

UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica

Departamento de Comunicações

Janeiro 1994

Este exemplar corresponde à redação final da tese defendida por MARCOS ANTONIO CASSIOLATO BATISTA aprovada pela Comissão Julgadora em 27 01 94.

Hélio Waldman
Orientador

CÓDIGOS DE LINHA PARA SISTEMAS ÓPTICOS DE ALTA CAPACIDADE.

Marcos Antonio Cassiolato Batista ³²²

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 Mestre em Engenharia Elétrica.

UNICAMP
BIBLIOTECA CENTRAL

940537

RESUMO

Foi realizado um estudo sobre várias técnicas de codificação de linha para sistemas ópticos de alta velocidade, nas quais o uso de grandes tabelas memorizadas é problemático. Duas soluções para este problema são discutidas neste trabalho. A primeira é uma versão modificada dos códigos PFmB(m+1)B, com vantagens de desempenho e sem nenhum acréscimo de complexidade.

O segundo esquema de codificação proposto combina técnicas de memorização das palavras-código em tabelas com técnicas algorítmicas, porém com a finalidade de reduzir ao mínimo imprescindível o tamanho destas tabelas para um determinado desbalanço especificado. O compromisso entre o tamanho das tabelas e o balanceamento alcançável é estudado e dimensionado. Finalmente, são estudados outros parâmetros de desempenho, como as propriedades espectrais, a sincronizabilidade e a monitorabilidade da taxa de erros.

ABSTRACT

Alternative coding techniques for high speed optical transmission systems, in which the use of extensive tables in memory is a problem, have been discussed in recent literature.

Two solutions for this problem are discussed in this work. The first is a modified version of the PFM(m+1)B codes, with higher performance and no added complexity.

The second coding method combines traditional codeword assignment techniques with algorithmic techniques. This method minimizes the size of the tables for the specified running digital sum. The compromise between the table size and sequence imbalance is studied and dimensioned. Finally, other performance parameters such as spectral properties, synchronizability, and error rate monitorizability, are studied.

AGRADECIMENTOS

Se fosse enumerar aqui todos a quem minha consciência obriga um sincero agradecimento, seria impraticável a extensão da lista. Portanto, nominalmente citarei alguns, reconhecendo que devo muito a muitos.

Antes de todos, agradeço a Deus pela minha existência e pela de todos.

Ao prof. Dr. Hélio Waldman: orientador criativo, profissional, dedicado; amigo.

Aos membros da banca examinadora por suas sugestões para aprimoração deste trabalho

Aos professores da FEE/UNICAMP, especialmente do DECOM, que encontrei sempre acessíveis e atenciosos.

À CAPES, que auxiliou-me com bolsa de estudos; e à UEL, que incentivou-me através do PICD.

Aos amigos da Universidade Estadual de Londrina pelo apoio e compreensão.

Àqueles que colaboraram para a confecção deste trabalho, principalmente os que desenvolveram os trabalhos manuais, que para mim são muito valiosos.

à minha família

ÍNDICE

Capítulo 1	Introdução	1
Capítulo 2	Códigos para sistemas ópticos de alta velocidade	4
	• Introdução	4
	• Alguns aspectos da codificação de linha	4
	• Códigos binários	7
	• O balanceamento	8
	• Códigos para sistemas ópticos de alta velocidade	9
	• O Diagrama Pantográfico	12
	• O código MPFmB(m+1)B	14
Apêndice 2A	Estudo do número de alfabetos necessários para códigos de blocos mB(m+1)B, m ímpar	26
Apêndice 2B	Estudo da extensão de erro	28
Capítulo 3	Códigos de memória reduzida	31
	• Introdução	31
	• Método para obtenção da tabela de tamanho mínimo	32
	• O código MDVmB(m+1)B	47
Capítulo 4	Propriedades de transmissão	50
	• Introdução	50
	• Análise espectral	50
	• Monitoração da taxa de erros	59
	• Análise de sincronismo	59
Capítulo 5	Conclusões	63
Apêndice A1	Listagem do programa utilizado para análise espectral dos códigos	65
Referências		68

CAPÍTULO 1

Introdução

Atualmente estamos vivendo o que muitos já estão considerando uma nova etapa na história da humanidade, a era pós-industrial, cujos valores, mercadorias, são informações e serviços. Neste sentido, a denominação *information highway* dada ao projeto norte-americano de instalar uma infra-estrutura de comunicações de alta velocidade é muito apropriada, pois serão verdadeiras auto-estradas de dados, através das quais transitarão serviços tão (ou mais) valiosos como aqueles que circulam por estradas e ferrovias.

O Brasil também está se preparando para esta nova era: recentemente foi inaugurado um sistema de comunicações por fibra óptica interligando São Paulo e Rio de Janeiro, sendo que a expansão prevê aproximadamente 15 mil km de fibras instaladas até o início do próximo milênio, conectando as principais cidades do país. Além disso, o Brasil está participando de dois consórcios internacionais que visam a implantação de troncos de comunicação óptica submarinos: o Unisur, interligando países platinos, e o Américas I, que partirá de Fortaleza até a Flórida, passando por vários países [SALLES 92].

O volume crescente de informações e a multiplicidade dos serviços - telefonia, dados, vídeo - arrastaram as taxas de transmissão a valores elevados, da ordem de Gbits/s. Nestas velocidades, códigos de linha convencionais se tornam inadequados. Esta, porém, não é uma das maiores preocupações das redes locais, pois aí as taxas não são tão altas. Os estudos neste campo, muito numerosos no momento, abrangem outras problemáticas. Destacariamos como temas sugestivos, merecedores de atenção, a interconectividade entre redes locais, o

interfaceamento redes locais-troncos de alta velocidade e o desafio da *information highway* de levar alta velocidade ao consumidor elementar, às residências.

Contudo, a preocupação deste trabalho é com os sistemas de alta velocidade. Nesta área de aplicação, não é indiferente o uso de um ou outro esquema de codificação, já que, normalmente, um codificador eleva a taxa de transmissão do sinal fonte e este excesso de velocidade introduzido pelo código deve ser minimizado, assim como o tempo gasto nas operações de codificação e decodificação.

O objetivo que nos propomos é - através do estudo de esquemas de codificação de linha apropriados para altas taxas - contribuir com o projeto de sistemas de transmissão digital por fibras ópticas de alta velocidade, apresentando elementos que avaliem a melhor alternativa e sugerindo novas opções de codificação.

No capítulo seguinte, partindo de uma sucinta revisão sobre aspectos gerais de codificação de linha, analisamos a classe dos códigos binários, concentrando nossa atenção para aqueles utilizados em sistemas ópticos. Neste contexto, alguns dos principais códigos para aplicações em altas taxas de transmissão foram comentados e então, finalizando o capítulo, apresentamos o código MPFmB(m+1)B, próprio para altas velocidades (>1Gbits/s), que elimina as operações de memorização no codificador e no decodificador.

Um aspecto de destaque, tratando-se ainda do capítulo 2, é a caracterização de alguns modernos códigos de linha de alta velocidade numa classe denominada *códigos algorítmicos*: a codificação e a decodificação são efetivadas através de operações elementares, algorítmicas, sobre os bits da palavra de informação e da palavra codificada, respectivamente. O código MPFmB(m+1)B, aqui desenvolvido, enquadra-se nesta classificação.

Outra proposta de codificação deste trabalho é o assunto do capítulo 3. Trata-se do código MDVmB(m+1)B, que mescla operações de codificação e de decodificação algorítmicas, como as do MPFmB(m+1)B, e operações mínimas de memorização em tabelas. A idéia deste estudo germinou quando trabalhávamos no código proposto no capítulo 2 para otimizar seu desempenho. Como os novos desafios sempre têm algo de provocador, ou melhor, de intimidador, fomos em

busca do elo que ligaria os códigos cujas palavras de codificação são todas armazenadas em tabelas e aqueles que não utilizam a memorização de palavras código. Assim, o desenvolvimento do capítulo 3 soube-nos um dos mais empolgantes e gratificantes de todos.

A análise das propriedades de transmissão está inserida no capítulo 4: utilizamos o método desenvolvido por [CARIOLARO 74] para obter o diagrama da densidade espectral de potência dos códigos; discutimos a sincronizabilidade dos códigos MPFmB(m+1)B e MDVmB(m+1)B, bem como a monitorabilidade da taxa de erros.

No capítulo final, o quinto, através dos resultados obtidos nos capítulos antecedentes, apresentamos nossas conclusões a respeito das propostas de codificação aqui desenvolvidas.

O apêndice traz a listagem do programa utilizado para o desenvolvimento deste trabalho, cuja fonte encontra-se também à disposição no banco de software do DECOM da FEE/UNICAMP.

CAPÍTULO 2

Códigos para sistemas ópticos de alta velocidade

• Introdução

A finalidade de um sistema de transmissão é uma comunicação eficiente e confiável. Geralmente, como o sinal da fonte não apresenta certas características convenientes ao meio transmissor, faz-se necessário uma codificação. O papel da codificação de linha é converter os dados da fonte para um formato robusto às deficiências que um meio específico pode apresentar, introduzindo, para isto, uma redundância no sinal a ser transmitido.

• Alguns aspectos da codificação de linha

Os códigos de linha apresentam vantagens tais como a possibilidade da conformação do espectro do sinal transmitido - [CAVERS 92] denomina-os de códigos de modelagem do espectro -; a possibilidade de monitoração da taxa de erros no terminal receptor ou nos receptores regenerativos intermediários; um rico conteúdo espectral, que facilita a extração do sinal de relógio; a possibilidade do uso de canais de serviço em baixa frequência.

A contrapartida destas vantagens são o aumento do nível de complexidade dos equipamentos terminais (transmissores e receptores); a necessidade de uma banda passante maior, devido ao crescimento da taxa de símbolos transmitida, causado pela introdução da redundância; e a conseqüente deterioração da margem de ruído nos regeneradores.

Uma pré-classificação dos códigos de linha poderia ser feita considerando-se os diferentes meios de transmissão. Assim, teríamos os códigos para cabos metálicos, para fibras ópticas, para rádio-enlaces. Poderíamos também distingui-los quanto ao tipo de transmissão: com portadora ou em banda base [CATTERMOLE 83]. Nosso estudo se restringirá aos códigos para sistemas ópticos de transmissão digital em banda base.

Não é um trabalho simples a escolha de um código de linha para um sistema de transmissão. São inúmeros fatores que devem ser levados em consideração, desde a tecnologia do sistema, a topologia da rede, até os fatores econômicos. É uma decisão importante, que repercute diretamente no desempenho do sistema e na complexidade de seus circuitos de regeneração.

Os requisitos normalmente exigidos para os códigos de linha para transmissão digital por fibra óptica são (cfr. [BROOKS 83] e [HOSOE 83]):

- O código deve ser independente da seqüência de bits (BSI - do inglês: Bit Sequence Independent), isto é, não pode impor nenhuma restrição sobre a mensagem a ser transmitida;
- A decodificação deve ser unívoca: cada palavra código deve ser mapeada numa única palavra de saída, de preferência independente do estado do decodificador;
- auto-sincronismo: O decodificador deve ser capaz de identificar a correta estrutura de bits enviada pelo codificador, ou seja, ter a capacidade de, automaticamente e de forma confiável, estar alinhado com o codificador, inclusive na presença de elevadas taxas de erro na linha;

- transmissão de suficiente informação de temporização: O código deve prover um elevado número de transições de forma a garantir a presença constante de uma raia espectral na frequência do sinal de relógio, que será extraído do próprio sinal digital;
- capacidade de detecção de erro para monitorar o desempenho do enlace e, em alguns casos, dos repetidores intermediários também;
- pequeno efeito de multiplicação de erro: dependendo do código, um simples erro na linha pode causar muitos erros na decodificação;
- pouco conteúdo espectral nas baixas frequências para facilitar o acoplamento AC com os repetidores e permitir canais de supervisão em baixa frequência;
- alta eficiência: para manter o menor possível a largura de banda do ruído e a elevação da taxa de símbolos transmitida;
- complexidade de implementação relativamente baixa, quando comparada com a complexidade total do sistema.

O código de linha mais apropriado para um sistema será o que melhor satisfizer as condições próprias do projeto em questão. Porém, a maioria destes nove requisitos anteriores são, geralmente, conflitantes, exigindo uma solução de compromisso entre os mesmos.

Para o caso das linhas ópticas, a grandeza modulada - intensidade óptica - é intrinsecamente positiva. Além disso, as fontes ópticas, utilizadas atualmente em sistemas práticos, sofrem a falta de linearidade no grau desejado para a adoção de três (ou mais) níveis de transmissão, que é uma das técnicas para introdução da redundância sem elevar a taxa de transmissão de símbolos na linha. Estes são alguns dos motivos pelos quais os projetos de comunicação óptica empregam códigos de linha binários, sobre os quais concentraremos nosso estudo.

• Códigos binários

A literatura apresenta uma diversidade de classificações que agrupam os códigos de linha binários de acordo com critérios como regras de codificação, balanceamento do sinal, etc. Analisando as mais clássicas ([CATTERMOLE 83], [BROOKS 83], [YOSHIKAI 86]), constatamos o descrito a seguir.

Os códigos de bloco são aqueles do tipo $mBnB$, onde m bits adjacentes da seqüência do sinal fonte (bits ou palavras de informação) são codificados em palavras-código de n bits adjacentes, sendo $n > m$. Conseqüentemente existe um aumento na taxa de símbolos transmitida, que é multiplicada pela razão $1/E$, onde $E = m/n$ é conhecida como eficiência do código. Para maior eficiência é sempre possível fazer $n = m + 1$, mas geralmente é necessário utilizar dois ou mais alfabetos (vide Apêndice 2A).

Os códigos de bloco que se utilizem de alfabetos (conjunto de palavras-código) dependentes do estado do codificador são conhecidos como *códigos de bloco alfabéticos*; e os que não utilizam, *códigos de bloco não-alfabéticos*. Os códigos $mB1C$ [YOSHIKAI 84] - que agregam um bit complementar ao m -ésimo bit de informação e os $DmB1M$ [YOSHIKAI 86] são exemplos de códigos não alfabéticos e também chamados de códigos de inserção de bits, devido ao modo como operacionalizam a codificação.

Pode-se distinguir dois grupos dentre os códigos de bloco alfabéticos: os que selecionam as palavras código através da consulta a uma tabela de codificação (look up table), exigindo para isso memórias tanto no codificador como no decodificador; e os códigos de bloco alfabéticos algorítmicos, cujas regras de codificação são simples algoritmos de operações elementares, que dispensam o uso de memórias.

Por ser recente o tratamento destes últimos tipos de códigos como "códigos algorítmicos" (cfr. [SLOMINSKI 92], [WALDMAN 92] e [FAIR 91]), começou-se a utilizar a expressão "códigos de bloco alfabéticos com look up table" para o outro grupo de códigos alfabéticos. Neste trabalho, por questão de praticidade, valerão as

seguintes reduções de terminologia: aos códigos de bloco alfabéticos com look up table nos reportaremos simplesmente como códigos alfabéticos; e aos códigos de bloco alfabéticos algorítmicos, como códigos algorítmicos.

Antes, contudo, de continuarmos a análise dos códigos de bloco binários e verificarmos a questão de qual deles é mais apropriado para altas taxas de transmissão, vamos expor alguns conceitos básicos que nos serão úteis em breve.

• O Balanceamento

Pode-se mostrar que o balanceamento do sinal é conseguido mediante o controle da soma digital corrida $SDC(i)$, definida a seguir conforme a recomendação CCITT para seqüências binárias [REC.G702]:

$$SDC(i) = \sum_{j=-\infty}^i g_j \quad (1)$$

onde:

$g_j = +0,5$ se o dígito da seqüência corresponder ao nível lógico "1".

$g_j = -0,5$ se o dígito da seqüência corresponder ao nível lógico "0".

Para que o sinal seja balanceado, ou seja, tenha densidade espectral zero na freqüência zero, é suficiente e necessário que a soma digital corrida seja mantida confinada entre valores finitos [JUSTESEN 82]. A diferença entre os valores máximo e mínimo da SDC é um parâmetro do código conhecido como variação da soma digital corrida, VDC . Quanto menor a VDC , menor é a flutuação pico-a-pico das baixas freqüências do sinal [ROMANO 84] e, conseqüentemente, melhor é seu balanceamento.

Os códigos que apresentam VDC finita são ditos *códigos balanceados*. Apesar de não ser um requisito forte, o projeto de

sistemas ópticos é facilitado com a utilização destes tipos de códigos. Além de facilitar a polarização do laser no transmissor - simplesmente mantendo-se a potência média num valor fixo -, permite utilização de receptores e repetidores com acoplamento AC sem a degradação das flutuações do nível de base e jitter. Por sua vez, os códigos desbalanceados, já que o sinal não pode sofrer processo de integração, impõem o uso de receptores transimpedância, sendo, por isso, menos versáteis.

O confinamento da SDC permite, além das vantagens acima, que se monitore erros de linha nos regeneradores ou nos terminais receptores. Sempre que houver um erro de bit isolado, este provocará uma violação de um dos limites da SDC.

Encerrando este item, vamos apresentar a definição da disparidade de uma palavra a de m bits:

$$d(a) = \sum_{i=1}^m a_i \quad (2)$$

onde:

$a_i = +0,5$ ou $a_i = -0,5$, correspondendo, respectivamente, aos níveis lógicos "1" e "0" dos dígitos da palavra a binária.

• Códigos para sistemas ópticos de alta velocidade

A demanda por serviços digitais de alta capacidade resultaram no desenvolvimento de sistemas de transmissão óptica operando acima de 500 Mbits/s. Para tanto, é desejável e econômico usar um código eficiente pelas seguintes razões [BROOKS 83]:

- com o aumento da velocidade, os circuitos eletrônicos tendem a ser mais difíceis, custosos e demandam maior potência;
- o fator limitante ao aumento de velocidade são os dispositivos eletrônicos e não os ópticos;
- degradação de sensibilidade ao ruído no receptor eletrônico;

- dispersão na fibra, que embora tenha diminuído muito, pode voltar a ser significativa com o uso do amplificador óptico;
- apesar do custo do decodificador ser maior para códigos mais eficientes, este custo é pequeno frente ao custo total do sistema.

Verifica-se, acompanhando o aumento da taxa de transmissão, a adoção de valores crescentes de m nos códigos de bloco $mBnB$ - com $n=m+1$, a fim de melhorar a eficiência: códigos 1B2B (Manchester) são usados para taxas de até 10 Mbits/s; códigos 3B4B, até 50 Mbits/s; códigos 5B6B, até 200 Mbits/s; códigos 7B8B, daí em diante¹ (todos os códigos citados são códigos alfabéticos balanceados, com exceção do Manchester que é não-alfabético, porém balanceado).

Esta evolução trouxe ao seu lado um crescimento exponencial do tamanho das tabelas memorizadas em ROM's para as operações de codificação e decodificação. Acima de 500 Mbits/s, a operação dessas ROM's se torna problemática por causa das dificuldades tecnológicas (cfr. [BROOKS 83] e [FAIR 91]).

Uma abordagem para o problema foi adotada em [YOSHIKAI 84], onde se propôs o código $mB1C$. É um código não-alfabético, desbalanceado, que codifica m bits numa palavra de $m+1$ bits: a palavra código é formada pelos m bits da palavra-fonte mais um bit (bit de inserção ou bit complementar), cujo valor é o complemento do último bit da palavra de informação. Deste modo, evita-se o uso de tabelas e m pode ser tal que a eficiência destes códigos os indiquem para altas taxas de transmissão.

Seguindo nesta mesma linha, surgiram os códigos $DmB1M$ [YOSHIKAI 86], evoluindo no sentido do balanceamento - são balanceados na média -, porém, em sentido estrito, são desbalanceados. Além disso é necessário embaralhamento prévio da seqüência para garantir bons resultados de desempenho como, por exemplo, de temporização.

Como vimos anteriormente, por serem desbalanceados, os códigos $mB1C$ e $DmB1M$ necessitam de receptores do tipo

¹ Devido à relação logarítmica entre a eficiência do código ($E = m/n$) e as perdas [BROOKS 83], há um ponto em que a redução das perdas não justifica o aumento na complexidade da codificação. Para os códigos alfabéticos, não é usual ter um código mais complexo que o 7B8B.

transimpedância e esta configuração é problemática em alta velocidade, por exigir amplificadores eletrônicos de alto ganho.

A abordagem na qual se insere nosso trabalho para se evitar ou minimizar o uso de ROM's é aquela que, não abandonando o balanceamento, busca operações algorítmicas simples sobre os dígitos de informação - daí a denominação "códigos algorítmicos".

Apesar de situado em outro contexto, Carter propôs códigos (que levam seu nome), cuja regra de codificação está baseada na transmissão da própria palavra de informação ou de sua inversa, em função do valor atual da SDC e da disparidade da referida palavra, acrescentado um bit, que conforme sua polaridade, indica a ocorrência da inversão [CARTER 65].

Com o crescimento das taxas de transmissão, os códigos Carter, anteriormente empregados em sistemas PCM a cabos metálicos, mereceram atenção em aplicações ópticas. Baseado nestes códigos e no proposto em [ALI 86], surgiram os códigos PFmB(m+1)B - "partially flipped" mB(m+1)B, ou mB(m+1)B parcialmente rebatido [KRZYMIEN 89].

Utilizando valores de m ímpares, a codificação PFmB(m+1) atua em dois estágios: uma pré-codificação, onde o (m+1)-ésimo bit é inserido com a finalidade de reduzir a disparidade dos m bits da palavra a ser codificada; e a transmissão destes bits, optando pela inversão ou não dos m primeiros bits, no sentido de reduzir - levar a zero - o valor absoluto da SDC.

