

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

BALANCEAMENTO, FAIXA MÍNIMA E EFICIÊNCIA EM CÓDIGOS TERNÁRIOS

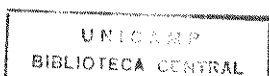
Este exemplar corresponde à
redução final da tese defendida
por Margarete Mitiko Iramina e
aprovada pela Comissão Julgadora
em 30 de agosto de 1989.

30/8/89. Hélio Waldman

MARGARETE MITIKO IRAMINA
Orientador: Prof. Doutor HÉLIO WALDMAN

Tese apresentada à Faculdade de Engenharia Elétrica
da Universidade Estadual de Campinas - UNICAMP,
como parte dos requisitos exigidos para a obtenção do
título de MESTRE EM ENGENHARIA ELÉTRICA.

AGOSTO - 1989



Resumo

Neste trabalho os conceitos de *Balanceamento e Faixa Mínima* são aplicados em códigos de bloco ternários. Discute-se também a existência e a eficiência de códigos ternários simplesmente balanceados, de faixa mínima e balanceados de faixa mínima (códigos que têm as duas propriedades conjugadas).

Verifica-se que é possível construir códigos com características semelhantes às dos códigos de resposta parcial, no que se refere à faixa mínima, com vantagens adicionais de redução de taxa de símbolos e proibição de longas seqüências de zeros. O desempenho destes códigos é avaliado em função da abertura do diagrama de olho que é obtido através de um método computacional baseado no algoritmo de Viterbi. Valores assintóticos de eficiência são discutidos para diversas condições de balanceamento e ou faixa mínima.

Aos meus pais
José Iramina e Fugiko Iramina

Agradecimentos

Impossível agradecer aqui a todos aqueles que conviveram comigo, através deste trabalho, e colaboraram para a sua realização. O que segue aqui é apenas um ponto de partida. Pela oportunidade, pelas idéias e dedicação, meu profundo agradecimento ao *Professor Doutor Hélio Waldman*; ao *Engenheiro Ralph Robert Heinrich*, Coordenador de Áreas de Sistemas Ópticos do CPqD/TELEBRÁS, e a todos os companheiros de trabalho, que muito me apoiaram e me incentivaram; ao *Anderson e Cladir*, que me ajudaram na parte computacional; à *Engenheira Mônica Lacerda Rocha*, pelos desenhos e pelo apoio; e a muitos outros, não mencionados aqui, porém não esquecidos, que muito me ajudaram.

Índice

1	INTRODUÇÃO	1
2	CÓDIGOS TERNÁRIOS BALANCEADOS	4
2.1	Introdução	4
2.2	Condição de Balanceamento	6
2.3	Exemplos	7
2.3.1	Código <i>AMI</i>	7
2.3.2	Classe <i>4B3T</i>	7
2.3.3	<i>MS</i> – 43	8
2.4	Máximo Desbalanço <i>P</i> e Máxima Redução de Velocidade	9
2.5	Tabelas	11
2.6	Limites Fundamentais de Eficiência	18
3	CÓDIGOS TERNÁRIOS DE FAIXA MÍNIMA	27
3.1	Introdução	27
3.1	Condição Generalizada para Faixa Mínima	33
4	CÓDIGOS TERNÁRIOS BALANCEADOS DE FAIXA MÍNIMA	47
4.1	Condição de Balanceamento e Faixa Mínima	47
4.2	Existência de Códigos Balanceados e de Faixa Mínima	51
5	CÁLCULO DO DIAGRAMA DE OLHO	63
5.1	Introdução	63
5.2	O Algoritmo de Viterbi	63
5.3	Cálculo do Diagrama de Olho	64
5.4	Programas	67
5.5	Resultados	85
6	CONCLUSÕES E COMENTÁRIOS	91
	APÊNDICE 1	92
	APÊNDICE 2	94
	APÊNDICE 3	96
	BIBLIOGRAFIA	100

Capítulo 1

INTRODUÇÃO

Geralmente, em sistemas de transmissão digital, a informação é submetida à codificação de linha, que tem a função de produzir propriedades adequadas à transmissão. As propriedades mais importantes que um código de linha pode proporcionar são:

- *Balanceamento*, para minimizar flutuações no nível de base do sinal causadas pelo bloqueio das baixas frequências devido à presença de transformadores e capacitores de acoplamento no sistema que cortam as baixas frequências;
- *Proibição de longas seqüências sem transição de nível*, para possibilitar a aquisição da informação sobre a temporização dos dígitos pelos circuitos de recuperação de relógio;
- *Mínima ocupação da faixa espectral*, para maximizar a relação Sinal/Ruído nos instantes de amostragem;
- *Máxima abertura do diagrama de olho*, para minimizar a degradação de desempenho causada pelo tremor (jitter) do relógio recuperado ou por erros estáticos de temporização;
- *Monitorabilidade de taxa de erros*, feita a partir da taxa de violações do código;
- *Decodificabilidade independente do estado do codificador*;
- *Máxima capacidade de informação*;
- *Máxima sincronizabilidade de blocos*, quando forem utilizados códigos de blocos;
- *Mínima propagação de erros*;

