 to semf1 do

   (* colocar depois as palavras de disparidade 0 complementares de alfabetos
diferentes *)
  if  dispnul(semente1[j]) then
          begin  coleta:=true;
                  if semf3<>0 then
                   for k:=1 to semf3 do
                    if semente1[j]=semente3[k] then coleta:=false;
                  if coleta then begin
                        semf3:=semf3+2;
                          semente3[semf3-1]:=semente1[j];
                          semente3[semf3]:=xor(aux,semente1[j]);
                                  end;
           end  ;
writeln('semf3=',semf3);
  (* colocar depois palavras nao balanceadas complementares *)
for j:=1 to semf1 do
  if not (dispnul(semente1[j])) then
          begin   coleta:=true;
                 for k:=1 to semf3 do
                    if semente1[j]=semente3[k] then coleta:=false;
                  if coleta then begin
                        semf3:=semf3+2;
                          semente3[semf3-1]:=semente1[j];
                          semente3[semf3]:=xor(aux,semente1[j]);
                                   end;
        end;

writeln('semf3=',semf3);
 (* inicializacao da busca com ponto inicial *)

 codigo[1]:=semente3[1];
 nivel3:=1;

   (* loop principal para geracao do codigo *)

   for l:=1 to semf3   do
    begin
     expandiu:=true;
     for j:=1 to nivel3 do
      if dist(semente3[l],codigo[j])< dmin then expandiu:=false;
      if expandiu then begin
                        nivel3:=nivel3+1;
                        codigo[nivel3]:=semente3[l];
                        end;
    end;
(* contagem da interseccao com alfabetos progressivo e regressivo *)
nivel1:=0;
nivel2:=0;
for j:=1 to nivel3 do
  for k:=1 to semf1 do
   begin
     if codigo[j]=semente1[k] then nivel1:=nivel1+1;
     if codigo[j]=xor(aux,semente1[k]) then nivel2:=nivel2+1;
    end;



(* saida de dados *)
```

```
    writeln(pf,'nivel3= ',nivel3:6);
    writeln(pf,'nivel1= ',nivel1:6);
    writeln(pf,'nivel2= ',nivel2:6);
    writeln(pf);
 end;
close(pf);
  (* for j:=1 to nivel do

  writeln(codigo[j]);*)
end.
```