Nesta mesma direção foram projetados os códigos MDM-mBnB [SLOMINSKI 92] - "minimum disparity and memoryless", com $n=m+1$ - desenvolvidos a partir dos códigos introduzidos pelos autores em [SLOMINSKI 89] e [WYSOCKI 89]. A idéia é uma combinação do esquema de codificação dos códigos de inserção de bits e dos códigos de baixa disparidade.

Uma técnica interessante é a proposta em [FAIR 91], onde se possibilita a conhecida técnica de auto-embaralhamento - que por si só é desbalanceada e não atende ao requisito BSI, sendo classificada como quasi-BSI - a capacidade de guiar-se para produzir uma seqüência balanceada de bits codificados. Os autores denominaram de

GS, "guided scrambling" (embaralhamento guiado), este tipo de codificação, que apresenta as seguintes opções: codificação em bloco, continua ou a combinação de ambas.

Um dos esquemas de codificação aqui propostos surgiu do estudo de uma solução para a degradação da VDC que os códigos $PFmB(m+1)B$ apresentaram - em comparação com os códigos alfabéticos $mBnB$ tradicionais - para conseguir a eliminação total de operações do tipo "table look up". Durante este estudo, apresentou-se-nos um novo panorama: haveria uma relação entre a diminuição de VDC e o número de palavras códigos a serem memorizadas? A resposta a esta pergunta gerou os códigos $MDVmB(m+1)B$ - memória-reduzida dependente de VDC -, nova proposta de codificação para sistemas ópticos de alta velocidade, que apresentaremos no capítulo seguinte.

• O Diagrama Pantográfico

O Diagrama Pantográfico (DP) (ALCÂNTARA, 1988) é uma ferramenta que foi bastante utilizada no desenvolvimento dos esquemas de codificação desenvolvidos neste trabalho e, por isso, merece uma pequena explanação. Sua origem é o Triângulo de Pascal, ao qual foram aplicadas algumas transformações para daí se extrair informações úteis para o projeto e/ou análise de um código de linha [ROMANO 84].

Conforme (1), cada marca (bit "1") acrescenta +0,5 no valor da SDC e cada espaço (bit "0") acrescenta -0,5. No DP a marca e o espaço são representados por um segmento de reta ascendente e descendente, respectivamente. Se o eixo das ordenadas representar a SDC e o eixo das abscissas a duração T de um símbolo da palavra codificada, facilmente pode-se visualizar a disparidade da palavra e, portanto, a transição entre estados terminais da SDC.

Observando a figura 2.1, verifica-se que: a palavra 000010 leva $SDC(0)=0$ a $SDC(6)=-2$, pois a palavra tem disparidade igual a -2; a palavra 110100 (disparidade=0) não altera o estado da SDC.

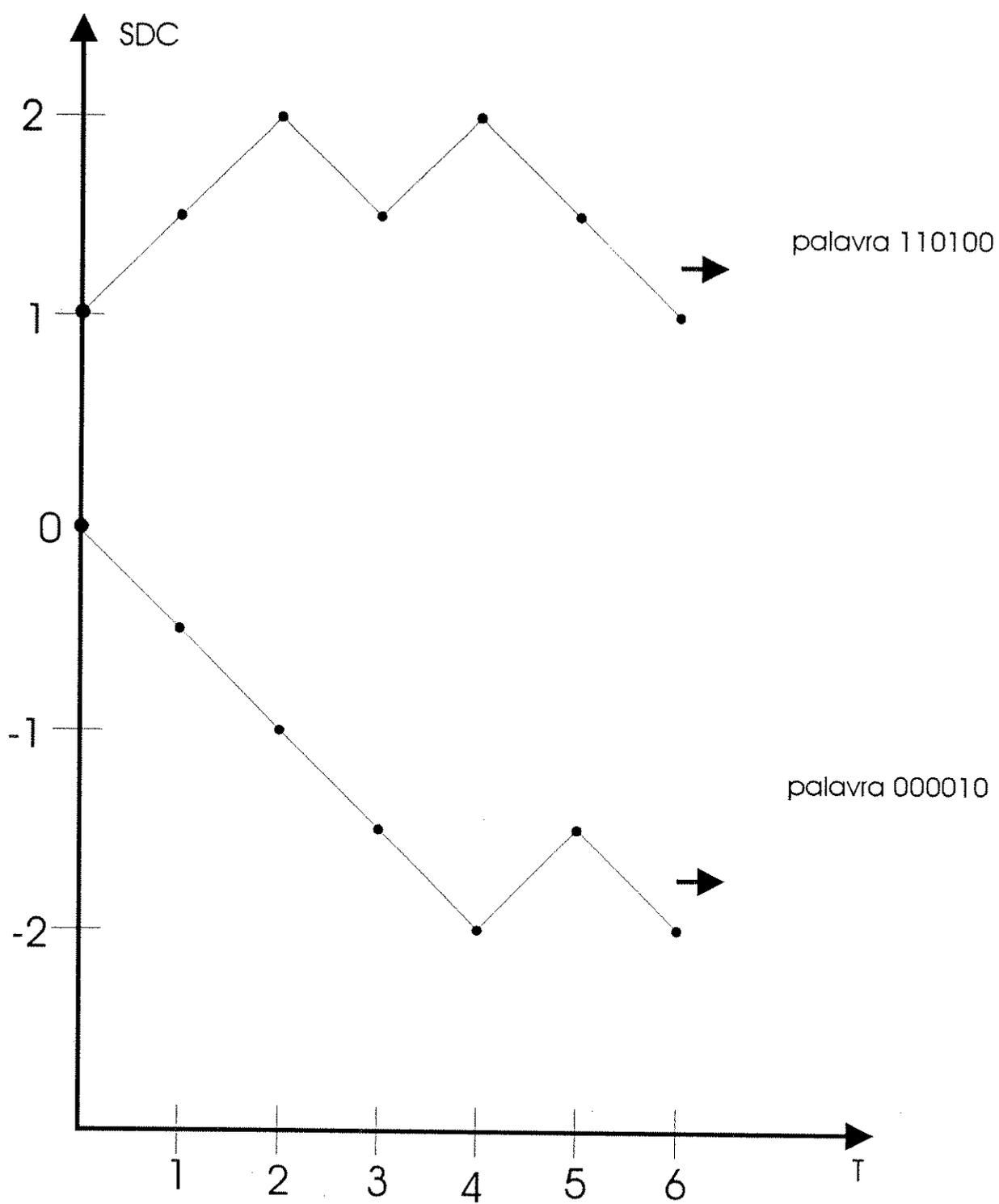


Figura 2.1

Continuando a exemplificação do DP, vamos confinar a SDC e verificar quantas palavras-código de 6 bits poderiam ser mapeadas: as linhas tracejadas da figura 2.2 são as barreiras de confinamento da SDC para obtermos $VDC=3$. Os "nós", que se formam no DP pela sobreposição de todas as possíveis palavras, recebem um valor cuja regra - assim como no Triângulo de Pascal - é: o valor de um nó é a soma dos valores dos nós precedentes, sendo que o valor do primeiro nó é a unidade. Os valores encontrados para os nós terminais significam que existem 14 palavras com disparidade +1, 20 com disparidade nula e 14 com disparidade -1, totalizando 48 palavras úteis. Se pensarmos em códigos $mB(m+1)B$, estas 48 palavras seriam suficientes para codificar palavras-fonte de 5 bits, ou seja, um total de 32 palavras-fonte. Conclusão: para o estado terminal $SDC(0)=0$ é possível um código 5B6B com $VDC=3$. Continuando o mesmo estudo para os casos de $SDC(0)=1$ e $SDC(0)=-1$, seríamos capazes de afirmar a realizabilidade de tal código.

Com este exemplo ilustrou-se a versatilidade do DP e, o fato de ser um instrumento de manuseio simples, não tira o mérito de seus resultados, os quais são transparentes e ricos em conteúdo, como poderemos observar já no próximo item, onde trataremos explicitamente de um dos esquemas de codificação propostos neste trabalho.

• O código $MPFmB(m+1)B$

Chamou-nos a atenção os códigos $PFmB(m+1)B$ propostos em [KRZYMIEN 89] tanto pela simplicidade de suas operações de codificação, como pelo valor da soma digital corrida apresentada: VDC igual a 10 para $PF7B8B$, enquanto alguns códigos de linha alfabéticos convencionais $7B8B$ atingem VDC igual a 4. A partir do estudo minucioso dos códigos $PFmB(m+1)B$, realizado gráfica e analiticamente, verificamos ser possível uma redução no valor da VDC. Desenvolvemos, então, uma nova versão destes códigos, à qual denominamos $MPFmB(m+1)B$ - "modified partially flipped" $mB(m+1)B$ ou $mB(m+1)B$ parcialmente rebatido modificado -, cujas regras de codificação passaremos a descrever a seguir.

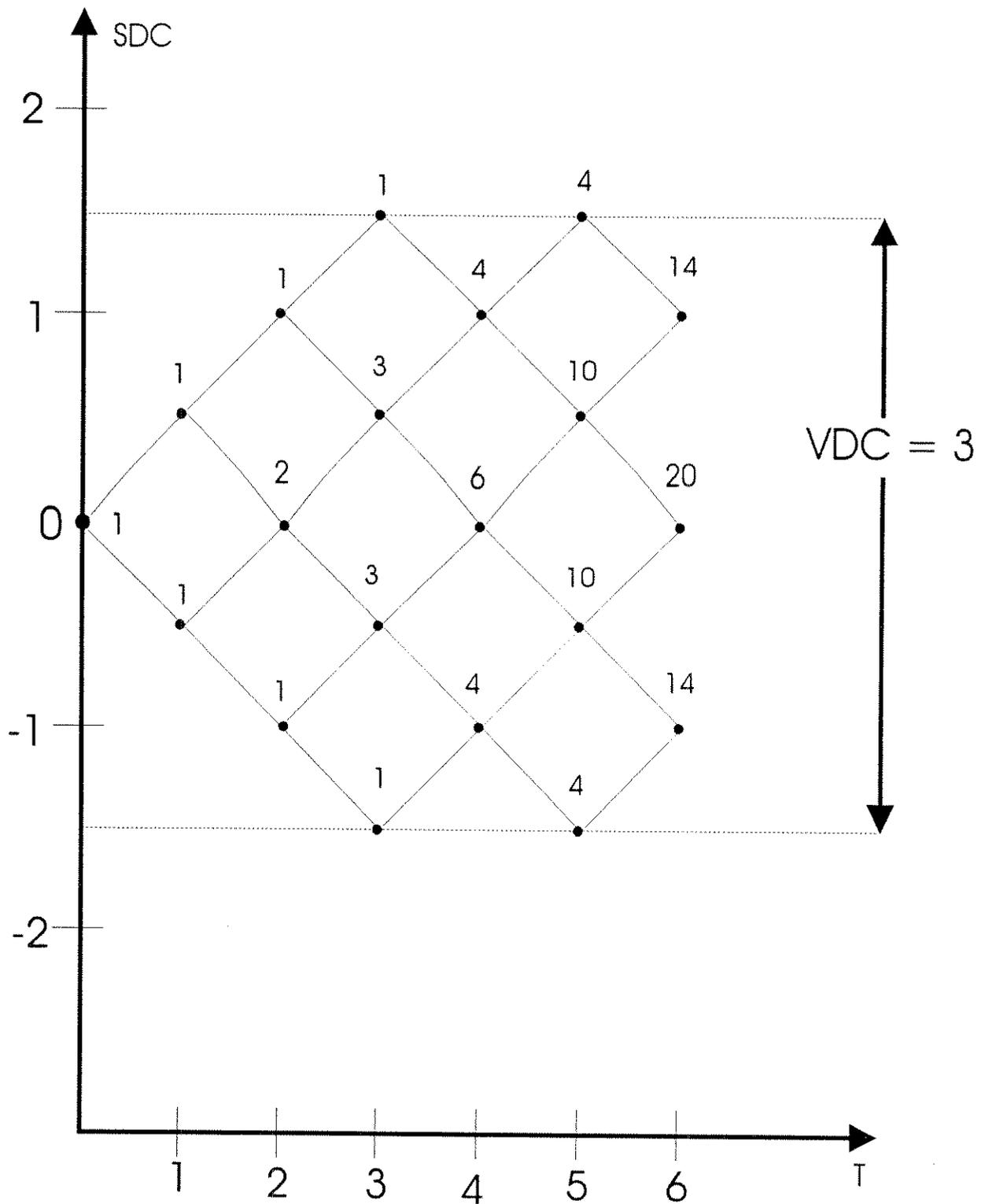


Figura 2.2

Seja $\Gamma = \text{SDC}(0)$, onde $\text{SDC}(0)$ é a soma digital corrida - expressa por (1) - no início do novo bloco, e seja $d(a)$ a disparidade - expressa por (2) - da palavra de informação a formada pelos dígitos a_1, a_2, \dots, a_m . Então os dígitos $b_1, b_2, \dots, b_m, b_{m+1}$ da palavra-código b serão assim construídos (sendo que $!$ é a representação da operação de inversão - ou negação - lógica):

$$(\alpha) \quad \text{se } \Gamma \cdot d(a) > 0, \quad \text{então: } b_i = !a_i \quad (\text{rebatimento})$$

$$(\beta) \quad \text{se } \Gamma \cdot d(a) \leq 0, \quad \text{então: } b_i = a_i, \quad \text{onde: } i=1, 2, \dots, m$$

$$(\gamma) \quad b_{m+1} = !\text{sinal}(d(a)), \quad \text{onde: } \text{sinal}(x) = \begin{cases} "0" & \text{para } x < 0 \\ "1" & \text{para } x > 0 \end{cases}$$

Impondo-se a restrição de que m seja ímpar - que é vantajosa até para os códigos de bloco alfabéticos tradicionais pela maior liberdade que traz ao projetista no uso da redundância [BROOKS 83] -, temos que $d(a)$ será sempre ou um valor positivo, ou negativo. Com esta premissa e sabendo que a regra γ é aplicada em todas as palavras, o processo de decodificação fica simples e independente do estado do decodificador. Basta apenas verificar o resultado do produto dado por $\Delta = d(b_1, b_2, \dots, b_m) \cdot d(b_{m+1})$. A análise do sinal do Δ indica se na codificação foi aplicada a regra α ou β , resultando nas seguintes regras de decodificação:

$$(\alpha') \quad \text{se } \Delta > 0, \quad \text{então } c_i = !b_i$$

$$(\beta') \quad \text{se } \Delta < 0, \quad \text{então } c_i = b_i$$

onde: $i=1, 2, \dots, m$ e c_i são os dígitos da palavra decodificada.

Com a aplicação das regras de codificação acima obtem-se três alfabetos implícitos, não memorizados, que correspondem aos casos: $\text{SDC}(0) < 0$, $\text{SDC}(0) = 0$ e $\text{SDC}(0) > 0$. Os gráficos da figura 2.3 ilustram os alfabetos progressivos (para $\text{SDC}(0) < 0$) e o alfabeto central ($\text{SDC}(0) = 0$) dos códigos $\text{MPFmB}(m+1)\text{B}$ e $\text{PFmB}(m+1)\text{B}$, para $m=7$. Os alfabetos regressivos são simétricos aos progressivos e por isso não os apresentamos.