Alguns dos requisitos acima são conflitantes entre si, fazendo com que o projeto de um código considerado ideal esteja associado a um compromisso entre as propriedades desejadas. Tal compromisso varia de acordo com o meio de transmissão utilizado, a capacidade do sistema e com o estado-da-arte da tecnologia envolvida na realização do mesmo.

Quanto à faixa espectral, existe um valor teórico mínimo que foi proposto por Nyquist^[1], $f_n = 1/2T$, onde T é o intervalo de sinalização. A máxima economia de faixa espectral resulta da possibilidade de operar com excesso de faixa nulo e, mesmo assim, manter o diagrama de olho aberto.

Os códigos de resposta parcial^[2] apresentam algumas das propriedades mencionadas, como por exemplo, a minimização da ocupação da faixa espectral. A Fig.1.1 mostra

CAPÍTULO 1 INTRODUÇÃO

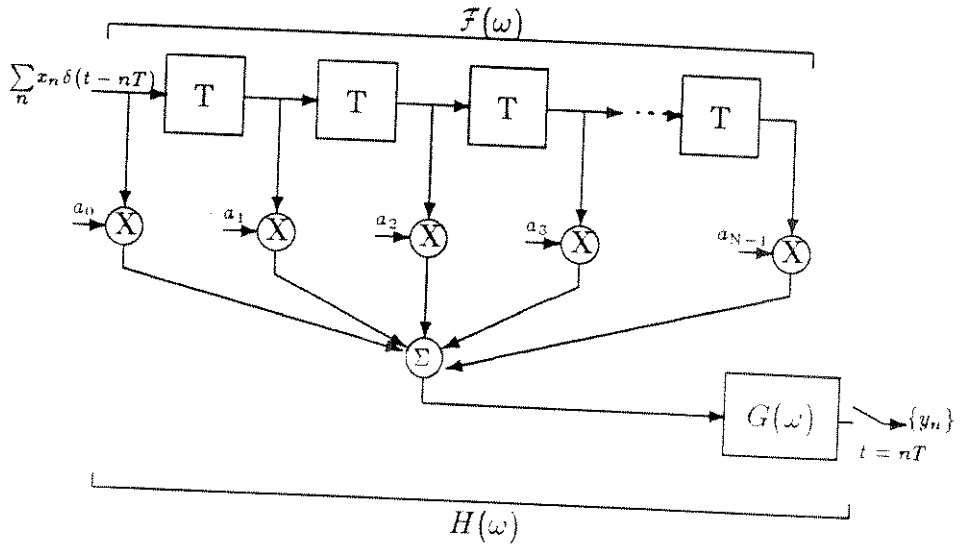


Figura 1.1: Modelo de um sistema de resposta parcial genérico

um esquema para gerar a função $H(\omega)$ dos sistemas de resposta parcial. O sistema constitui-se de uma linha de atraso com derivações com coeficientes $\{a_n\}$ em cascata com o filtro $G(\omega)$. O filtro transversal tem a resposta periódica dada por:

$$\mathcal{F}(\omega) = F(D) \Big|_{D=e^{-j\omega T}} = \sum_{n=0}^{N-1} a_n e^{-jn\omega T}$$

onde T é o intervalo entre símbolos.

Desta forma a resposta impulsiva de $H(\omega)$, $h(t)$, assumirá valores das amostras $\{a_n\}$ se e somente se $G(\omega)$ satisfizer o critério de Nyquist[1], ou seja:

$$\sum_{k=-\infty}^{\infty} G(\omega - \frac{k2\pi}{T}) = T$$

Pode-se dizer então que os códigos de resposta parcial usam, basicamente, combinações de seqüências binárias iguais atrasadas entre si, dando origem a um sinal multinível. As operações envolvidas nesse processo são todas lineares, não produzindo portanto, redução de velocidade.

Nestes códigos a propriedade de faixa mínima está associada à existência de um zero espectral em $\omega = \pi/T$ de $\mathcal{F}(\omega)$ o que torna $H(\omega)$ contínua, mesmo $G(\omega)$ sendo descontínuo nesta frequência. A existência desse zero faz com que a característica abrupta do filtro na frequência de Nyquist resulte num espectro contínuo de $H(\omega)$, cuja resposta

impulsiva $h(t)$ decai assintoticamente no mínimo, com $1/t^2$. Este decaimento rápido reduz a energia nas caudas do pulso, limitando a interferência entre símbolos (*IES*) no sistema a valores finitos que tendem a zero nos instantes de amostragem. Portanto, nessa condição, qualquer que seja a combinação de amostras devidamente espaçadas convergirá, garantindo um diagrama de olho aberto.

Em momento algum na abordagem de sistemas de resposta parcial foi citada a necessidade destes sistemas para se obter a propriedade de faixa mínima. A estratégia linear usada nos sistemas de resposta parcial é apenas suficiente, mas não necessária para produzir tal propriedade.

Para códigos de bloco não-lineares esse critério de faixa mínima não é aplicável. Entretanto, obteve-se recentemente um critério mais geral para códigos de bloco binários no trabalho desenvolvido por *Alcântara & Waldman*[3]. Neste trabalho verificar-se-á que o mesmo é possível para códigos de bloco ternários. A vantagem adicional que estes apresentam em relação aos de resposta parcial e ao binário proposto em [3] de faixa mínima é a redução de taxa de símbolos, pois nos códigos do tipo $mBnT$ cada bloco de m dígitos binários é codificado num bloco de n dígitos ternários.

Os códigos ternários têm se mostrado bastante viáveis como códigos de bloco não-lineares, comumente utilizados em transmissão digital. Estes são largamente utilizados em meios metálicos e menos freqüentes em fibras ópticas, onde são mais comuns os códigos binários. A utilização de códigos ternários é ilustrada em [4], onde apresenta-se um sistema de transmissão óptica. Porém, os códigos utilizados atualmente não são de faixa mínima.

No Capítulo 2 é estabelecida a condição de balanceamento para códigos ternários. O estudo feito é semelhante ao desenvolvido para o caso binário realizado em [5]. Um produto bastante interessante deste capítulo é o mapa de códigos eficientes, que permite a visualização destes em relação ao desbalanceamento permitido.

No Capítulo 3 discute-se e estabelece-se a condição generalizada de faixa mínima para o caso ternário[6,7], obtendo-se um mapa de códigos eficientes semelhante ao obtido no capítulo anterior.

Baseado nos capítulos anteriores uma condição simultânea de balanceamento e faixa mínima é estabelecida no Capítulo 4. Alguns códigos que possuem estas propriedades são investigados, observando ainda se estes proporcionam redução de velocidade.

E, finalmente, no Capítulo 5 é apresentado um programa computacionalmente rápido que permite o cálculo do diagrama de olho, baseado no *Algoritmo de Viterbi*[8]. Assim, pode-se comprovar que um código de faixa mínima produz um diagrama de olho aberto, mesmo quando transmitido com excesso de faixa nulo; pode-se ainda comparar os diagramas de olho dos códigos propostos com os já utilizados na prática.

Capítulo 2

CÓDIGOS TERNÁRIOS BALANCEADOS

2.1 Introdução

Um sistema ternário em banda base está representado na Fig.2.1. onde o canal considerado é ideal. Este sistema consiste de um codificador C , um formatador de pulso $G(\omega)$, um detetor de limiar e um decodificador C^{-1} .

C codifica a seqüência binária de dados $\{x_n\}$ em seqüência de símbolos ternários $\{y_n\}$. A seqüência $\{y_n\}$ pode ser representada tanto no domínio do tempo quanto no domínio da freqüência, de acordo com as seguintes expressões:

$$y(t) = \sum_{n=-\infty}^{\infty} y_n \delta(t - nT) \quad (2.1)$$

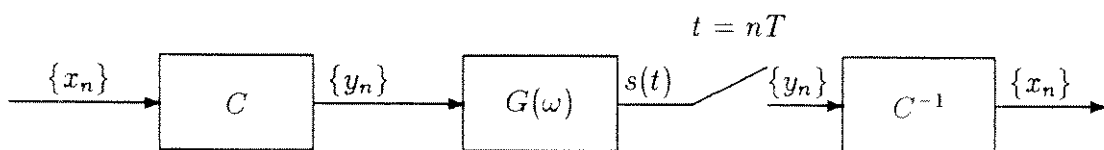


Figura 2.1: Diagrama em blocos de um sistema de transmissão de dados em banda base

$$Y(\omega) = \sum_{n=-\infty}^{\infty} y_n e^{-jn\omega T} \quad (2.2)$$

O formatador de pulso é um filtro que satisfaz o critério de *Nyquist*, dado por:

$$\sum_{k=-\infty}^{\infty} G(\omega - k\frac{2\pi}{T}) = T \quad (2.3)$$

$$g(t = nT) = \begin{cases} 1 & \text{para } n = 0 \\ 0 & \text{para } n \neq 0 \end{cases} \quad (2.4)$$