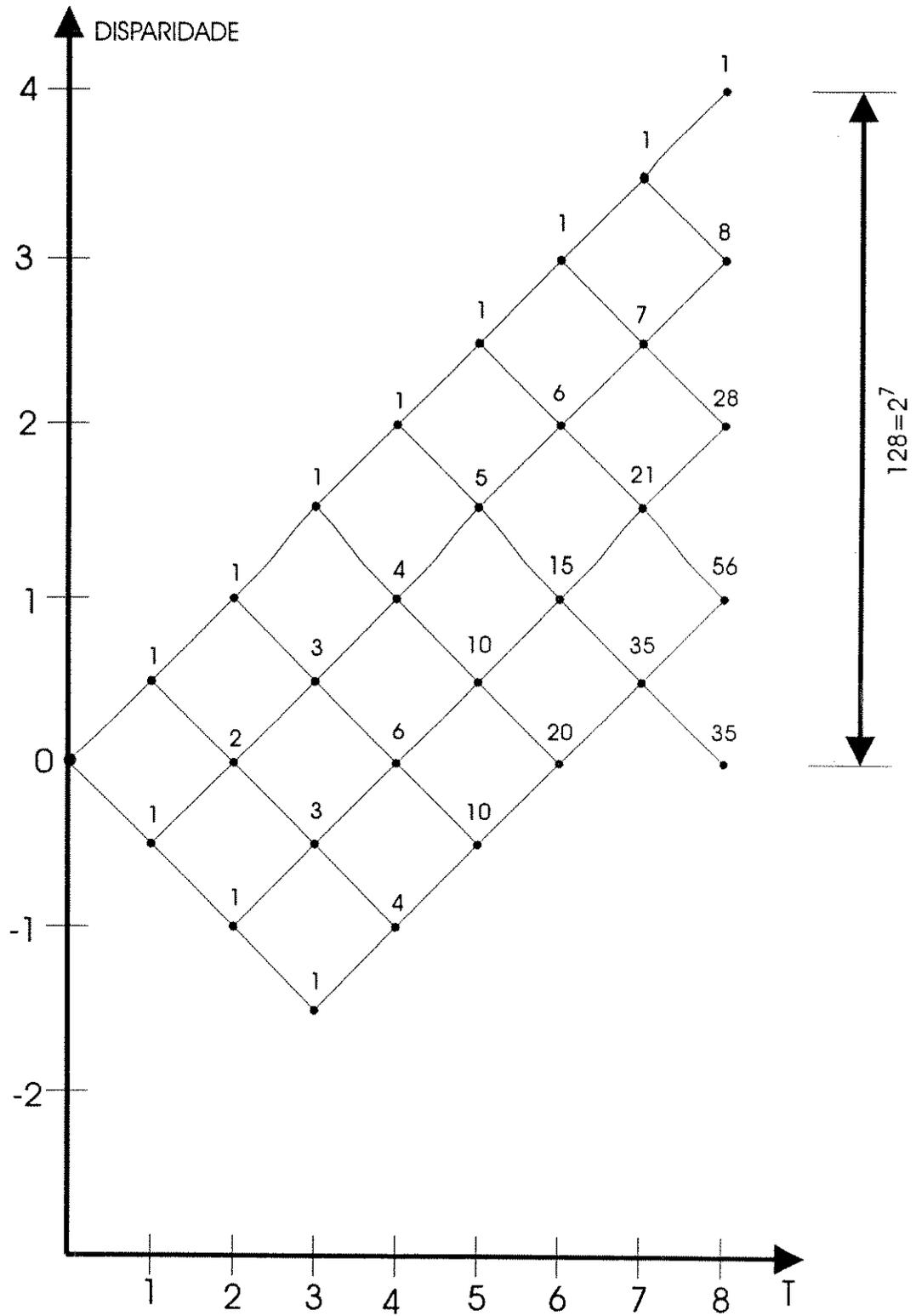


Figura 2.3.a - Alfabeto progressivo MPF7B8B

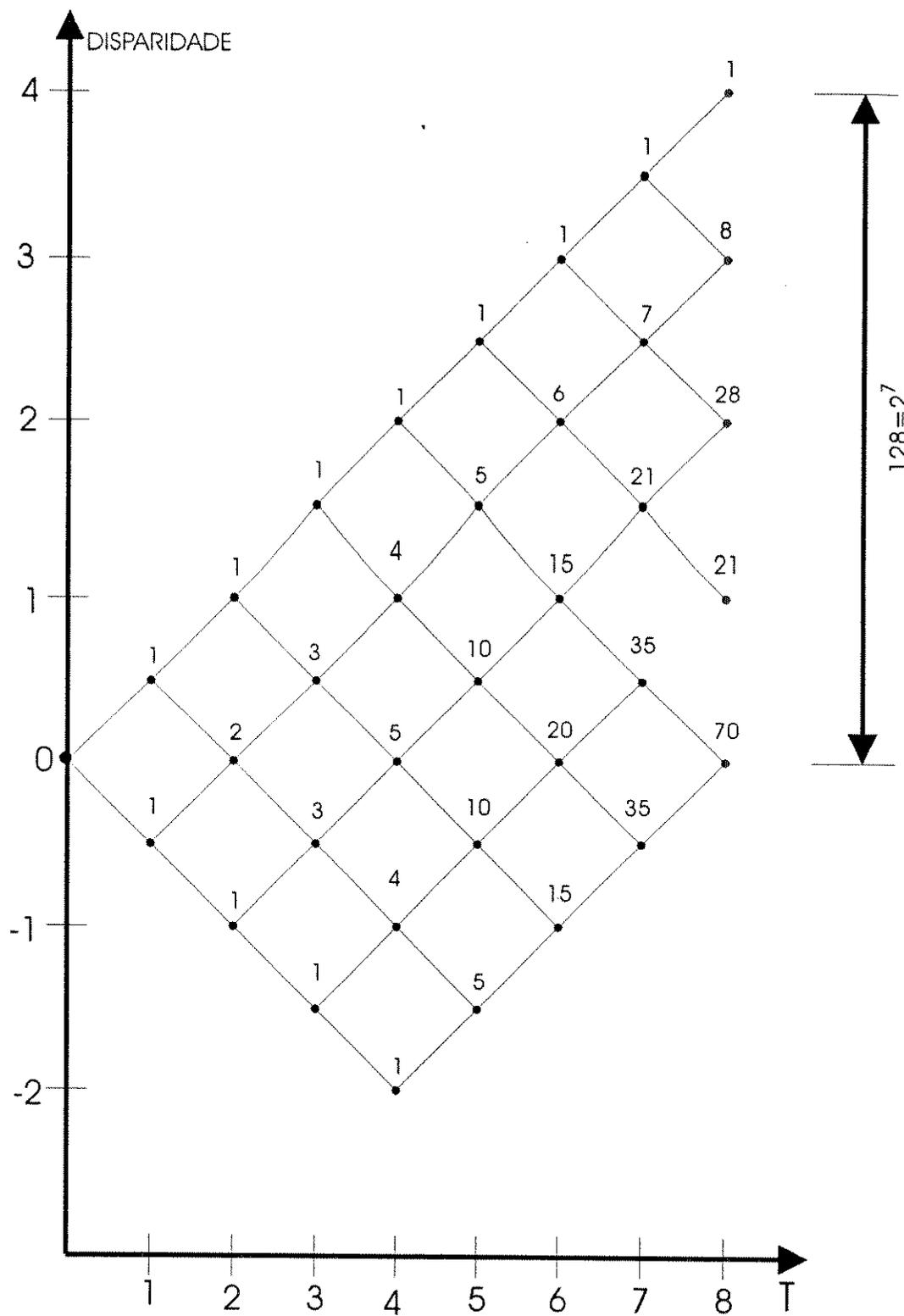


Figura 2.3.b - Alfabeto progressivo PF7B8B

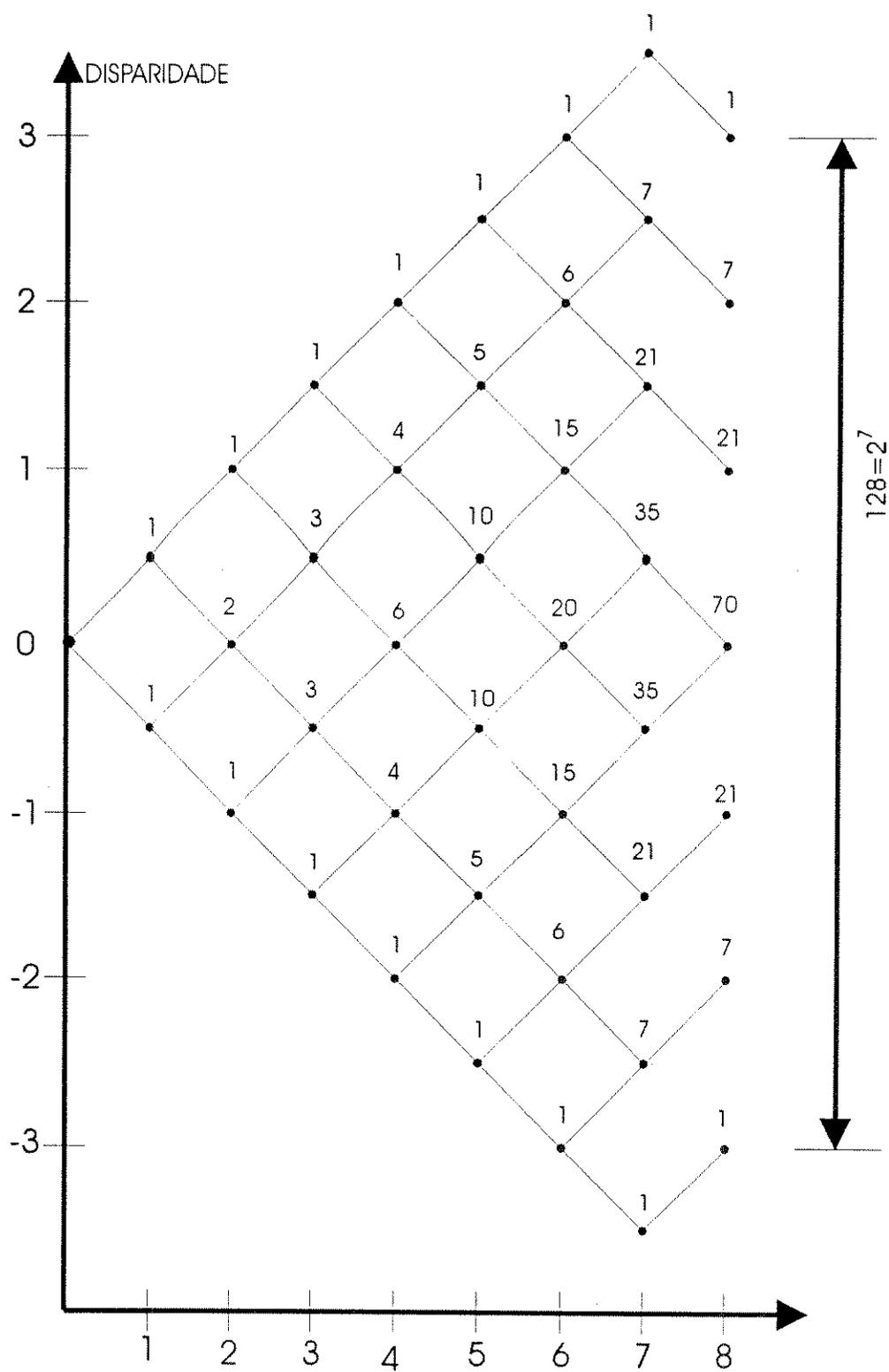


Figura 2.3.c - Alfabeto Central: PF7B8B e MPF7B8B

Fazendo a superposição de cada um dos alfabetos, temos - conforme mostra a figura 2.4 - todas as trajetórias da SDC para os códigos (note que, por se tratar de uma sobreposição de alfabetos, nem todas as trajetórias desta figura são possíveis). Por simples inspeção destes gráficos, verifica-se que os códigos MPFmB(m+1)B apresentam VDC menor que os códigos PFmB(m+1)B.

Analiticamente temos o seguinte: como (m+1) é par, cada palavra código adiciona um valor inteiro à SDC. Tomando então os estados terminais SDC(0) como inteiros, o maior valor terminal de SDC é $[-1+(m+1)/2]$, que corresponde à soma de -1 - maior valor negativo de SDC(0) - com (m+1)/2 - maior disparidade positiva da palavra-código, quando $\Gamma < 0$ (cfr. figura 2.5). Por simetria, o menor valor terminal corresponde a $1-(m+1)/2$. Portanto, os valores dos estados terminais da SDC são tomados do conjunto $\{-(m-1)/2, -(m-3)/2, \dots, (m-3)/2, (m-1)/2\}$. Para SDC(0)=0, os maiores incrementos são $\pm(m-1)/2$, que levam a SDC a estados já contemplados no conjunto acima.

Para os estados intermediários SDC(i), $i=1, 2, \dots, (m-1)$, a SDC pode atingir seu valor mínimo (SDC_{min}) e máximo (SDC_{máx}), que serão, respectivamente, (m-1)/4 abaixo do menor estado terminal e (m-1)/4 acima do maior estado terminal. Assim temos:

$$\begin{aligned} \text{SDC}_{\text{min}} &= -(m-1)/2 - (m-1)/4 = -(3m-3)/4 \\ \text{SDC}_{\text{máx}} &= (m-1)/2 + (m-1)/4 = (3m-3)/4 \end{aligned}$$

A VDC do código será portanto:

$$\text{VDC} = \text{SDC}_{\text{máx}} - \text{SDC}_{\text{min}} = (3m-3)/4 - [-(3m-3)/4] = (3m-3)/2, \quad (3)$$

que proporciona redução de 1 na VDC apresentada pelos códigos PFmB(m+1)B, quantificada pela expressão $(3m-1)/2$ [KRZYMIEN 89].

Algumas propriedades de transmissão do código MPFmB(m+1)B como conteúdo espectral, monitorabilidade da taxa de erro e sincronismo, serão discutidas posteriormente. Quanto ao fenômeno da extensão de erro, este código apresenta as mesmas características do PFmB(m+1)B, bem como quanto ao número máximo de símbolos idênticos consecutivos, que é dado por $2(m+1)$ - vide Apêndice 2B.

Constatamos, portanto, um progresso dentro dos códigos de linha algorítmicos, trazida pelo $MPFmB(m+1)B$, já que, uma diminuição da VDC, como vimos, implica na redução do valor pico-a-pico da flutuação das baixas frequências do sinal.

Contudo, os códigos balanceados que fazem uso de tabelas memorizadas conseguem valores inferiores para VDC. Tomando como exemplo $m=7$, teríamos, para o $PF7B8B$, $VDC = 10$; e para o $MPF7B8B$, $VDC=9$; enquanto que a VDC pode ser reduzida até 4 para os $7B8B$ alfabéticos convencionais [BROOKS 83]. O intervalo compreendido por códigos com $VDC=4$ sem restrição de memória e códigos com $VDC=9$ sem nenhuma necessidade de memória, apresentou-se como um campo a ser explorado, cujo estudo apresentamos a seguir.

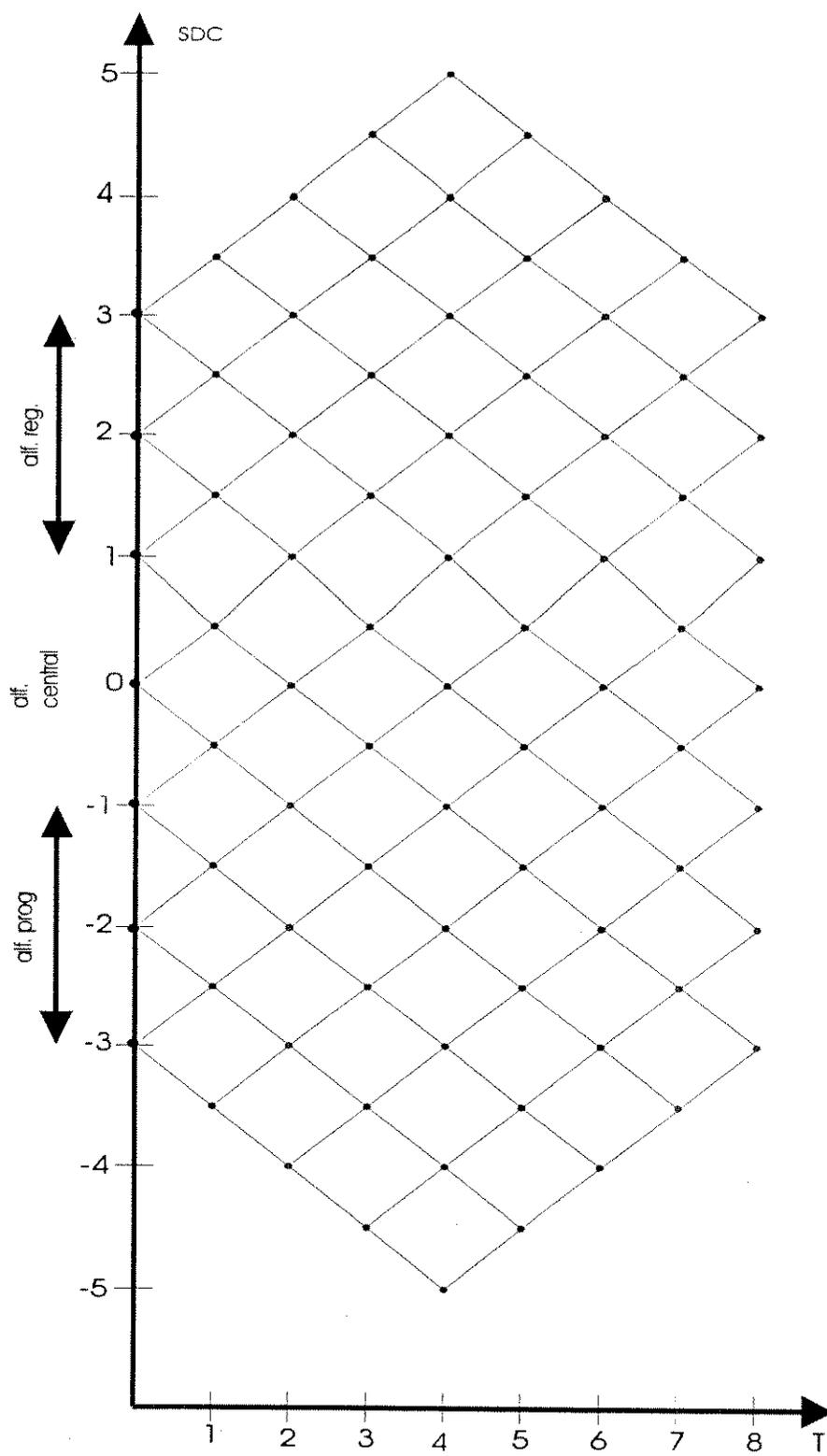


Figura 2.4.a - PF7B8B

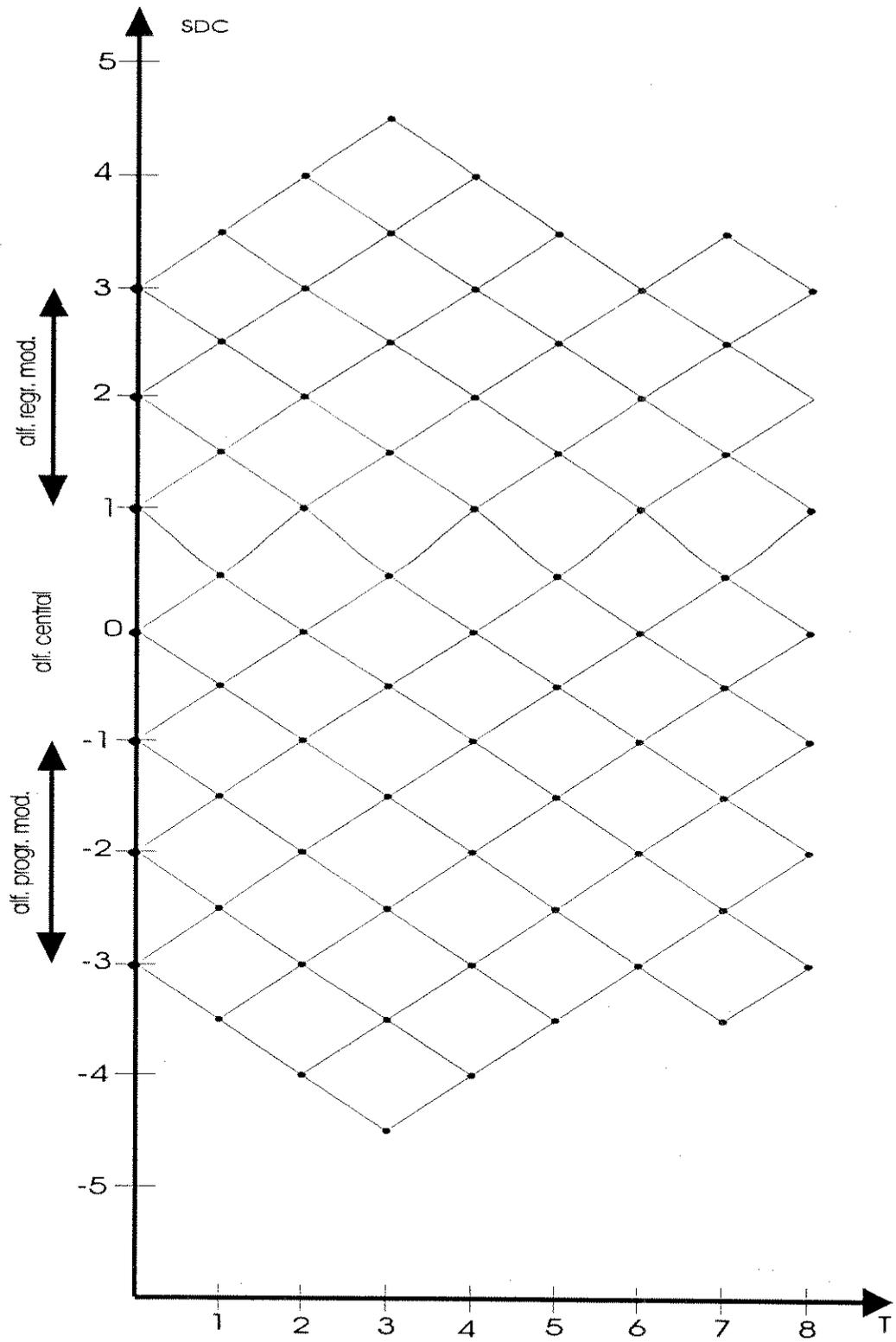


Figura 2.4.b - MPF7B8B

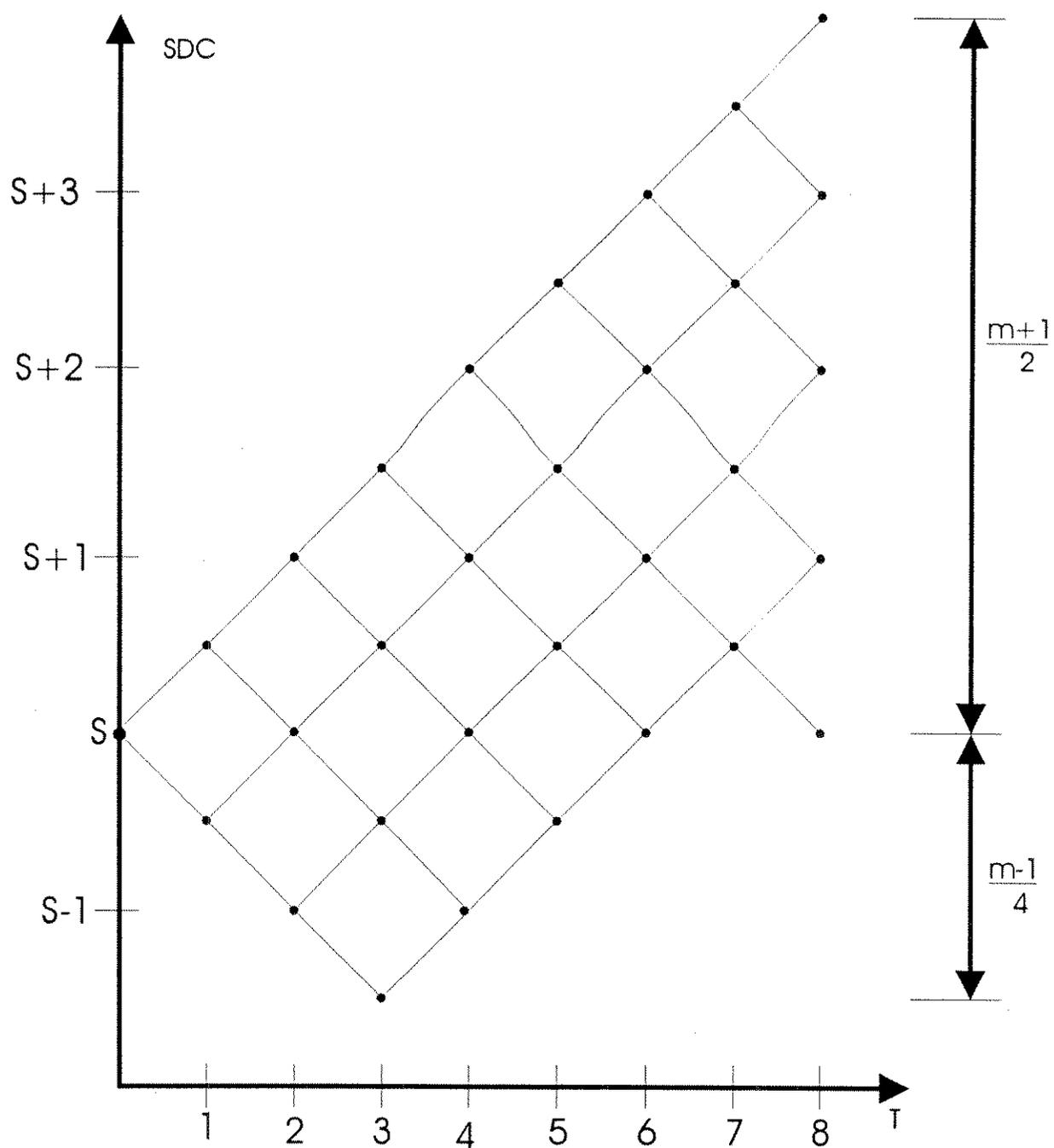


Figura 2.5.a

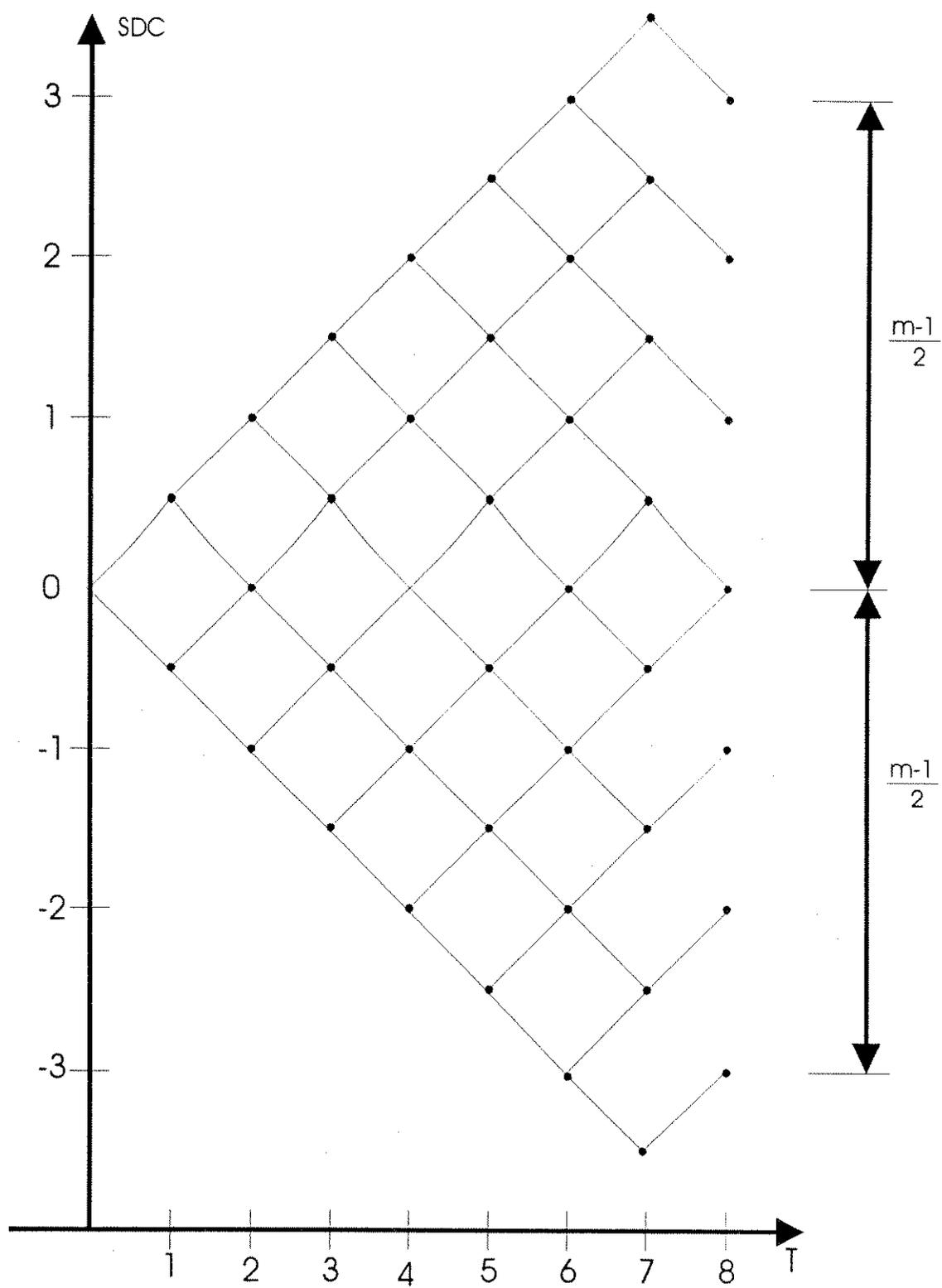


Figura 2.5.b

APÊNDICE 2A

Estudo do número de alfabetos necessários para códigos de blocos $mB(m+1)B$, m ímpar

Os códigos de bloco $mBnB$ necessitarão de dois alfabetos, no mínimo, quando $n=m+1$. A única exceção é para $m = 1$.

$$\text{Seja: } \binom{x}{y} = \frac{x!}{y!(x-y)!}$$

a) Análise para $m = 1$

Neste caso teremos $2^1 = 2$ palavras de informação e $\binom{2}{1} = 2$ palavras-código com disparidade nula. Portanto é possível a utilização de um único alfabeto, pois:

$$2^m \leq \binom{m+1}{(m+1)/2}$$

b) Análise para $m > 1$

$$\begin{aligned} \text{Seja: } \quad m^n &= m+2 \\ n &= m+1 \Rightarrow n^n = n+2 \end{aligned}$$

Então teremos:

$$2m^n = 2m+2 = 4 \cdot 2m$$

$$\binom{n^n}{n^n/2} = \frac{n^n!}{(n^n/2)!(n^n/2)!} = \frac{(n+2)(n+1)}{(n/2+1)^2} \times \frac{n!}{(n/2)!(n/2)!} = 4 \frac{n^2 + 3n + 2}{n^2 + 4n + 4} \times \binom{n}{n/2}$$

É fácil verificar que : $4 \frac{n^2 + 3n + 2}{n^2 + 4n + 4} < 4$ para $n > 0$.

Então, como: $\binom{n}{n/2} = \binom{m+1}{(m+1)/2} \Rightarrow 2^m > \binom{m+1}{(m+1)/2}$, para $m > 1$,

confirmando a afirmação inicial da necessidade de mais de um alfabeto para os códigos de bloco mBnB com $n=m+1$, m ímpar.

APÊNDICE 2B

• Estudo da extensão de erro

As regras de codificação dos códigos PFmB(m+1)B e MPFmB(m+1)B são tais que a análise do efeito da extensão de erro apresentada a seguir é válida para ambos [KRZYMIEN 89].

Consideremos a possibilidade de ocorrer um erro simples na palavra-código $b = b_1, b_2, \dots, b_m, b_{m+1}$. Haverá extensão de erro na decodificação em três situações:

a) o número de bits "1" em b é $(m-1)/2$ e um erro afeta um bit "0" desta palavra. Este evento tem possibilidade P_a :

$$P_a = \binom{m+1}{(m-1)/2} \cdot p^{(m-1)/2} \cdot (1-p)^{(m+1)/2} \cdot \frac{m+1}{2m} \quad \text{e gera } m-2 \text{ erros adicionais.}$$

onde: p = probabilidade de ser "1" o bit b_i da palavra b , $i = 1, 2, \dots, m$.

b) o número de bits "1" em b é $(m+1)/2$ e um erro afeta um bit "1". Isto pode ocorrer com probabilidade P_b :

$$P_b = \binom{m+1}{(m+1)/2} \cdot p^{(m+1)/2} \cdot (1-p)^{(m-1)/2} \cdot \frac{m+1}{2m} \quad \text{e gera } m-2 \text{ erros adicionais.}$$

c) o erro ocorre no bit b_{m+1} . A probabilidade desta ocorrência é:

$$P_c = 1/(m+1) \quad \text{e gera } m \text{ erros adicionais.}$$

Sendo P_e a taxa de erros na linha, a probabilidade total de erro na seqüência decodificada é:

$$Pd \cong Pe [1 + (m-2).Pa + (m-2).Pb + m.Pc]$$

Substituindo os valores de Pa, Pb e Pc, temos:

$$Pd \cong \left\{ 1 + m/(m+1) + \right. \\ \left. + [(m-2).(m+1)/(2m)] \cdot \binom{m+1}{(m-1)/2} \cdot [p^{(m+1)/2} \cdot (1-p)^{(m-1)/2} + p^{(m-1)/2} \cdot (1-p)^{(m+1)/2}] \right\}$$

Para o caso de $m = 7$ e $p = 0,5$, temos que $Pd \cong 3,438 Pd$.

• Considerações sobre o número máximo de símbolos idênticos consecutivos

Um dos indicadores da qualidade das propriedades de recuperação de relógio de um código é o número de símbolos idênticos consecutivos ($NC_{\text{máx}}$). Para os códigos $PFmB(m+1)B$ e $MPFmB(m+1)B$ este valor é:

$$a) PFmB(m+1)B = NC_{\text{máx}} = (m+1)/2 + m + 1 + (m+1)/2 = 2(m+1)$$

$$b) MPFmB(m+1)B = NC_{\text{máx}} = (m+3)/2 + m + 1 + (m-1)/2 = 2(m+1)$$

Alguns autores consideram o resultado acima elevado. Porém, $NC_{\text{máx}}$ só é atingido para algumas seqüências específicas de palavras-fonte que devem ocorrer a partir de determinados valores para $SDC(0) = 0$, ou seja, com a utilização de embaralhamento dos dígitos de informação, garante-se que $NC_{\text{máx}}$ é atingido com baixa probabilidade.

Exemplificando, temos o seguinte: para $m = 7$, foi apresentada a comparação abaixo [KRZYMIEN 89]:

- $PF7B8B$: $VDC = 10$, $NC \text{ máx} = 16$, $N_{\text{med}} = 1,97$
- $7B8B$ convencional: $VDC = 6$, $NC \text{ máx} = 6$, $N_{\text{med}} = 1,78$

Observa-se expressiva diferença entre $NC_{\text{máx}}$ dos códigos em questão, contudo esta discrepância não ocorre para o número médio de símbolos idênticos consecutivos (N_{med}). O resultado da comparação confirma o que foi comentado mais acima e sugere que as propriedades de recuperação de relógio do PF7B8B e, por extensão, do MPF7B8B, não são tão inferiores às dos códigos alfabéticos tradicionais, como se era levado a deduzir confrontando valores de $NC_{\text{máx}}$.

Capítulo 3

Códigos de memória reduzida

• Introdução

O hiato entre os valores de VDC citados no capítulo anterior, sugeriu-nos a existência de um compromisso entre o tamanho de tabela memorizada e VDC. Nosso objetivo é caracterizar esse compromisso quantitativamente para realizações especificadas por operações simples efetuadas sobre os dígitos de informação, isto é, sem abandonar a linha mestra dos códigos algorítmicos.

Em [Bi 89] é apresentada uma solução para códigos 5B6B que minimiza o número de portas lógicas utilizadas dentro de uma tecnologia específica de hardware. A técnica desenvolvida neste capítulo, bem como a do capítulo anterior, não está vinculada a nenhum hardware particular.

As operações consideradas são as mesmas do capítulo precedente: rabatimento ou não, seguido de um bit de redundância - verificação ("check bit") -, que informa a ocorrência do rabatimento ao decodificador.

Se, para chegarmos ao $MPFmB(m+1)B$ buscávamos regras de codificação tais que o resultado fosse uma VDC menor possível sem a necessidade de tabelas, estamos agora interessados em impor valores de VDC e verificar qual o mínimo número de palavras que devem ser

memorizadas em tabelas, fazendo uso do esquema básico de codificação dos códigos $MPFmB(m+1)B$.

Esta ação de impor valores de VDC inferiores ao valor da equação (3) é quem determinará quantas e quais palavras deverão ser tabeladas, já que a codificação por operações elementares não será possível para todas as palavras-fonte.

• Método para obtenção da tabela de tamanho mínimo

Para ilustrar a metodologia utilizada no estabelecimento desta tabela reduzida, vamos considerar o caso de $m=7$. O DP contendo o conjunto de palavras-fonte sem qualquer codificação é mostrado pela figura 3.1.

Este diagrama é então aplicado num gráfico de SDC, no qual estão marcados os limites da VDC desejada. A figura 3.2 ilustra o caso de $VDC=6$, que vamos tomar como exemplo.

Ao delimitarmos a SDC pelo valor da VDC escolhido, estamos implicitamente restringindo o número (ou o conjunto) de estados terminais: como $m+1$ é par, as disparidades das palavras-código são valores inteiros, significando que os estados terminais estarão sempre espaçados de 1 entre si. Somado a isso, é fácil mostrar que, para $m>5$, os estados terminais localizados sobre o limite da VDC, ou distanciados de apenas 0,5 deste limite, são inviáveis pois o número máximo de palavras-código que, partindo de um destes estados, respeitam os limites da soma digital corrida, é inferior ao número de palavras-fonte.

Para o caso que estamos analisando, por exemplo, partindo de um dos estados $SDC(0)=\pm 3$ ou $SDC(0)=\pm 2,5$ só existem 70 ou 126 palavras-código disponíveis, respectivamente. Como necessitamos codificar $2^7=128$ palavras-fonte, estes estados não se qualificam como estados terminais da SDC. Restam então, os valores terminais que satisfazem $-2,5 < SDC(0) < 2,5$, seguindo-se que o número de estados terminais será no máximo 5.

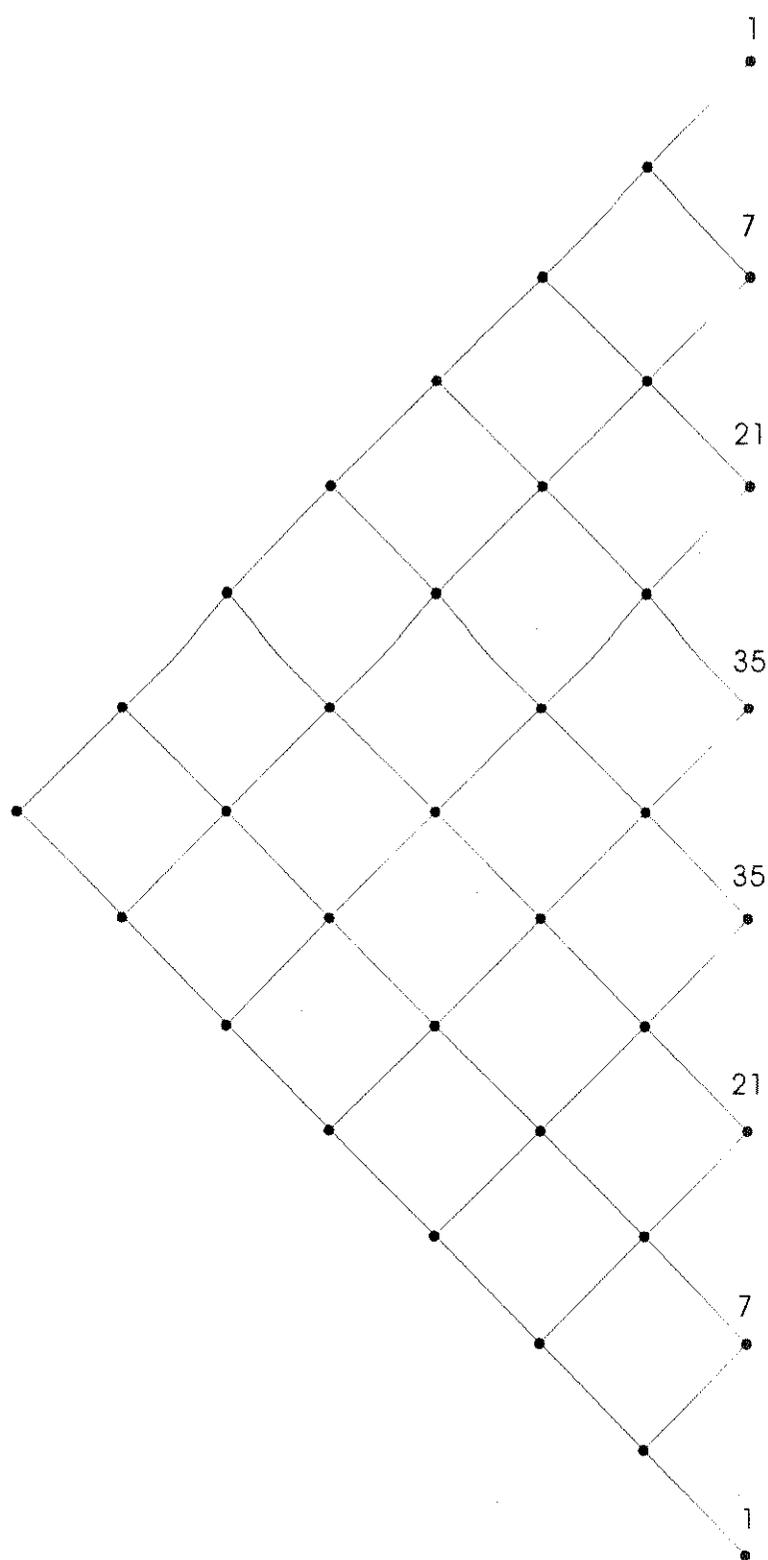


Figura 3.1`

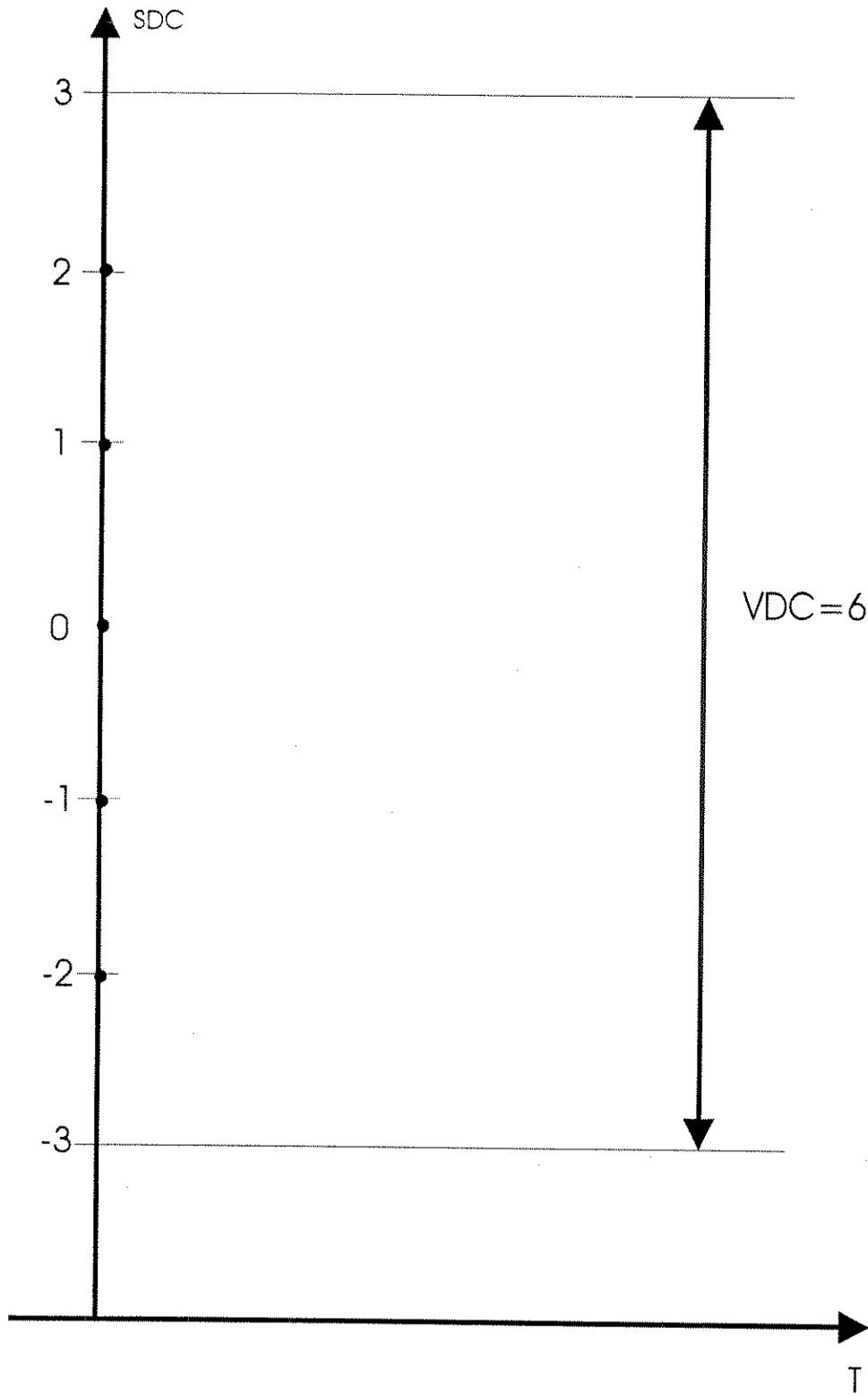


Figura 3.2

Generalizando para qualquer valor de m , $m > 5$, o número máximo de estados terminais será quantificado por VDC-1. Entretanto, o número mínimo de estados terminais é dependente do valor de m . Para o caso de $m=7$, fazendo uso do DP verifica-se sem dificuldade que é igual a 3.

No exemplo em questão, para o valor de VDC=6 escolhido, o número de estados terminais terá valores no conjunto $\{3, 4, 5\}$. Vamos considerar o caso de cinco estados terminais, o que implica estudar os estados do conjunto $\{-2, -1, 0, 1, 2\}$, em destaque na figura 3.2.

A figura¹ 3.3 mostra a superposição de todas as trajetórias da SDC para as 128 palavras de informação a partir de $SDC(0)=2$.

A parte das trajetórias que ultrapassa o limite da SDC aparece tracejada. Essas violações precisam ser corrigidas ou por rebatimento da palavra toda ou por tabelamento das palavras envolvidas. As outras palavras seguem a seguinte regra:

$$(A) \quad \begin{aligned} b_i &= a_i, & i=1, \dots, m \\ b_{(m+1)} &= \text{!sinal}[d(a)] \end{aligned}$$

onde: a_i - dígitos da palavra de informação
 b_i - dígitos da palavra-código

A operação de rebatimento é dada pela regra:

$$(B) \quad \begin{aligned} b_i &= \text{!}a_i, & i=1, \dots, m \\ b_{(m+1)} &= \text{!sinal}[d(a)] \end{aligned}$$

Da figura 3.3 resulta que deve ser aplicada a regra do rebatimento para todas as palavras com disparidades 3,5, 2,5 e 1,5 (1, 7, e 21 palavras, respectivamente) e também para algumas palavras (sete) com disparidade 0,5. Dentre as palavras com disparidade -0,5 (que ocupam o paralelogramo ABCD) somente a palavra 1110000

¹ Legenda das figuras a partir da figura 3.3:

- número em um nó: total de palavras que atingem aquele nó;
- * : palavra que necessita ser tabelada;
- arco com *número*: operação de rebatimento em *número* palavras;
- ramo com (A) ou (B): indica tipo de codificação - regra (A) ou (B)
- ramos com T_i : palavra utilizada para codificar a palavra i que deve ser tabelada

atinge o ponto B, que excede os limites da VDC. Ao tentarmos aplicar a regra do rebatimento para esta palavra, sua codificação seria assim determinada:

$$\begin{array}{ccc} 1110000 & \xrightarrow{\text{(B)}} & 0001111 \\ \text{palavra fonte} & & \text{palavra-código} \end{array}$$

Considerando que estamos analisando o estado $SDC(0)=2$, a transmissão desta palavra-código levaria $SDC(0)$ ao valor $SDC(8)=3$ (pois $d(0001111)=1$), que não pertence ao conjunto de estados terminais. Portanto, a palavra-fonte 1110000 deve ser tabelada.

Codificando a palavra 1110000 por 01010100, cujas transições são indicadas por T_3 na figura 3.4, temos que todas as palavras-fonte ficam contempladas, encerrando a análise para $SDC(0)=2$.