O sinal de linha na saída de $G(\omega)$ é dado por:

$$s(t) = \sum_{n=-\infty}^{\infty} y_n g(t - nT) \quad (2.5)$$

$$S(\omega) = Y(\omega)G(\omega) \quad (2.6)$$

Um dos parâmetros de sistema bastante conhecido na literatura [5] é a Soma Digital Corrida *SDC* definida por:

$$SDC = \sum_{n=1}^J y_n \quad (2.7)$$

De maneira alternativa, pode-se definir a *SDC* também como:

$$SDC = Y_{I,J}(\omega) \Big|_{\omega=0} \quad (2.8)$$

onde:

$$Y_{I,J}(\omega) = \sum_{n=1}^J y_n e^{-jn\omega T}$$

Outro parâmetro que também é interessante definir é a Variação da Soma Digital Corrida, $VSDC$ dada por:

$$VSDC = \max_{I,J} |Y_{I,J}(\omega)|_{\omega=0} \quad (2.9)$$

2.2 Condição de Balanceamento

Um código é dito balanceado se a $VSDC$ for finita, o que resulta no anulamento de $Y(\omega)$ na frequência $\omega = 0$.

Demonstração:

Seja $W_s(\omega)$ o espectro de potência de $s(t)$:

$$W_s(\omega) = Y(\omega) |G(\omega)|^2 \quad (2.10)$$

Como o pulso considerado é o de Nyquist, seu espectro de potência é um filtro passa-baixas ideal e que apenas limita a faixa do sinal, não afetando a sua forma. Assim, considera-se apenas $Y(\omega)$ na análise do espectro.

A densidade espectral pode ser obtida a partir da transformada de Fourier pela seguinte expressão[9] :

$$W_y(\omega) = \lim_{N \rightarrow \infty} \frac{1}{2NT} |Y_N(\omega)|^2 \quad (2.11)$$

Para $\omega = 0$, tem-se:

$$W_y(0) = \lim_{N \rightarrow \infty} \frac{1}{2NT} \left| \sum_{n=-N}^N y_n \right|^2 \quad (2.12)$$

Seja $VSDC = M$, onde M é um número finito, então:

$$W_y(0) \leq \lim_{N \rightarrow \infty} \frac{1}{2NT} (M)^2 = 0$$

portanto:

$$W_v(0) = 0$$

c.q.d.

2.3 Exemplos

Existem na literatura alguns códigos de bloco ternário balanceados bastante conhecidos, tais como: *AMI*, *4B3T*, *MS43* etc. O código *HDB - 3* é balanceado e limita o número de zeros consecutivos em 3. A seguir será apresentada uma breve revisão dos códigos de bloco balanceados:

2.3.1 Código *AMI*

Sua lei de formação é bastante simples: um espaço é representado por um intervalo de tempo sem pulso e uma marca por um pulso positivo ou negativo, de forma a obter uma alternância entre pulsos consecutivos independente do número de espaços entre eles.

vantagens:

- o espectro da densidade de potência possui componente dc nula e componentes em baixa frequência bastante pequenas; a *VSDC* é igual a 1.
- a informação de temporização pode ser facilmente recuperada de a partir do sinal recebido através de um retificador de onda completa seguido de um filtro passa-faixa estreito, ou de um *PLL (Phase Locked Loop)*
- simplicidade dos circuitos de codificação e decodificação.
- a monitoração de erro pode ser feita simplesmente observando-se as violações do código.

desvantagens:

- permite a ocorrência de longas seqüências de zeros, o que pode dificultar a recuperação de relógio.

2.3.2 Classe *4B3T*

No código *4B3T*, blocos de quatro dígitos binários são mapeados em palavras de três dígitos ternários. Tem-se no total $3^3 = 27$ palavras ternárias possíveis, sendo que apenas

palavra binária	palavra ternária	
	$D \geq 0$	$D \leq 0$
0000	0 - +	0 - -
0001	- + 0	- + 0
0010	- 0 +	- 0 +
0011	+ - +	- + -
0100	0 + +	0 - -
0101	0 + 0	0 - 0
0110	0 0 +	0 0 -
0111	- + +	+ - -
1000	0 + -	0 + -
1001	+ - 0	+ - 0
1010	+ 0 -	+ 0 -
1011	+ 0 0	- 0 0
1100	+ 0 +	- 0 -
1101	+ + 0	- - 0
1110	+ + -	- - +
1111	+ + +	- - -

Tabela 2.1: Tabela de codificação do 4B3T

7 destas palavras têm disparidade zero, que são:

$$0000, 0-- , 0-+ , +0- , +-0 , -0+ , -+0$$

Como $7 < 2^4$, é impossível se construir um código 4B3T balanceado e monoalfabético. Verificar-se-á, então, se é possível construir um código 4B3T balanceado com dois alfabetos, progressivo¹ e regressivo², investigando o número de palavras com disparidade³ positiva ou nula ($D \geq 0$):

Como o número total de palavras é 27, sendo que 7 dentre elas têm disparidade nula, restando portanto, 20 palavras. Por simetria, conclue-se que existem 10 palavras com disparidade positiva e 10 palavras com disparidade negativa. Portanto, o número total de palavras com disparidade positiva ou nula é 17. Isto significa que é possível mapear as 16 palavras binárias e ainda eliminar uma palavra do alfabeto, como por exemplo, 000.

No exemplo da Tabela 2.1, a SDC pode ocupar 8 estados: 0 a 7 ($VSDC = 7$) e o número de estados terminais é 6. Foram utilizados dois alfabetos: o progressivo e o regressivo.

2.3.3 MS - 43

Este código tem uma alocação mais sofisticada das palavras ternárias, resultando num melhor balanceamento e sincronismo.

¹alfabeto formado por palavras cuja disparidade é positiva ou nula

²alfabeto formado por palavras cuja disparidade é negativa ou nula

³A disparidade D de uma palavra é dada por: $D = \sum_{i=1}^n a_i$, onde n é o comprimento da palavra

palavra binária	palavra ternária		
	$D > 0$	$D \geq 0$ e $D < 0$	$D \leq 0$
0000	+ + +	- + -	- + -
0001	++ 0	0 0 -	0 0 -
0010	+ 0 +	0 - 0	0 - 0
0011	0 - +	0 - +	0 - +
0100	0 ++	- 0 0	- 0 0
0101	- 0 +	- 0 +	- 0 +
0110	- + 0	- + 0	- + 0
0111	- + +	- + +	- + +
1000	+ - +	+ - +	+ - +
1001	0 0 +	0 0 +	0 0 +
1010	0 + 0	0 + 0	0 + 0
1011	0 +-	0 +-	0 +-
1100	+ 0 0	+ 0 0	+ 0 0
1101	+ 0 -	+ 0 -	+ 0 -
1110	-- 0	+ - 0	+ - 0
1111	+ - -	+ - -	+ - -

Tabela 2.2: Tabela de Codificação MS43

A SDC pode ocupar 6 estados, de 0 a 5, fazendo com que a $VSDC$ seja igual a 5 e o número de estados terminais é igual a 4.

O $MS - 43$ utiliza três alfabetos: o positivo, o negativo e o misto, que é constituído de palavras com disparidade positiva, negativa e zero.

2.4 Máximo Desbalanço P e Máxima Redução de Velocidade

O intervalo dinâmico P da Soma Digital Corrida é dado por:

$$P = S + 2D \quad (2.13)$$

onde S é o intervalo dinâmico da Soma Digital Corrida Terminal, D é a extensão deste intervalo para cima e para baixo, devido ao efeito dos dígitos intermediários (não-terminais) e, finalmente P é o máximo desbalanço.

Seja $R(n, S, D)$ o número de palavras de comprimento n para cada par (S, D) que determinam um certo valor de P . Para um determinado P tem-se uma combinação de valores de (S, D) . Por exemplo, para $P = 5$, tem-se as seguintes possibilidades:

$$P = 5 \begin{cases} S = 1 & D = 2 \\ S = 3 & D = 1 \\ S = 5 & D = 0 \end{cases}$$

			1	
		1	3	
	1	2	6	
1	1	3	7	
	1	2	6	
		1	3	
			1	

Figura 2.2: Triângulo semelhante ao de Pascal

n	0	1	2	3
d				
3				1
2			1	3
1		1	2	6
0	1	1	3	7
-1		1	2	6
-2			1	3
-3				1

Tabela 2.3: Exemplo para a determinação de $R(n)$

Considerando-se que o cálculo de $R(n, S, D)$ por técnicas combinatórias é bastante complicado, desenvolveu-se um método simples baseado no triângulo semelhante ao de Pascal, mostrado na Fig. 2.2.

De acordo com a Fig.2.2, cada coluna é inicializada e finalizada pela unidade e os elementos intermediários são obtidos somando-se três elementos adjacentes. Tomou-se como exemplo o número 7 da figura, onde $7 = 2 + 3 + 2$. Este valor corresponde ao número de caminhos que chegam à posição do número em questão, partindo do vértice do triângulo e passando apenas uma vez em cada coluna e deslocando-se sempre em linhas adjacentes.

Baseado neste princípio obtém-se o número de palavras de tamanho n e disparidade d à partir de uma tabela, conforme exemplo da Tabela 2.3. Nesta tabela não se considerou a extensão do intervalo, nem os valores das disparidades permitidas.

Se as disparidades intermediárias das palavras forem limitadas, obtém-se $R(n, S, D)$.