Como o estado $SDC(0)=-2$ é simétrico ao estado analisado anteriormente, valem para este as mesmas considerações feitas acima e teremos também uma palavra-fonte a ser tabelada, obviamente, por simetria, 0001111, a qual, para manter a simetria do código, será codificada por 10101011.

Passemos a avaliar o estado $SDC(0)=1$. A figura 3.5 mostra o conjunto de todas as trajetórias de SDC para as 128 palavras de informação que partem deste estado.

Temos, observando as linhas tracejadas deste diagrama, que apenas uma palavra com disparidade 1,5 e todas (sete) com disparidade 2,5 devem ser rebatidas - codificadas pela regra (B). Aplicando a regra (B) para a palavra 1111111, obtem-se a palavra código 00000000, cuja disparidade igual a -4 leva $SDC(0)=1$ a $SDC(8)=-3$, que não pertence ao conjunto dos estados terminais permitidos. Portanto, a palavra 1111111 é a única que deve ser tabelada e, se escolhermos 00110010 para codificá-la, fica completa a análise deste estado terminal, que está ilustrado pela figura 3.6.

Recorrendo novamente à simetria, temos que, para o estado terminal $SDC(0)=-1$, a única palavra a ser tabelada é 0000000, que será codificada por 11001101.

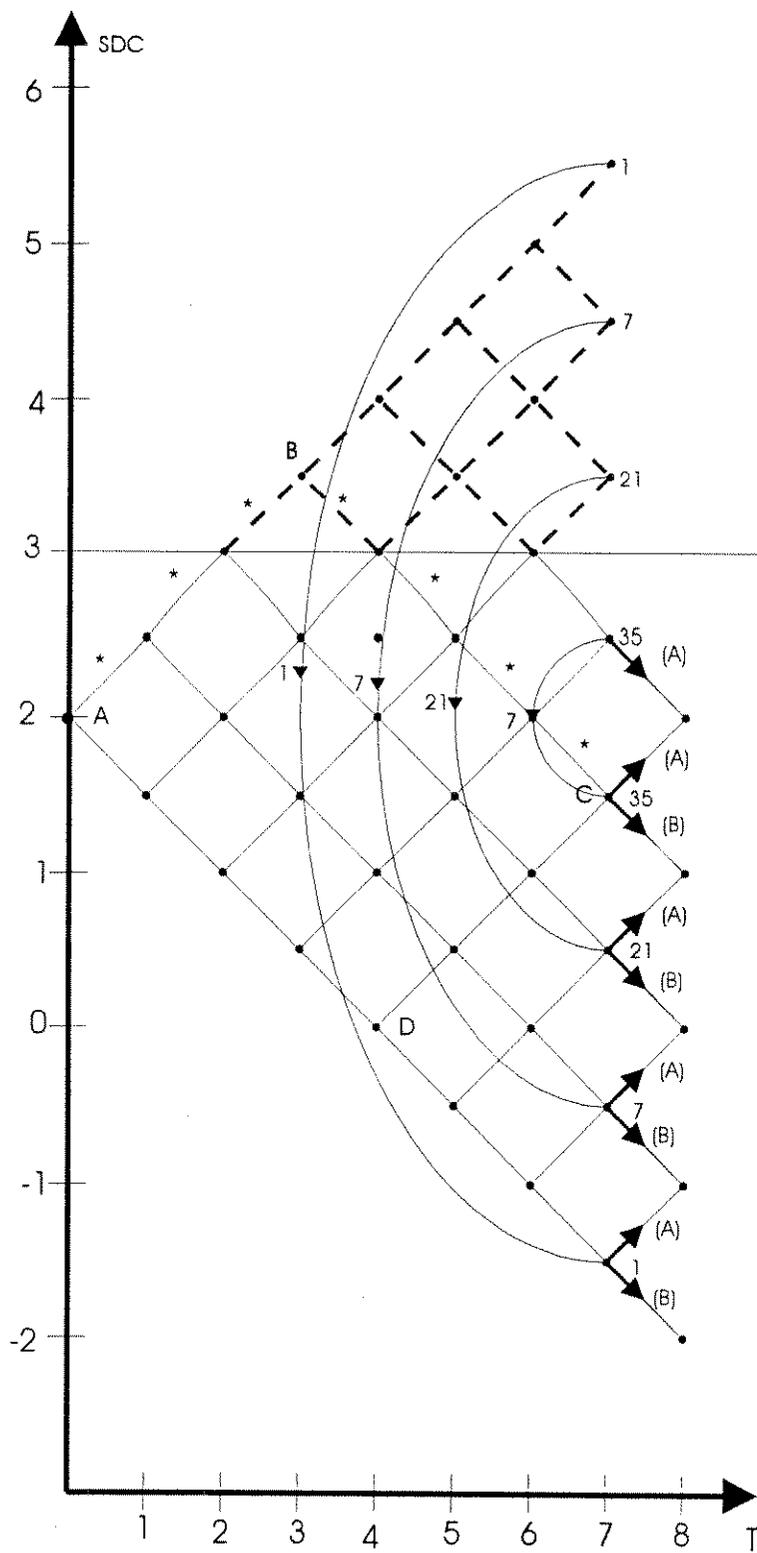


Figura 3.3

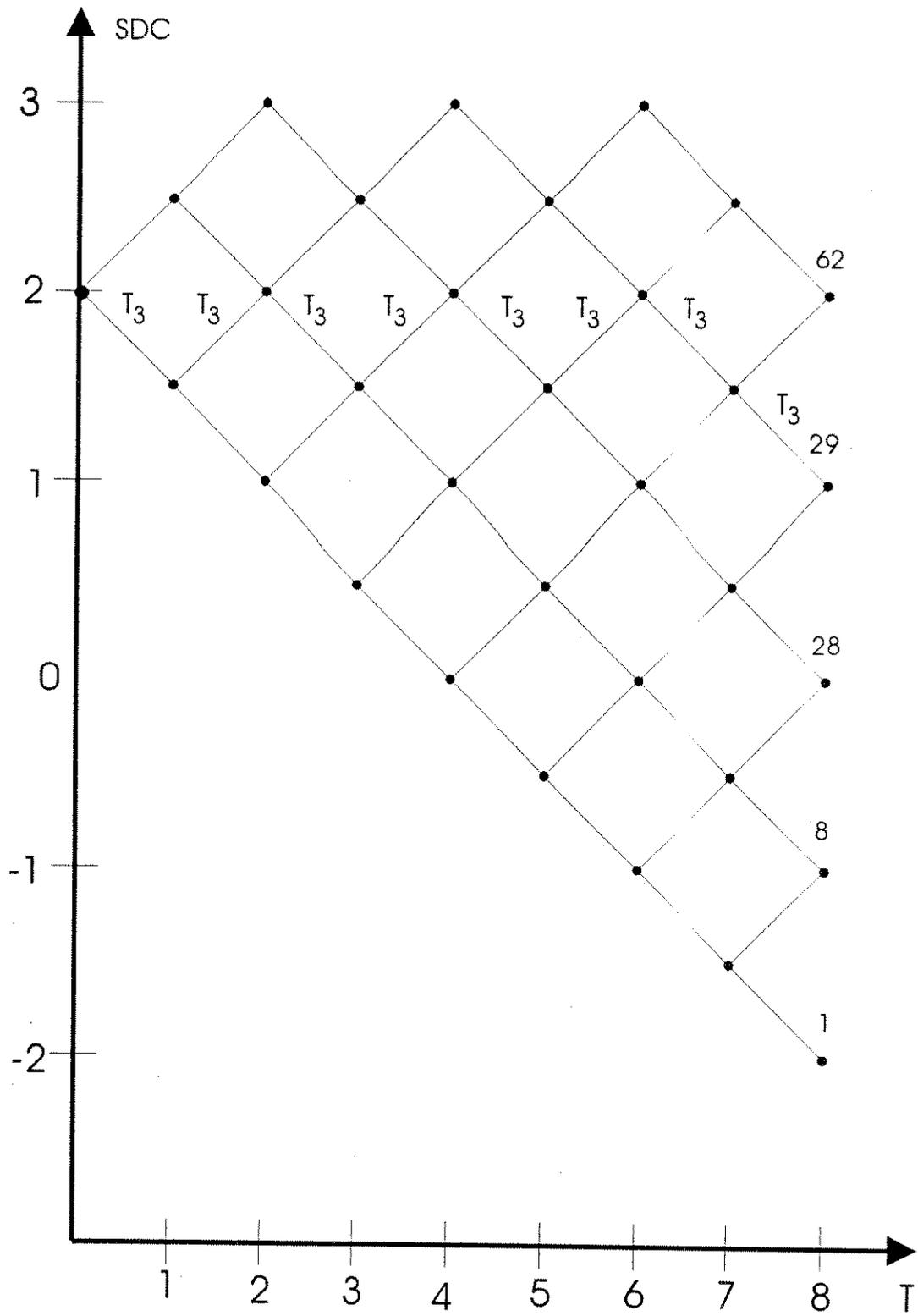


Figura 3.4

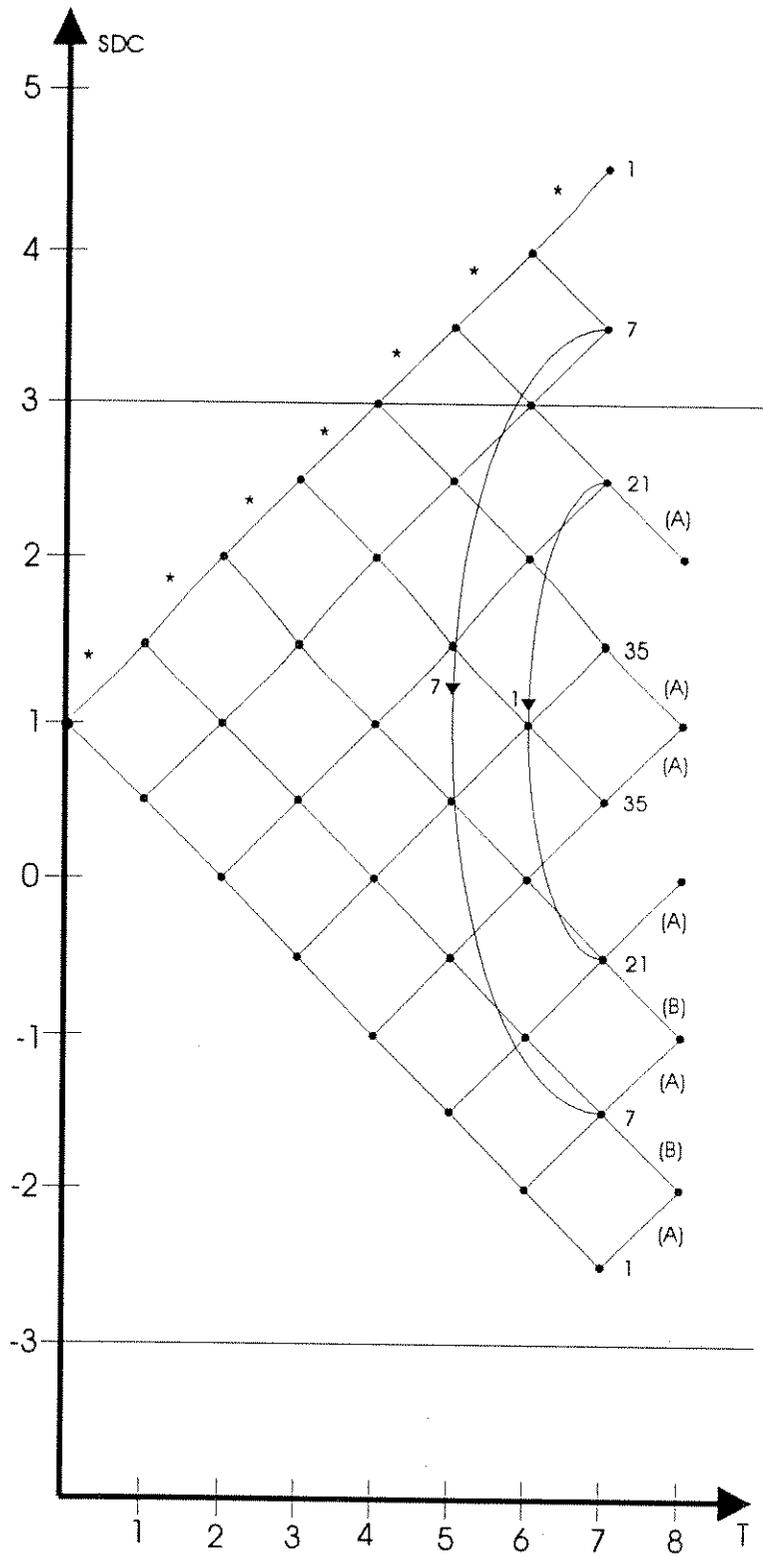


Figura 3.5

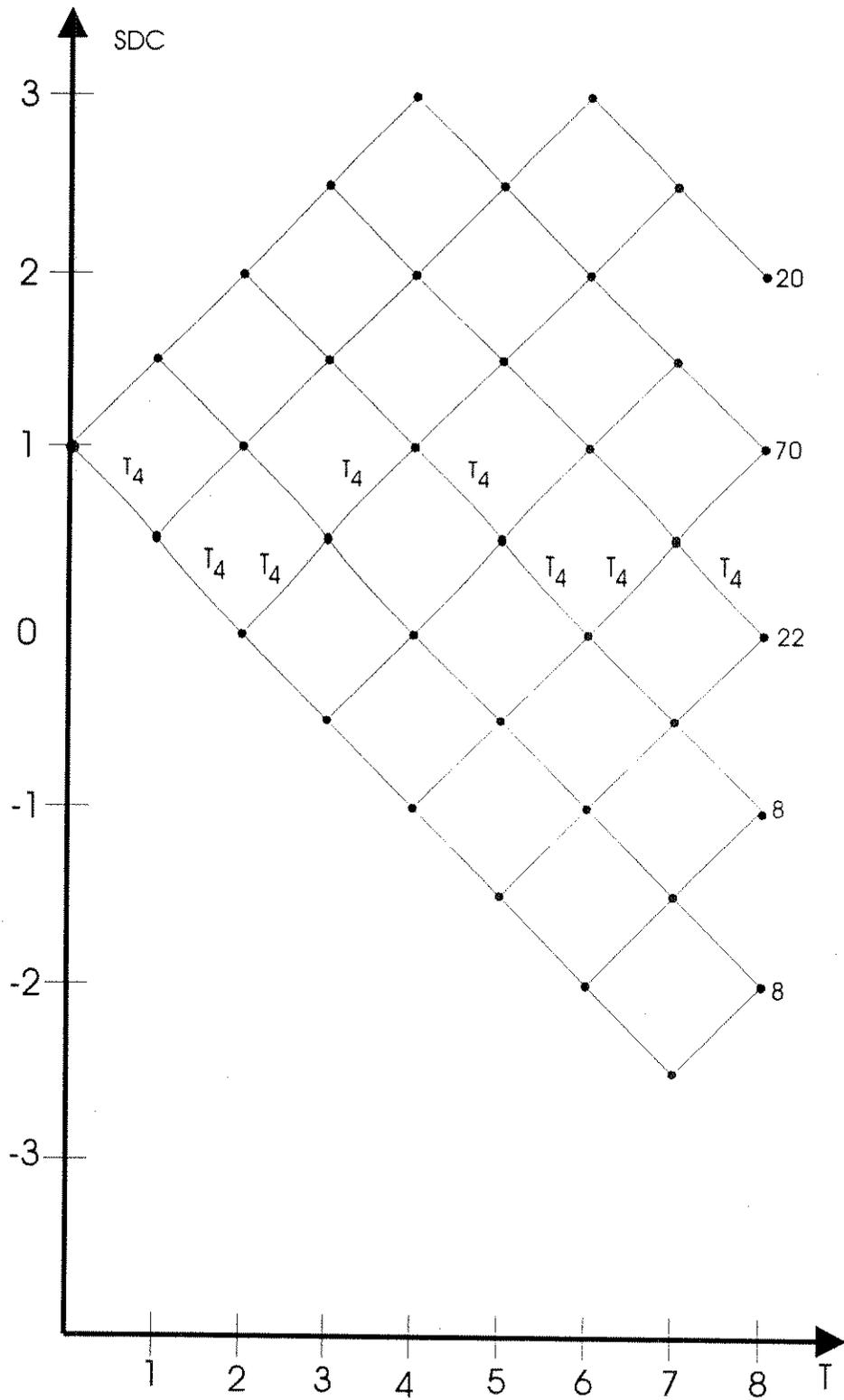


Figura 3.6

Resta-nos analisar o estado $SDC(0)=0$, aplicando a mesma metodologia. A figura 3.7 representa o conjunto das 128 palavras de informação partindo deste estado.

Através da inspeção da figura 3.7, resulta que as palavras 0000000 e 1111111 não têm solução para sua codificação através das regras (A) ou (B) e devem ser tabeladas. Estas duas palavras de informação já fazem parte da tabela - imposição dos casos de $SDC(0)=1$ e $SDC(0)=-1$ e, portanto, nada mais há que fazer, encerrando a análise dos cinco estados terminais para $VDC=6$. A figura 3.8 ilustra a codificação para $SDC(0)=0$.

A superposição dos diagramas finais obtidos para cada um dos estados é mostrada na figura 3.9, que não tem outro significado senão indicar quais são os possíveis estados do codificador num instante qualquer. Portanto não deve ser interpretado como um diagrama de transições possíveis.

Temos, portanto, para $m=7$, $VDC=6$, com utilização de 5 estados terminais, apenas 4 palavras-fonte tabeladas para algum estado da SDC . Todas as demais 124 palavras são codificadas e decodificadas algoritmicamente.

A escolha das palavras-código para as palavras-fonte da tabela seguiu alguns critérios, que passaremos a descrever. Primeiramente, como é lógico, procurou-se palavras que em nenhum estado da codificação poderiam ser utilizadas pelas regras (A) ou (B) e que pudessem ser empregadas em qualquer estado, quando necessário. Como as palavras-fonte com disparidade= $\pm 0,5$ são normalmente codificados pela regra (A), optou-se por palavras-código que, para estas disparidades utilizassem a regra (B). O resultado obtido foi o seguinte:

palavra-fonte	palavra-código
0000000	11001101
0001111	10101011
1110000	01010100
1111111	00110010