As tabelas apresentadas na seção seguinte foram obtidas para valores de desbalanço P , tal que $2 \leq P \leq 7$, variando-se ainda as disparidades terminais permitidas, as quais estão indicadas por um (*).

Para determinar o número de palavras com comprimento n disponíveis para a formação do alfabeto, basta considerar apenas os números com as disparidades terminais permitidas. A partir desse número de palavras, obtém-se o máximo valor de m , que é o comprimento da palavra binária, de acordo com a seguinte expressão:

$$m_{m\acute{a}x} = \text{int}[\log_2 R(n)] \quad (2.14)$$

Uma vez determinado $m_{m\acute{a}x}$ para cada valor de n , determina-se a máxima redução de velocidade possível para um dado desbalanço P . O valor da máxima redução de velocidade, em porcentagem, é dado por:

$$\Delta(\%) = -\frac{n - m_{m\acute{a}x}}{m_{m\acute{a}x}} 100 \quad (2.15)$$

2.5 Tabelas

Nesta seção são apresentadas as tabelas para $2 \leq P \leq 7$ com todos os respectivos pares (S, D) possíveis.

n d	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	5	12	29	70	169	408	985	2378	5741	13860
* 0	1	3	7	17	41	99	239	577	1393	3363	8119	19601
-1	1	2	5	12	29	70	169	408	985	2378	5741	13860
$R(n)$	1	3	7	17	41	99	239	577	1393	3363	8119	19601
$m_{máx}$	0	1	2	4	5	6	7	9	10	11	12	14
$\Delta\%$								11,11	10	9,09	8,33	14,28

Tabela 2.4: $P = 2$ com $S = 0$ e $D = 1$

n d	1	2	3	4	5	6	7	8	9	10	11	12
* 1	1	2	4	9	21	50	120	289	697	1682	4060	9801
* 0	1	2	5	12	29	70	169	408	985	2378	5741	13860
* -1		1	3	8	20	49	119	288	696	1681	4059	9800
$R(n)$	2	5	12	29	70	169	408	985	2378	5741	13860	33461
$m_{máx}$	1	2	3	4	6	7	8	9	11	12	13	15
$\Delta\%$					16,67	14,28	12,50	11,11	18,18	16,67	15,38	20,00

Tabela 2.5: $P = 2$ com $S = 2$ e $D = 0$

n d	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	5	12	30	76	195	504	1309	3410	8900	23256
* 0	1	3	7	18	46	119	309	805	2101	5490	14356	37557
* -1	1	2	6	16	43	114	301	792	2080	5456	14301	34468
-2		1	3	9	25	68	182	483	1275	3355	8811	23112
$R(n)$	2	5	13	34	89	233	610	1597	4181	10946	28657	75025
$m_{máx}$	1	2	3	5	6	7	9	10	12	13	14	16
$\Delta\%$				20,00	16,67	14,28	22,22	20,00	25,00	23,08	21,43	25,00

Tabela 2.6: $P = 3$ com $S = 1$ e $D = 1$

n d	1	2	3	4	5	6	7	8	9	10	11	12
* 1	1	2	4	9	21	51	127	322	826	2135	5545	14445
* 0	1	2	5	12	30	76	195	504	1309	3410	8900	23256
* -1		1	3	9	25	68	182	483	1275	3355	8811	23112
* -2			1	4	13	38	106	288	771	2046	5401	14212
$R(n)$	2	5	13	34	89	233	610	1597	4181	10946	28657	75025
$m_{máx}$	1	2	3	5	6	7	9	10	12	13	14	16
$\Delta\%$				20,00	16,67	14,28	22,22	20,00	25,00	23,08	21,43	25,00

Tabela 2.7: $P = 3$ com $S = 3$ e $D = 0$

n d	1	2	3	4	5	6	7	8	9	10	11	12
2		1	3	9	25	69	189	517	1413	3861	10549	28821
1	1	2	6	16	44	120	328	896	2448	6688	18272	49920
* 0	1	3	7	19	51	139	379	1035	2827	7723	21099	57643
-1	1	2	6	16	44	120	328	896	2448	6688	18272	49920
-2		1	3	9	25	69	189	517	1413	3861	10549	28821
$R(n)$	1	3	7	19	51	139	379	1035	2827	7723	21099	57643
$m_{máx}$	0	1	2	4	5	7	8	10	11	12	14	15
$\Delta\%$						14,28	12,50	20,00	18,18	16,67	21,43	20,00

Tabela 2.8: $P = 4$ com $S = 0$ e $D = 2$

n d	1	2	3	4	5	6	7	8	9	10	11	12
2	1	2	5	12	30	76	196	512	1352	3600	9648	25984
* 1	1	3	7	18	46	120	316	840	2448	6048	16336	44256
* 0	1	2	6	16	44	120	328	896	2448	6688	18272	49920
* -1		1	3	10	30	88	252	712	1992	5536	15312	42208
-2			1	4	14	44	132	384	1096	3088	8624	23936
$R(n)$	2	6	16	44	120	328	896	2448	6688	18272	49920	136384
$m_{máx}$	1	2	4	5	6	8	9	11	12	14	15	17
$\Delta\%$			25,00	20,00	16,67	25,00	22,22	27,27	25,00	28,57	26,67	29,41

Tabela 2.9: $P = 4$ com $S = 2$ e $D = 1$

n d	1	2	3	4	5	6	7	8	9	10	11	12
* 2	1	2	4	9	21	51	127	323	835	2187	5787	15435
* 1	1	2	5	12	30	76	196	512	1352	3600	9648	25984
* 0		1	3	9	25	69	189	517	1413	3861	10549	28821
* -1			1	4	14	44	132	384	1096	3088	8624	23936
* -2				1	5	19	63	195	579	1675	4763	13387
$R(n)$	2	5	13	35	95	259	707	1931	5275	14411	39371	107563
$m_{máx}$	1	2	3	5	6	8	9	10	12	13	15	16
$\Delta\%$				20,00	16,67	25,00	22,22	20,00	25,00	23,08	26,67	25,00

Tabela 2.10: $P = 4$ com $S = 4$ e $D = 0$

n d	1	2	3	4	5	6	7	8	9	10	11	12
2		1	3	9	25	69	189	518	1442	3914	10802	29887
* 1	1	2	6	16	44	120	329	904	2492	6888	19085	52988
* 0	1	3	7	19	51	140	386	1070	2974	8283	23101	64494
* -1	1	2	6	16	45	126	355	1000	2817	7930	22308	62712
-2		1	3	10	30	89	259	747	2139	6095	17303	48984
-3			1	4	14	44	133	392	1139	3278	9373	26676
$R(n)$	2	5	13	35	96	266	741	2070	5791	16213	45409	127206
$m_{m\acute{a}x}$	1	2	3	5	6	8	9	11	12	13	15	16
$\Delta\%$				20,00	16,67	25,00	22,22	27,27	25,00	23,08	26,67	25,00

Tabela 2.11: $P = 5$ com $S = 1$ e $D = 2$

n d	1	2	3	4	5	6	7	8	9	10	11	12
2	1	2	5	12	30	76	196	512	1352	3605	9692	26242
* 1	1	3	7	18	46	120	316	840	2253	6087	16550	45238
* 0	1	2	6	16	44	120	328	901	2482	6858	18996	52729
* -1		1	3	10	30	88	257	741	2123	6051	17183	48655
* -2			1	4	14	49	156	481	1446	4274	12476	36084
-3				1	5	19	68	224	705	2151	6425	18901
$R(n)$	2	6	17	48	134	377	1057	2963	8304	23270	65205	182706
$m_{m\acute{a}x}$	1	2	4	5	7	8	10	11	13	14	15	17
$\Delta\%$			25,00	20,00	28,57	25,00	30,00	27,27	30,77	28,57	26,67	29,41

Tabela 2.12: $P = 5$ com $S = 3$ e $D = 1$

n d	1	2	3	4	5	6	7	8	9	10	11	12
* 2	1	2	4	9	21	51	127	323	835	2187	5793	15490
* 1	1	2	5	12	30	76	196	512	1352	3606	9697	26262
* 0		1	3	9	25	69	189	517	1419	3904	10772	29798
* -1			1	4	14	44	132	390	1133	3262	9329	26556
* -2				1	5	19	69	226	710	2163	6455	18977
* -3					1	6	25	94	320	1030	3193	9648
$R(n)$	2	5	13	35	95	265	738	2062	5769	16152	45239	126731
$m_{m\acute{a}x}$	1	2	3	5	6	8	9	11	12	13	15	16
$\Delta\%$				20,00	16,67	25,00	22,22	27,27	25,00	23,08	26,67	25,00

Tabela 2.13: $P = 5$ com $S = 5$ e $D = 0$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
3			1	4	14	44	133	392	1140	3288	9437	27004
2		1	3	10	30	89	259	748	2148	6149	17567	50126
1	1	2	6	16	45	126	356	1008	2861	8130	23122	65792
* 0	1	3	7	19	51	141	393	1105	3121	8843	25103	71347
-1	1	2	6	16	45	126	356	1008	2861	8130	23122	65792
-2		1	3	10	30	89	259	748	2148	6149	17567	50126
-3			1	4	14	44	133	392	1140	3288	9437	27004
$R(n)$	1	3	7	19	51	141	393	1105	3121	8843	25103	71347
$m_{máx}$	0	1	2	4	5	7	8	10	11	13	14	16
$\Delta\%$						14,28	12,50	20,00	18,18	23,08	21,43	25,00