Tabela 3.1

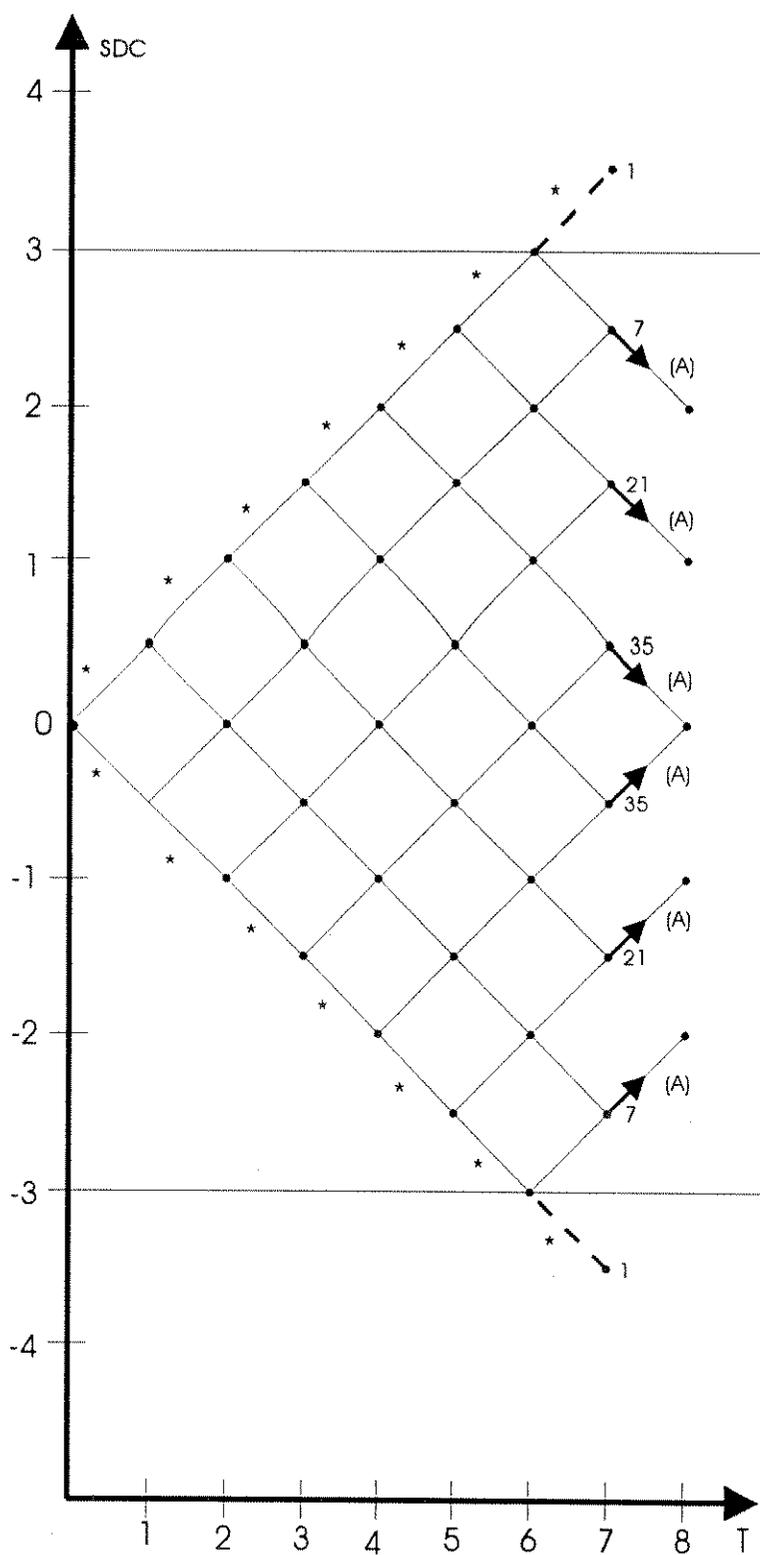


Figura 3.7

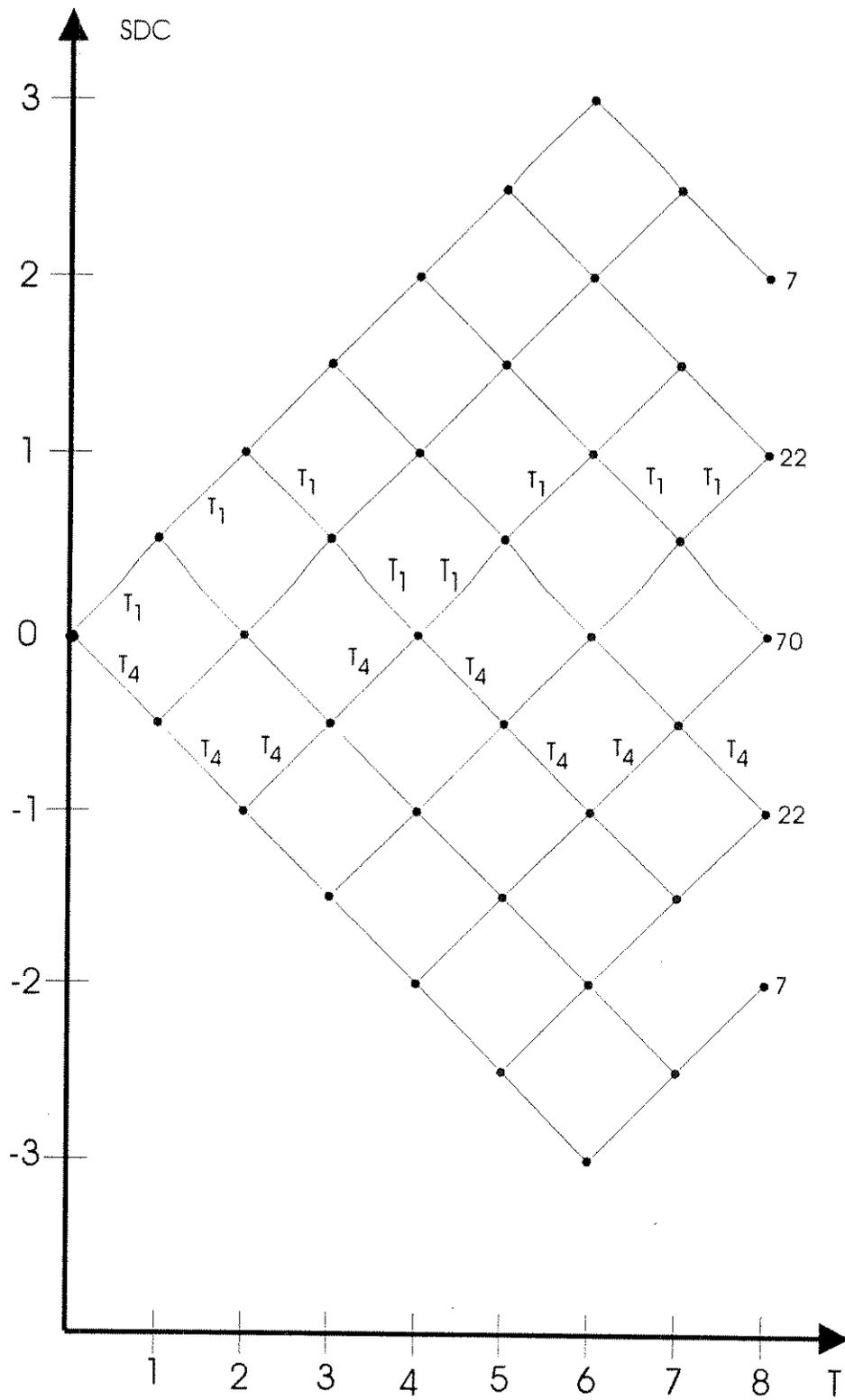


Figura 3.8

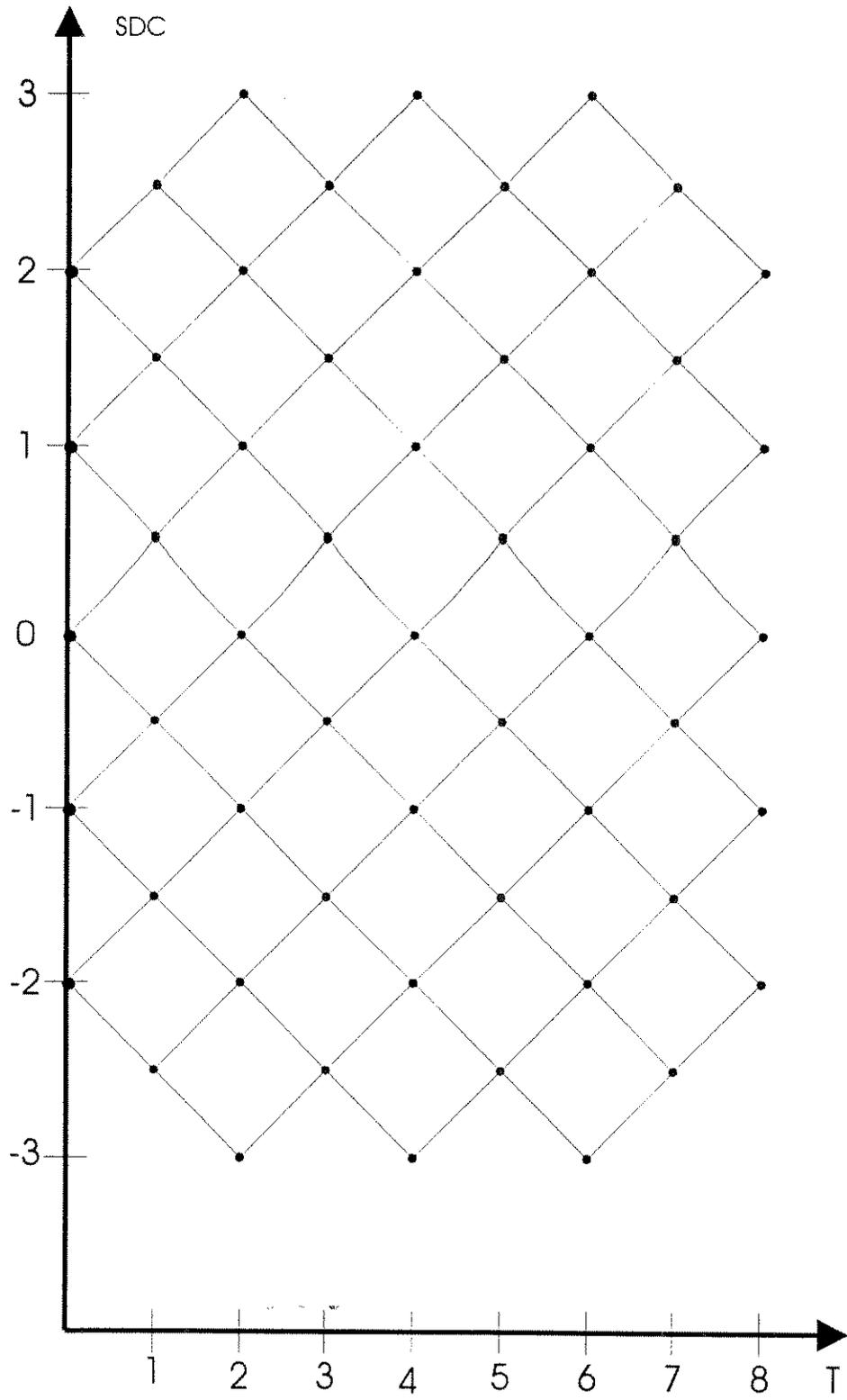


Figura 3.9

Porém, para o caso que estamos considerando há 14 palavras (7 para $SDC(0)=2$ e 7 para $SDC(0)=-2$) que são codificadas pela regra (B) e possuem disparidade $=\pm 0.5$. Então, para diferenciar as palavras-código utilizadas na tabela daquelas 14, basta verificar que as 4 palavras-código da tabela possuem disparidade acumulada até o 4º bit (inclusive) igual a zero.

Deste modo, a identificação no decodificador de que uma palavra-código recebida deve acionar a tabela poderá ser implementada pelo menos de dois modos: o primeiro, onde um "hardware" combinacional identifica as 4 palavras e as decodifica; o segundo que analisa as seguintes condições: disparidade acumulada até o 4º bit nula, disparidade até o 7º bit igual a ± 0.5 e codificação pela regra (B).

Seja qual for o caminho seguido, uma vez determinado o acionamento da tabela, para selecionar uma dentre as quatro, fica a sugestão de se utilizar apenas os dois últimos bits da palavra-código (cfr. destaque em negrito na tabela 3.1). Além disso, ambos modos são muito simples e rápidos, pois os elementos de decisão são valores normalmente monitorados pelo decodificador.

A decodificação das palavras que não acionam a tabela é feita de modo idêntico ao MPFmB(m+1)B: pela verificação do sinal do produto $Z=d(b_1, b_2, \dots, b_m).d(b_{m+1})$, que informa se a codificação foi através da regra (A) ou (B). Assim, as regras utilizadas pelo decodificador para palavras não tabeladas são:

$$(A') \quad \text{se } Z > 0, \quad \text{então } c_i = !b_i$$

$$(B') \quad \text{se } Z < 0, \quad \text{então } c_i = b_i$$

onde: $i=1, 2, \dots, m$

b_i = dígitos da palavra codificada

c_i = dígitos da palavra decodificada

A tabela 3.2 resume o caso do exemplo considerado - $m=7$, $VDC=6$ e 5 estados terminais -, mostrando a descrição funcional de uma possível implementação.

	pal. inform.	S=2	S=1	S=0	S=-1	S=-2
CO	0000000	algor.	algor.	11001101	11001101	algor.
DI	0001111	algor.	algor.	algor.	algor.	10101011
FI	1110000	01010100	algor.	algor.	algor.	algor.
CA	1111111	algor.	00110010	00110010	algor.	algor.
DOR	outras(124)	algor.	algor.	algor.	algor.	algor.
DECO	aciona	"hardware"	"hardware"	"hardware"	"hardware"	"hardware"
DIFI	tabela					
CADOR	outras	algoritmo	algoritmo	algoritmo	algoritmo	algoritmo

Tabela 3.2 - onde: $S=VDC(0)$

Repetindo-se a análise anterior, onde consideramos a utilização de cinco estados terminais, para as outras duas situações (4 e 3 estados terminais) encerra-se o estudo da imposição de $VDC=6$.

Mantendo-se o valor de $m=7$, utilizamos a metodologia apresentada anteriormente para os outros valores de VDC de 4 a 9 (limite teórico para $m=7$, e valor obtido da equação (3), respectivamente). Para cada valor de VDC , foram considerados todos os possíveis valores de estados terminais, desde o mínimo possível de 3 até o máximo de $VDC-1$. A tabela 3.3 mostra o número de palavras a serem memorizadas em cada caso.

número de esta- dos ter- minais	V D C					
	9	8	7	6	5	4
3	18	18	18	18	18	22
4	16	16	16	16	18	
5	2	2	2	4		
6	2	2	4			
7	0	2				
8	2					

Tabela 3.3

• O código MDVmB(m+1)B

Do padrão de comportamento observado na construção da tabela 3.3, derivamos um novo esquema de codificação: um código algorítmico situado num ponto intermediário entre os extremos representados pelos códigos alfabéticos - que utilizam tabelamento de todas as palavras - e os códigos totalmente algorítmicos - que excluem o uso de tabelas. Denominamos-lhe MDVmB(m+1)B, ou código mB(m+1)B com memória dependente de VDC. O nome traz consigo uma destacada característica do código: partindo de um nível de, digamos assim, complexidade permitida (tamanho da tabela - ou memória - disponível), obter o melhor desempenho em termos de VDC. Ou vice-versa, partindo de um desempenho desejado (VDC), obter o mínimo tabelamento.

O algoritmo para se obter estas informações, ou seja, para se construir uma tabela como a tabela 3.3 para qualquer valor de m, é o descrito pela seguinte seqüência de passos:

PASSO 1: Calcular $VDC_{max} = (3m-3)/2$, isto é, obter o resultado da expressão (4), que informa o valor de VDC para o tamanho da tabela igual à zero

PASSO 2: Obter $VDC_{min} = EsTerMin + 1$

Para m ímpar temos que EsTerMin vale:

- 2, para $m = 5$
- 3, para $17 \leq m \leq 7$
- 4, para $19 \leq m \leq 21$

onde: EsTerMin = menor número de estados terminais permitido

PASSO 3: para $VDC = VDC_{min}$ até VDC_{max}

para número_de_estados de $VDC_{max}-1$ até EsTerMin

para cada valor de SDC(0) possível

verifica_tabelamento;

onde: a ação verifica_tabelamento explicitada a seguir indicará quantas e quais dentre todas as palavras-fonte de m bits necessitarão ser codificadas por tabela. Seja:

$d(a)$ = disparidade de uma palavra de m bits dada por (2) e

$d(a_i)$ = disparidade de uma palavra até seu i -ésimo bit, $i=1, 2, \dots, m$
e assim:

$d(a) = d(a_i)$, quando $i=m$. Então temos:

verifica_tabelamento

se $d(a_i) > VDC/2 - SDC(0)$ ou
se $d(a_i) < -VDC/2 - SDC(0)$

ou

se $d(a) > \text{número_de_estados}/2 - SDC(0)$ ou
se $d(a) < -\text{número_de_estados}/2 - SDC(0)$

então

a palavra de informação será tabelada:

se $-d(a) > \text{número_de_estados}/2 - SDC(0) - 1$ ou
se $-d(a) < \text{número_de_estados}/2 - SDC(0) + 1$

De posse das informações obtidas pela aplicação do algoritmo acima, seleciona-se o par de valores desejado (VDC e número de estados terminais) e implementa-se a codificação através de um simples algoritmo, a saber:

PASSO 1: Leia palavra-fonte (**a**)

PASSO 2: Caso seja palavra da tabela, codifique **a** segundo a tabela e retorne ao PASSO 1. Caso contrário, execute o PASSO 3

PASSO 3: se $d(a_i) > VDC/2 - SDC(0)$ ou
se $d(a_i) < -VDC/2 - SDC(0)$

ou

se $d(a) > \text{número_de_estados}/2 - SDC(0)$ ou
se $d(a) < -\text{número_de_estados}/2 - SDC(0)$

então

codifique a aplicando a regra (B) - rebatimento - e retorne ao PASSO 1. Caso contrário, execute o PASSO 4

PASSO 4: Codifique a pela regra (A) e retorne ao PASSO 1

No receptor a operação de decodificação seguirá o seguinte algoritmo:

PASSO 1: Leia a palavra-código

PASSO 2: Caso seja palavra da tabela, decodifique-a segundo a tabela e retorne ao PASSO 1. Caso contrário, execute o PASSO 3

PASSO 3: Decodifique-a segundo as regras (A') ou (B') e retorne ao PASSO 1

Para fazer a análise das características de transmissão dos códigos $MDV_mB(m+1)B$, consideramos somente o caso de $m=7$. É o caso mais frequente na literatura e, portanto, podemos fazer comparações mais concretas, tratadas nos capítulos seguintes.

Capítulo 4

Propriedades de transmissão

• Introdução

O propósito deste capítulo é verificar as características de transmissão dos códigos desenvolvidos neste trabalho e apresentados anteriormente. No apêndice 2B já tratamos do fenômeno da extensão de erros e da recuperação do sinal de relógio. Agora analisaremos a densidade espectral de potência, a monitorabilidade da taxa de erros e a questão do sincronismo.

Estas características nos permitem mais ampla avaliação destes novos esquemas de codificação aqui propostos. Faremos isto, comparando com os resultados obtidos para códigos já em uso, ou conhecidos da literatura específica.

• Análise espectral

Para obtermos o diagrama de densidade espectral de potência dos códigos, utilizamos o método desenvolvido por [CARIOLARO 74]. A vantagem deste "algoritmo" para a obtenção do espectro de potência é que o resultado é uma forma fechada, analítica, precisa, e não apenas uma estimativa obtida por simulação. Sua dificuldade é o esforço computacional elevado, que foi amenizado com a utilização do software MATLAB¹. Consideramos as palavras-fonte equiprováveis.

¹ O MATLAB é marca registrada da The Math Works Inc.

A densidade espectral de potência normalizada de um código é dada por [CARIOLARO 74]:

$$D(f) = \frac{C(f) \cdot T}{|P(f)|^2} = X_c(f) + X_d(f)$$

onde:

- C(f) = espectro de potência
- T = intervalo elementar do sinal digital
- P(f) = transformada de Fourier do pulso transmitido
- X_c(f) = componente contínua
- X_d(f) = componente discreta

A componente discreta do espectro de potência é a mesma para os códigos PFmB(m+1)B e MDVmB(m+1)B e está mostrada na figura 4.1 para m=7.

O gráfico da citada figura apresenta a envoltória que multiplica os impulsos unitários $\delta(f - n/(m+1)T)$, para $n=0, 1, 2, \dots$. Verifica-se, analisando esta figura, que nas frequências dos impulsos a envoltória é nula. Portanto, a componente discreta de tais códigos também é nula, refletindo a complementariedade dos alfabetos progressivos e regressivos nos dois casos.

As componentes contínuas dos espectros de potência dos códigos PF7B8B e alguns códigos MDV7B8B são mostrados nos gráficos X_c versus fT da figura 4.2. É bom notar que o código MDV7B8B para 7 estados terminais e VDC=9 é o próprio MPF7B8B. Verifica-se o nulo em $f=0$, como era de se esperar, sendo todos estes códigos balanceados. Observa-se também, que a potência média é praticamente a mesma e o diagrama da densidade espectral de potência para o código MDV7B8B com 3 estados terminais e VDC=5 é mais plano e tem a subida suave, confirmando o esperado, já que, como visto anteriormente, para menores valores de VDC, menor é o conteúdo espectral nas frequências próximas a zero.

Para o caso de acoplamento capacitivo, haverá o corte nas baixas frequências. Analisando essa região com mais detalhes constatou-se um interessante fato, mostrado pela figura 4.3: apesar de apresentar VDC menor que o PF7B8B, o MPF7B8B possui densidade espectral de potência superior até 1,8% da faixa. Fizemos a integral das curvas das densidades espectrais de ambos códigos e verificamos que: até 4% da faixa, a energia do PF7B8B é inferior à do MPF7B8B (cfr. figura 4.4) e, como na prática o corte do filtro passa altas está em torno deste ponto, a probabilidade de erro devido ao valor RMS da flutuação (provocada por este corte) na linha de base do sinal é menor

para o PF7B8B. Com relação ao erro devido ao valor de pico desta flutuação - que representa o pior caso -, a vantagem fica com o MPF7B8B, pois tem VDC menor.

No caso dos códigos MDV7B8B com VDC inferior a 9, o desempenho é superior a ambos os considerados acima, como pode ser observado diretamente nos gráficos da figura 4.2.

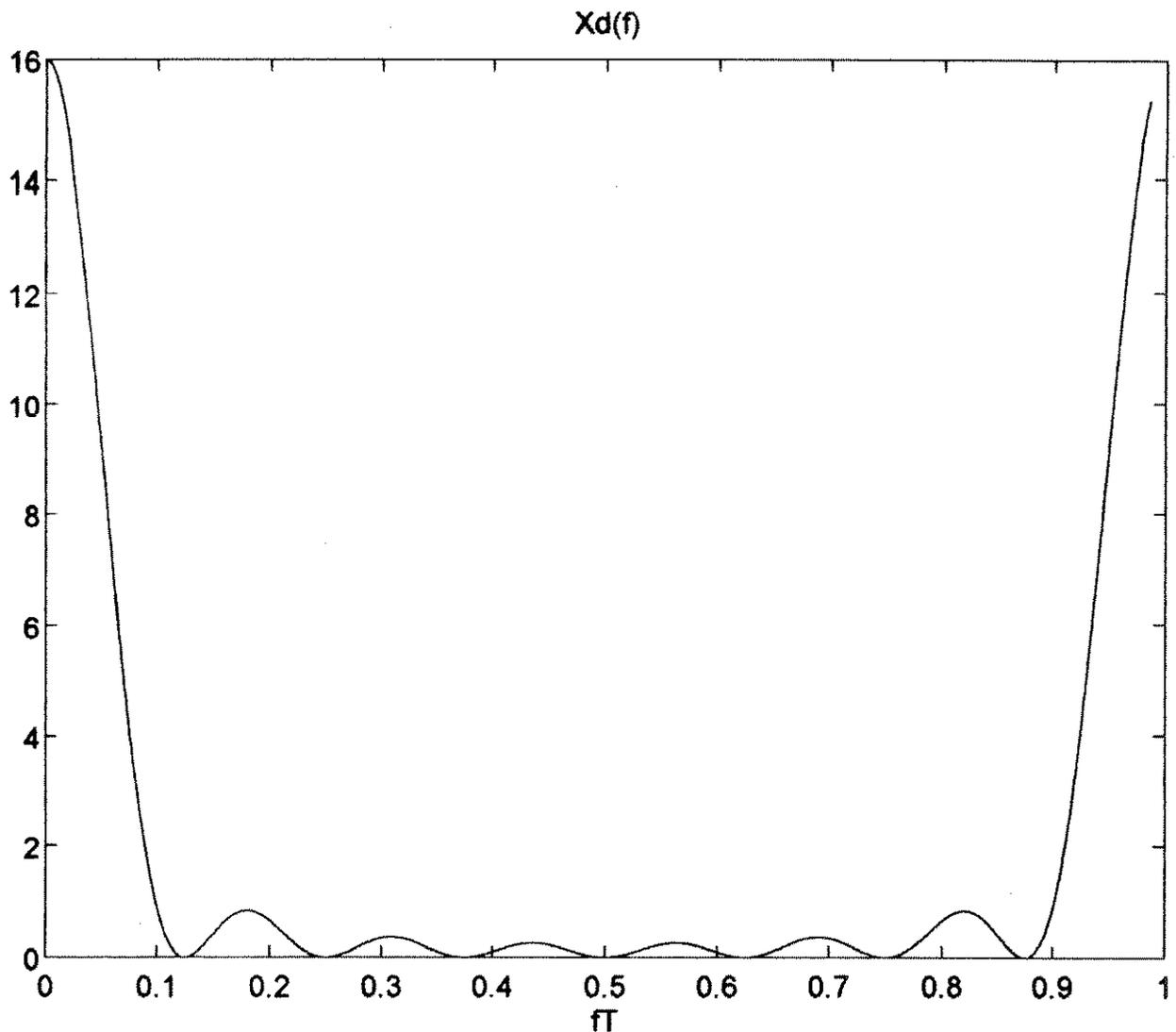


Figura 4.1 - Componente discreta dos espectros de potência dos códigos considerados.

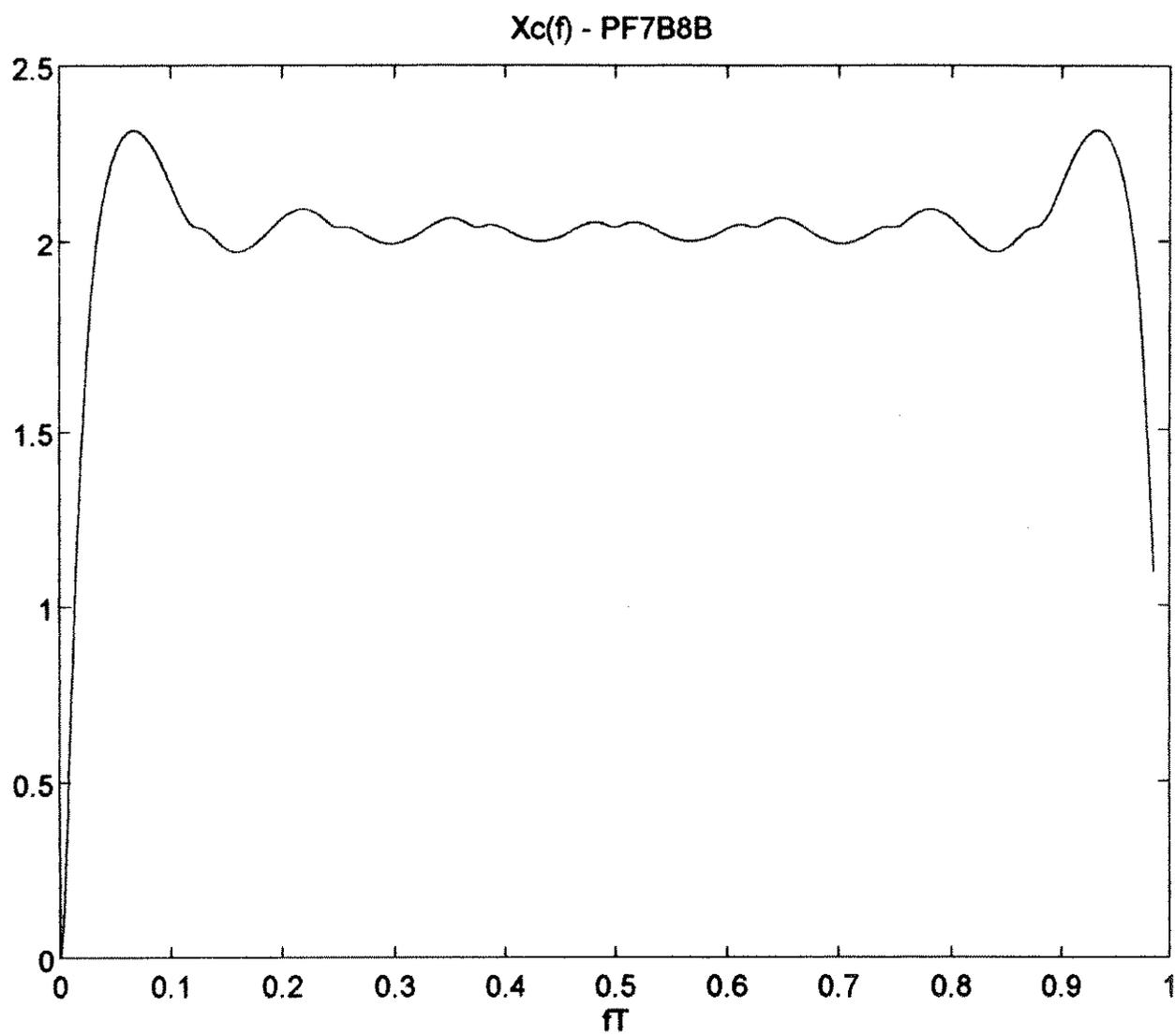


Figura 4.2.a - Componente contínua do espectro de potência do código PF7B8B.

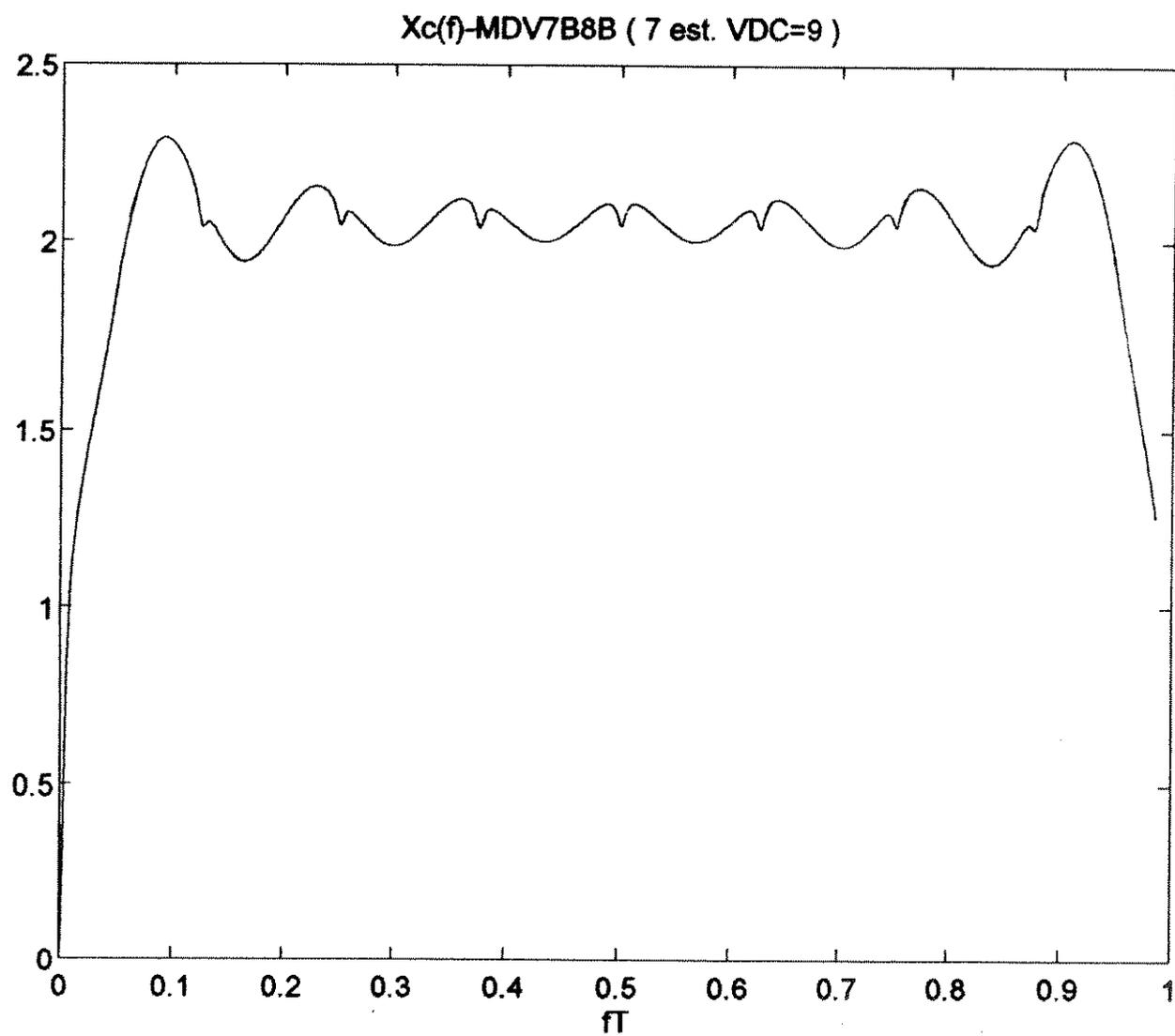


Figura 4.2.b - Componente contínua do espectro de potência do código MPF7B8B.

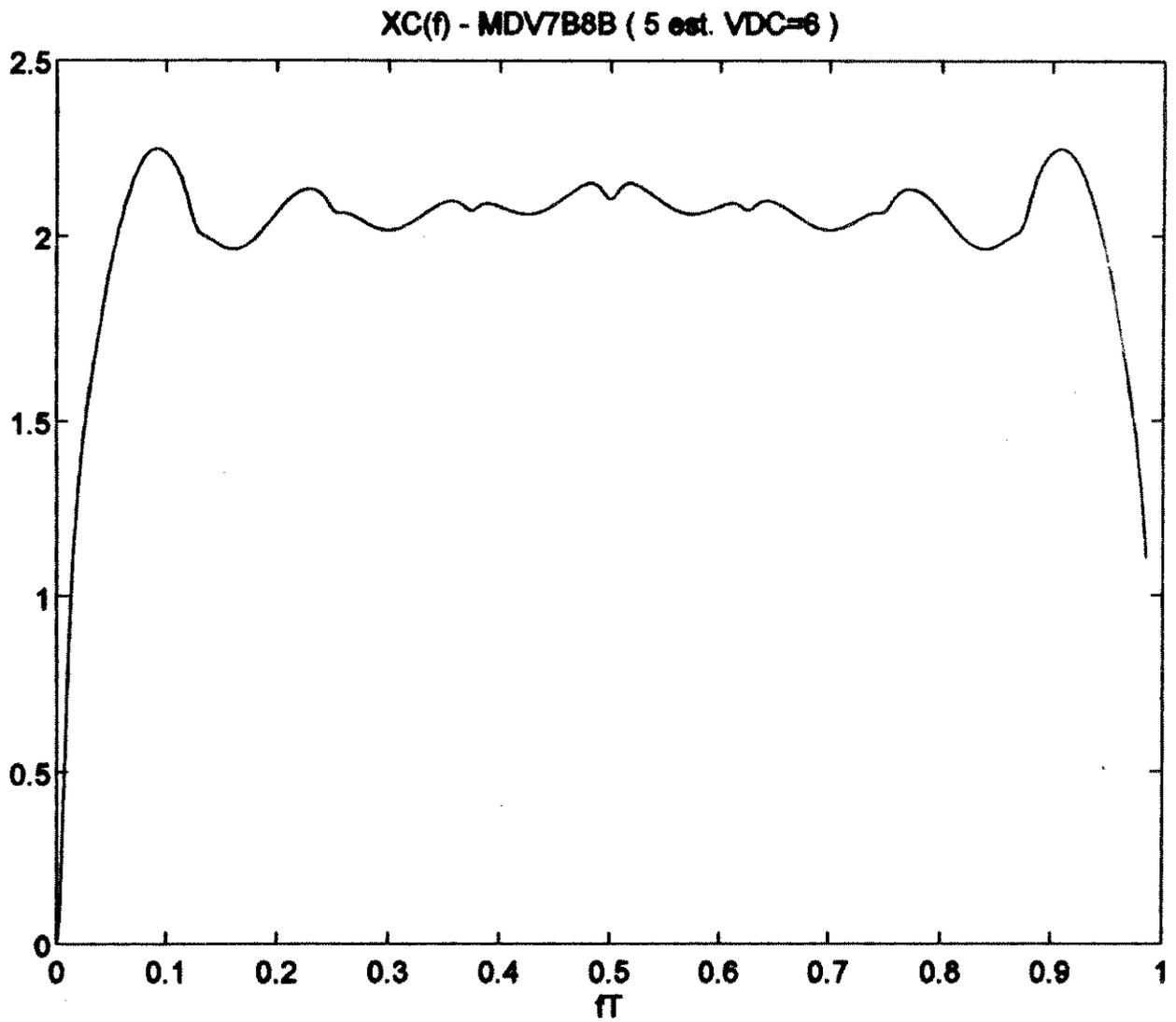


Figura 4.2.c - Componente contínua do espectro de potência do código MDV7B8B.

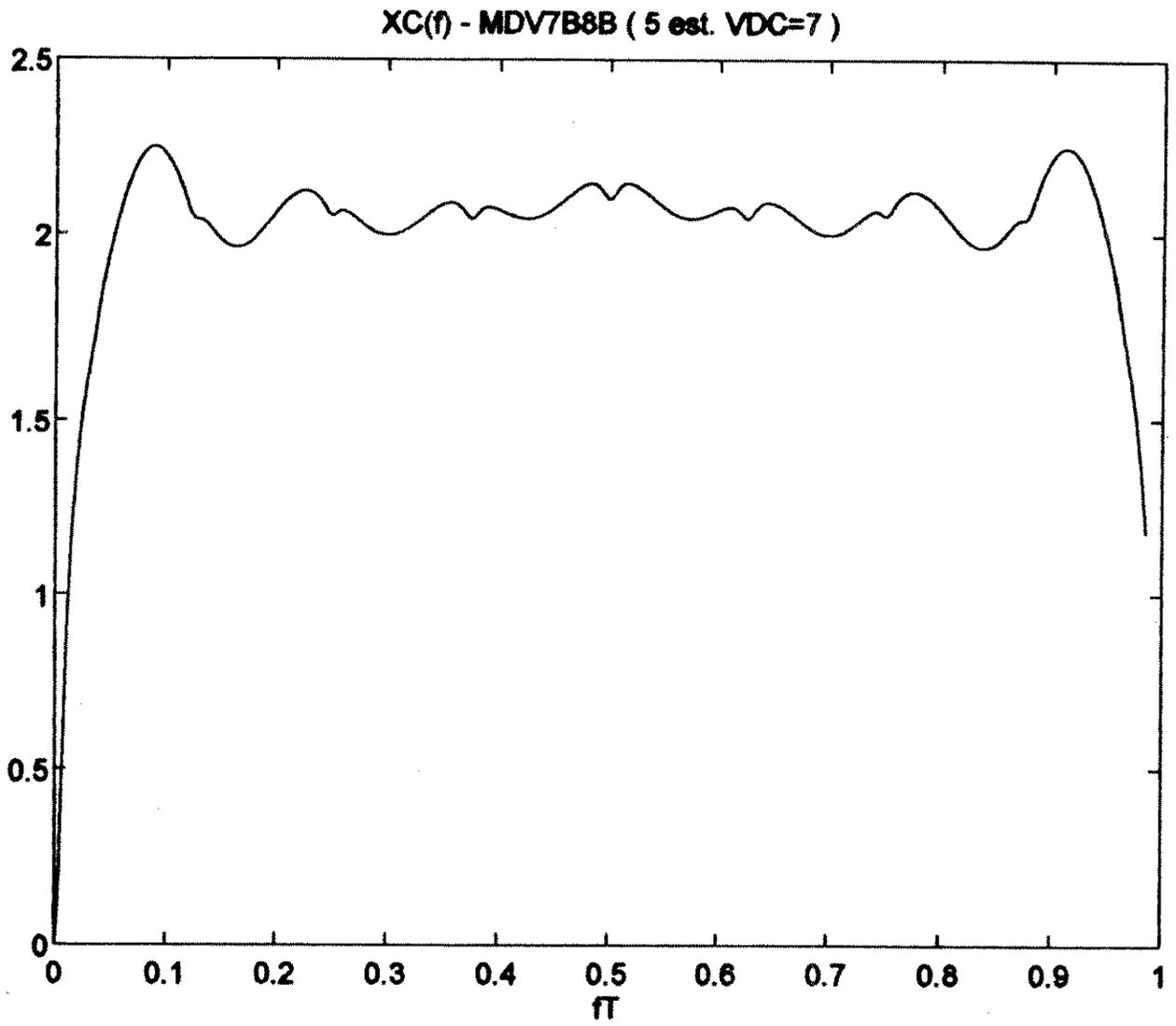


Figura 4.2.d - Componente contínua do espectro de potência do código MDV7B8B.

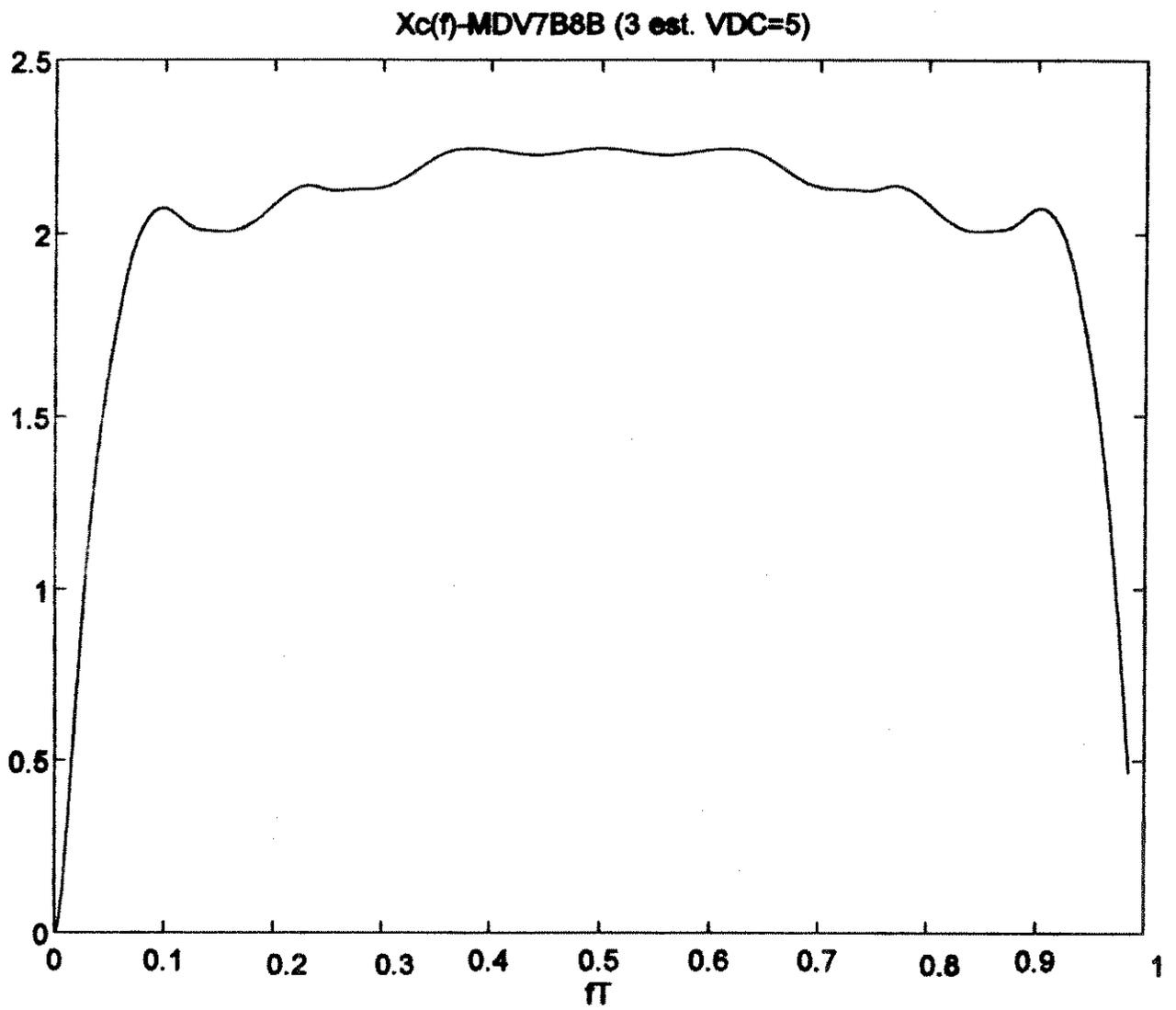


Figura 4.2.e - Componente contínua do espectro de potência do código MDV7B8B.

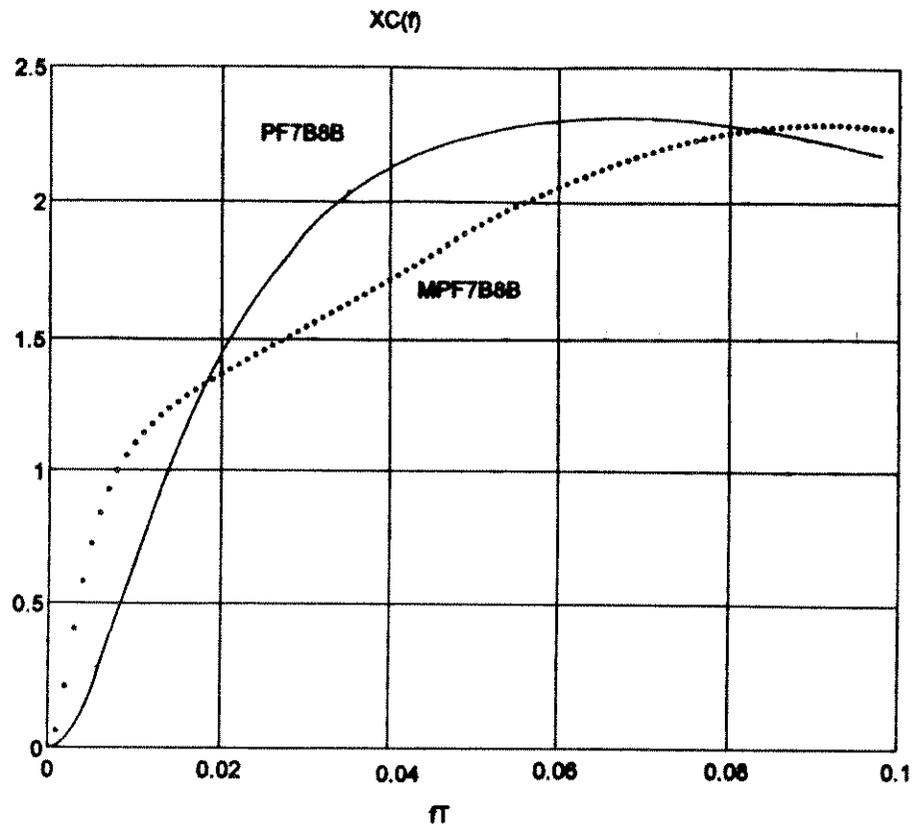


Figura 4.3 - Detalhe da superposição das componentes contínuas dos espectros dos códigos PF7B8B e MPF7B8B

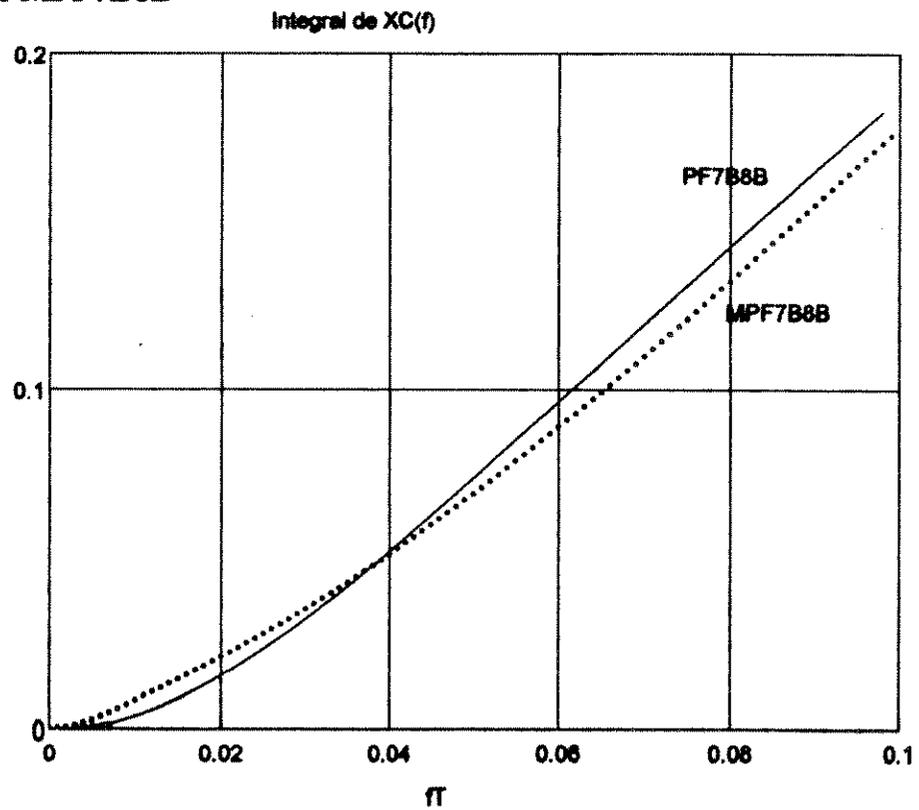


Figura 4.4 - Detalhe da superposição das integrais das curvas de $XC(f)$ dos códigos PF7B8B e MPF7B8B

• Monitoração da taxa de erros

Atualmente a tendência para o controle de erros de transmissão em troncos ópticos está sendo transferida às camadas mais elevadas de um sistema, isto é, ao nível de usuário, pois são diversos os tipos de informação que transitam por estes troncos: telefonia, som, vídeo, dados. E, como são diferentes as tolerâncias a erros para cada um dos tipos citados, torna-se complexo um controle abrangente a todos.

Os códigos de que estamos tratando, restringem-se à camada física e, com o exposto acima, não seria necessário então que estas codificações proovessem a capacidade de controle de erros. Porém isto é feito. Não com o objetivo de correção dos erros ou outros quaisquer, mas para garantir o desempenho de transmissão como, por exemplo, a decisão sobre o sincronismo.

As características dos códigos aqui apresentados permitem que facilmente se monitore os erros na linha. Um modo imediato é um contador tipo "up-down" que é incrementado para cada marca ("1" lógico) recebida e decrementado para cada espaço ("0" lógico). Se a variável C representar o valor deste contador, toda vez que $|C| > VDC$, sabe-se que ocorreu um erro na linha, pois estes códigos têm a SDC confinada, o que implica que um erro isolado causará em algum momento a violação dos limites de confinamento.

Existem vários outros modos de realizar a monitoração. Quisemos com o exposto acima apenas exemplificar o propósito deste item: notificar que os códigos $MPFmB(m+1)B$ e $MDVmB(m+1)B$ possibilitam que se monitore os erros de linha tantos nos repetidores intermediários, como nos receptores, sem dificuldades práticas.

• Análise de sincronismo

Um código de linha, cujas palavras-código possuem n bits, é denominado auto-sincronizável se o receptor consegue selecionar n bits adjacentes na fase correta, que é uma dentre as n possíveis sem nenhuma informação adicional, ou seja, contando apenas com a seqüência binária codificada transmitida. Isto é possível porque estando fora de fase (desalinhado) com o transmissor, aspectos estruturais do código não estarão presentes na seqüência decodificada: serão observadas palavras ilícitas, estados e seqüências de estado que não correspondem ao esquema de codificação utilizado. Com base nestas violações é executada uma estratégia de alinhamento que

informará se o sistema está desalinhado e, em caso afirmativo, tentará recuperar a sincronização o mais rapidamente possível.

É preciso discernir, contudo, se as violações observadas não são devidas a simples erros de linha ou se ocorreu realmente o desalinhamento.

Uma boa estratégia de alinhamento apresenta as seguintes características:

a) Tempo de retenção de alinhamento (T_{ret}) - tempo para acusar uma perda de alinhamento falsa, causada por erros de linha - muito elevado, da ordem de centenas de anos.

b) Tempo de recuperação de alinhamento (T_{rec}) - tempo que o receptor consome para alinhar-se novamente após uma perda de sincronismo real - muito pequena, da ordem de algumas palavras-código.

Não é nosso propósito apresentar nenhuma estratégia de alinhamento em concreto, pois a escolha dessa estratégia é uma função do projetista do sistema. Mostraremos a seguir apenas que os códigos que desenvolvemos apresentam características tais que o tornam auto-sincronizável, ou seja, que existem estratégias possíveis para obter valores aceitáveis para T_{ret} e T_{rec} .

Seja n o número de bits da palavra-código e $P_v(d=i)$ a probabilidade de se detectar uma violação com o receptor defasado i bits do transmissor, onde $0 \leq i \leq n$. Então um código será auto-sincronizável se apresentar:

a) $P_v(d=0)$ muito pequena.

Sendo P_e a taxa de erros na linha, para $P_e=0$, $P_v(d=0)$ é nula também. Na prática P_e é baixa. O sistema é considerado gravemente alarmado com $P_e=10^{-3}$, pois a situação normal de operação é para $P_e=10^{-9}$.

b) $P_v(d \neq 0)$ deve ser muito elevada.

Estando o sistema desalinhado, o número de violações detectado deve ser muito maior que o número de violações encontradas para o sistema sincronizado na presença de elevada taxa de erros na linha. Assim é possível separar os falsos dos verdadeiros alarmes indicativos da ausência de sincronismo.

Os valores de T_{ret} e T_{rec} são funções (particulares para cada estratégia de alinhamento), de $P_v(d=0)$ e $P_v(d \neq 0)$. A tabela 4.1 mostra alguns valores de $P_v(d \neq 0)$

encontrados para os códigos apresentados neste trabalho e, para efeitos de comparação, também os resultados para outros códigos.

Para a confecção desta tabela foram utilizadas os programas CODJM e BOLJM [ROMANO 84], que calculam P_v ($d \neq 0$) analisando as violações por palavras proibidas e infrações da lei do código, respectivamente.

d	5B6B JM		MDV 2E3V		MDV 2E4V		MDV 4E5V		3B4B CC		ELO 34	
	P_v	\tilde{N}	P_v	\tilde{N}	P_v	\tilde{N}	P_v	\tilde{N}	P_v	\tilde{N}	P_v	\tilde{N}
1	0,16	7	0,11	10	0,18	6	0,33	3	0,25	4	0,34	3
2	0,16	7	0,14	8	0,19	6	0,34	3	0,06	16	0,31	4
3	0,17	6	0,11	9	0,15	7	0,28	4	0,25	4	0,31	4
4	0,16	7	0,14	8	0,19	6	0,31	3				
5	0,16	7	0,12	9	0,17	6	0,26	4				

Tabela 4.1.a - Análise por palavras proibidas.

d	5B6B JM		MDV 2E3V		MDV 2E4V		ELO 34	
	\tilde{N}	Nm	\tilde{N}	Nm	\tilde{N}	Nm	\tilde{N}	Nm
1	7	23	9	26	7	25	11	38
2	8	34	11	46	8	34	8	34
3	6	20	7	23	6	22	7	25
4	8	34	8	34	7	27		
5	7	23	7	23	7	22		

Tabela 4.1.b - Análise por violação da lei do código.

onde:

\tilde{N} e Nm são os números médio e máximo, respectivamente, de palavras-código necessárias para se detetar uma violação;

5B6B JM - código apresentado em [ROMANO 84]²;

MDV xEyV - códigos MDV5B6B com x estados terminais e VDC=y;

ELO 34 - código proposto para o sistema homônimo [ALCÂNTARA 88]²;

3B4B CC - código apresentado em [CCITT 79].

² Não foram considerados os alfabetos dos canais de serviço.

Alguns comentários sobre as tabelas acima: por causa da estrutura do programa BOLJM, não foi possível obtermos os resultados para o MDV5B6B com 4 estados terminais e $VDC=5$. Porém, observa-se que, embora havendo diferenças entre as duas formas de análise (violação por palavra proibida e violação da lei do código), estas não são tão consideráveis. A exceção é o código 3B4B [CCITT 79], que possui $VDC=2$, e não é sincronizável por violação da lei do código. Este também não seria sincronizável por palavra proibida caso se utilizasse canais de serviço. Foi por este motivo que no sistema ELO 34 optou-se por um código com VDC maior: perdeu-se em desbalanço máximo permitido para se ganhar em sincronizabilidade.

Este fato da relação de proporcionalidade inversa entre VDC e capacidade de sincronismo também pode ser observada nos códigos $MDVmB(m+1)B$: há um crescimento nos valores de P_v - desejado para melhor capacidade de realinhamento - com a degradação dos valores de VDC.

Enfim, podemos concluir que a diferença entre $P_v(d=0)$ - que é da ordem de grandeza da taxa de erros na linha P_e - e os valores de $P_v(d \neq 0)$ encontrados é tal que os códigos $MDVmB(m+1)B$ são sincronizáveis por alguma estratégia de alinhamento montada sobre estes resultados. É bom notar que não houve de nossa parte preocupação alguma de maximizar valores de $P_v(d \neq 0)$ através da escolha apropriada das palavras-código que, para cada caso, necessitavam ser tabeladas, e sobre as quais se tem certo grau de liberdade. Ou seja, é possível atingir melhores resultados para a sincronizabilidade, ficando em aberto para futuros estudos determinar o conjunto de palavras-código utilizadas para a codificação por tabelas que otimizem o sincronismo dos códigos $MDVmB(m+1)B$.

CAPÍTULO 5

CONCLUSÕES

Foi realizado um estudo sobre vários esquemas de codificação de linha para sistemas ópticos de alta capacidade, nos quais a utilização das tradicionais técnicas de memorização em ROM's das palavras-código torna-se, por razões tecnológicas, problemática.

Analizamos várias propostas de codificação recentes que encaminham soluções para este problema. Detivemo-nos mais sobre os códigos PFmB(m+1)B, interessante técnica que realiza a codificação por simples operações algorítmicas sobre a seqüência binária da fonte de informação, resultando num código balanceado, que dispensa o uso de tabelas memorizadas e de fácil implementação, ou seja, próprio para altas taxas de transmissão.

Desenvolvemos os códigos MPFmB(m+1)B, uma versão modificada do apresentado por [KRZYMIEN 89], que possui as seguintes características: é algorítmico, dispensa o uso de tabelas memorizadas, é sincronizável, possui as mesmas complexidade e propriedade de recuperação de relógio que o PFmB(m+1)B, e vantagem com relação a este no máximo desbalanço permitido.

Numa posição intermediária entre os códigos que não se utilizam de tabelas memorizadas e os que utilizam, encontram-se os códigos MDVmB(m+1)B, que também desenvolvemos neste trabalho. É um código algorítmico, cujas vantagens são a pequena complexidade e grande redução no tamanho da tabela a ser memorizada.

Poderia se pensar que, fazendo uso de tabelas, este não se configurasse como um código propício para alta velocidade de transmissão. No entanto, verifica-se através da tabela 3.3 que o número de palavras que devem ser memorizadas é bem menor que as 128 palavras-fonte a serem codificadas. E também deve ser considerada

a contrapartida oferecida: menores valores de VDC em comparação com os códigos totalmente algorítmicos acima citados. A disponibilidade de hardware capaz de operar na velocidade dimensionada para o sistema, o máximo desbalanço desejado e outros fatores específicos determinarão as decisões de projeto dentro do contexto analisado neste trabalho.

Em conclusão, foram discutidas as soluções do estado-da-arte para a codificação de linha em sistemas ópticos de alta capacidade, sendo apresentadas algumas alternativas novas com vantagens aparentes de desempenho.

APÊNDICE A1

Listagem do programa utilizado no MATLAB

```
%function mdv7e9v

clc
m=7;
n=8;
nstat=7;
k=2^m;
qu=1/k;
bilhao=10;

load a7e9vm.dat
A = a7e9vm;
clear a7e9vm

load emd7e9vm.dat
E = emd7e9vm;
clear emd7e9vm

%zerar MATPI

MATPI = zeros(1:nstat,1:nstat);
for u = 0:(k-1)
    li = u*nstat + 1;
    lf = li + nstat - 1;
    EU = E(li:lf,:);
    MATPI = MATPI + qu*EU;
end
```

```

PIINF = MATPI;

for jota = 1:bilhao
    PIINF = PIINF*PIINF;
end

DINF = PIINF;
DINF = diag(diag(DINF));

%ZERAR R0 C1 C2

R0 = zeros(n);
C1 = zeros(nstat,n);
C2 = C1;

for u = 0:(k-1)
    li = u*nstat + 1;
    lf = li + nstat - 1;
    EU = E(li:lf,:);
    AU = A(li:lf,:);
    EUTR = EU';
    AUTR = AU';
    C1 = C1 + qu*EUTR*DINF*AU;
    C2 = C2 + qu*AU;
    R0 = R0 + qu*AUTR*DINF*AU;
end
C1TR = C1';
RINF = C1TR*PIINF*C2;
doispi = 2*pi;
ft = 0:.001:1;
U = eye(size(MATPI));
PIMPIN = MATPI - PIINF;
UMPIN = U - PIINF;
RMRIN = R0 - RINF;
X1 = 0*(0:1000);
X2 = X1;
XD = X1;
X2RE = X1;
XC = X1;

```

```

for iter = 0:1000
    arg = (doispi*n*(ft(iter+1)));
    w = sin(arg+(pi/2)) + i*sin(arg);
    OMGU = inv(w*U - PIMPIN);
    SIG2 = C1TR*UMPIN*OMGU*C2;
    V = 0*(1:n);
    V(1) = 1;
    argu = (doispi*ft(iter+1));
    v = sin(argu+(pi/2)) + i*sin(argu);
    V(1)=v;

    for var = 2:n
        V(var) = V(var-1)*v;
    end

    VTR = V;
    X1(iter+1) = V*RMRIN*VTR;
    X2(iter+1) = V*SIG2*VTR;
    XD(iter+1) = V*RINF*VTR;
    X2RE(iter+1) = real(X2(iter+1));
    XC(iter+1) = X1(iter+1) + 2*X2RE(iter+1);
end

plot(ft,XC), title('Xc(f) - MDV7B8B (7 est. 9 VDC)'), xlabel('fT')
pause
disp(XC(1))
plot(ft,XD), title('Xd(f) - MDV7B8B (7 est. 9 VDC)'), xlabel('fT')
disp(XD(1))
end

```

REFERÊNCIAS

[ALCÂNTARA, 88] - ALCÂNTARA, H. J. M. de. Códigos de bloco binários de faixa mínima para comunicações via fibras ópticas. Campinas, 1988. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas.

[ALI, 86] - ALI, A. M. Multiplexing and line coding schemes for broadband ISDN. In: PROC. INT. CONF. COMMUN. , Toronto, Ont., Canada, p. 371-375, june 1986.

[BI, 89] - BI, G. ; JONES, E. U. Line code conversion for high speed optical transmission systems-some comparisons. International Journal of Electronics, v. 67, n. 3, p. 365-376, 1989.

[BROOKS, 83] - BROOKS, R. M. ; JESSOP, A. Invited paper: line coding for optical fibre systems. International Journal of Electronics, v. 55, n. 1, p. 81-120, 1983.

[CARIOLARO, 74] - CARIOLARO, G. L. ; TRONCA, G. P. Spectra of block coded digital signals. IEEE Transactions on Communications, v. COM-22, n. 10, p. 1555-156, oct. 1974.

[CARTER,65] - CARTER, R. O. Low-disparity binary coding system. Electronics Letters, v. 1, n. 3, p. 67-68, may 1965.

[CATTERMOLE, 83] - CATTERMOLE, K. W. Invited paper: principles of digital line coding. International Journal of Electronics, v. 55, n. 1, p. 3-33, 1983.

[CAVERS, 92] - CAVERS, J. K. ; MARCHETTO, R. F. A new coding technique for spectral shaping of data. IEEE Transactions on Communications, v. 40, n. 9, p. 1418-1422, sept. 1992.

[CCITT, 79] - CCITT. Binary line codes for digital transmission on optical fibers. Study Group 18, 1979. (Contribution n. 291)

[FAIR, 91] - FAIR, I. J. et al. Guided scrambling: a new line coding technique for high bit rate fiber optic transmission systems. IEEE Transactions on Communications, v. 39, n. 2, p. 289-297, feb. 1991.

[HOSOE, 83] - HOSOE, P. T. Codificação digital para Comunicações ópticas. Campinas, 1983. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia, Universidade Estadual de Campinas.

[JUSTESEN, 82] - JUSTESEN, J. Information rates and power spectra of digital codes. IEEE Transactions on Information Theory, v. IT-28, p. 457-472, 1982.

[KRZYMIEN, 89] - KRZYMIEN, W. A. Transmissions performance analysis of a new class of line codes for optical fiber system. IEEE Transactions on Communications, v. 37, n. 4, p. 402, apr. 1989.

[REC.G702] - CCITT. Vocabulary of Pulse Code Modulation (PCM) and Digital Transmission Terms. Recommendation G.702, v. 3, 4 - blue book.

[ROMANO, 84] - ROMANO, J. M. T. Códigos balanceados para transmissão digital por fibra óptica. Campinas, 1984. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia, Universidade Estadual de Campinas.

[SALLES, 92] - SALLES, A. A. A. de. Progressos recentes em comunicações óticas no Brasil. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 10. , Brasília, 1992. Anais...

[SLOMINSKI, 89] - SLOMINSKI, M. M. mB1b code and its sinchronizability. In: URSI INTERNATIONAL SYMPOSIUM ISSSE, 1989. Proceedings...

[SLOMINSKI, 92] - SLOMINSKI, M. M. ; WYSOCKI, T. ; HIROSAKI, B. Performance of the MDM-7B8B code for fibre optic links. In: URSI INTERNATIONAL SYMPOSIUM ISSSE, Paris, 1992. Proceedings...

[WALDMAN, 92] - WALDMAN, H. ; BATISTA, M. A. C. Códigos de linha para sistemas ópticos de alta velocidade. In: SIMPÓSIO

BRASILEIRO DE TELECOMUNICAÇÕES, 10. , Brasília, 1992.
Anais...

[WYSOCKI, 89] - WYSOCKI JR. ,T. Easy encodable mBnB codes.
In: URSI INTERNATIONAL SYMPOSIUM ISSSE, 1989.
Proceedings...

[YOSHIKAI, 84] - YOSHIKAI, N. ; KATAGIRI, K. ; TAKESHI, I.
MB1C code and its performance in an optical communication system.
IEEE Transactions on Comuunications, v. COM-32, n. 2, p. 163-168,
feb. 1984.

[YOSHIKAI, 86] - YOSHIKAI, N. ; NISHI, S. ; YAMADA, J. Line
code and terminal configuration for very large-capacity optical
transmission system. IEEE Journal on Selected Areas in
Communications, v. SAC-4, n. 9, p. 1432-1437, dec. 1986.