Tabela 2.14: $P = 6$ com $S = 0$ e $D = 3$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
3		1	3	9	25	69	189	518	1422	3914	10807	29936
2	1	2	6	16	44	120	329	904	2493	6893	19129	53251
* 1	1	3	7	19	51	140	386	1070	2979	8322	23315	65477
* 0	1	2	6	16	45	126	355	1005	2851	8100	23033	65533
* -1		1	3	10	30	89	264	776	2270	6611	19185	55507
-2			1	4	14	49	157	489	1490	4474	13289	39152
-3				1	5	19	68	225	714	2204	6678	19967
$R(n)$	2	6	16	45	126	355	1005	2851	8100	23033	65533	186517
$m_{máx}$	1	2	4	5	6	8	9	11	12	14	15	17
$\Delta\%$			25,00	20,00	16,67	25,00	22,22	27,27	25,00	28,57	26,67	29,41

Tabela 2.15: $P = 6$ com $S = 2$ e $D = 2$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
3	1	2	5	12	30	76	196	512	1352	3605	9693	26254
* 2	1	3	7	18	46	120	316	840	2253	6088	16561	45314
* 1	1	2	6	16	44	120	328	901	2483	6868	19060	53062
* 0		1	3	10	30	88	257	742	2132	6104	17441	49770
* -1			1	4	14	49	157	489	1489	4469	13269	39082
* -2				1	5	20	75	258	848	2696	8372	25544
-3					1	6	26	101	359	1207	3903	12275
$R(n)$	2	6	17	49	139	397	1133	3230	9205	26225	74703	212772
$m_{máx}$	1	2	4	5	7	8	10	11	13	14	16	17
$\Delta\%$			25,00	20,00	28,57	25,00	30,00	27,27	30,77	28,57	31,25	29,41

Tabela 2.16: $P = 6$ com $S = 4$ e $D = 1$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
* 3	1	2	4	9	21	51	127	323	835	2187	5793	15491
* 2	1	2	5	12	30	76	196	512	1352	3606	9698	26274
* 1		1	3	9	25	69	189	517	1419	3905	10783	29873
* 0			1	4	14	44	133	392	1139	3284	9422	26954
* -1				1	5	20	70	230	726	2233	6749	20142
* -2					1	6	27	104	368	1232	3971	12458
* -3						1	7	34	138	506	1738	5709
$R(n)$	2	5	13	35	95	267	749	2082	5977	16953	48154	136901
$m_m \acute{a}x$	1	2	3	5	6	8	9	11	12	14	15	17
$\Delta\%$				20,00	16,67	25,00	22,22	27,27	25,00	28,57	26,67	29,41

Tabela 2.17: $P = 6$ com $S = 6$ e $D = 0$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
3			1	4	14	44	133	392	1140	3288	9438	27016
2		1	3	10	30	89	259	748	2148	6150	17578	50203
* 1	1	2	6	16	45	126	356	1008	2862	8140	23187	66132
* 0	1	3	7	19	51	141	393	1106	3130	8897	25367	72490
-1	1	2	6	16	45	126	357	1016	2905	8330	23936	68872
-2		1	3	10	30	90	266	783	2295	6709	19569	56979
-3			1	4	15	50	160	496	1509	4530	13474	39808
-4				1	5	20	70	230	726	2235	6765	20239
$R(n)$	2	5	13	35	96	267	750	2122	6035	17227	49303	141362
$m_m \acute{a}x$	1	2	4	5	7	8	10	11	13	14	16	17
$\Delta\%$				20,00	16,67	25,00	22,22	27,27	25,00	28,57	26,67	29,41

Tabela 2.18: $P = 7$ com $S = 1$ e $D = 3$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
3		1	3	9	25	69	189	518	1422	3915	10813	29964
2	1	2	6	16	44	120	329	904	2493	6898	19151	53340
* 1	1	3	7	19	51	140	386	1071	2983	8338	23376	65714
* 0	1	2	6	16	45	126	355	1008	2862	8140	23187	66132
* -1		1	3	10	30	90	266	783	2295	6709	19569	56979
* -2			1	4	15	50	161	504	1552	4720	14223	42548
-3				1	5	21	77	265	873	2794	8756	27015
-4					1	6	27	104	369	1242	4036	12792
$R(n)$	2	6	16	49	141	406	1169	3366	9692	27907	80355	231373
$m_m \acute{a}x$	1	2	4	5	7	8	10	11	13	14	16	17
$\Delta\%$			25,00	20,00	28,57	25,00	30,00	27,27	30,78	28,57	31,25	29,41

Tabela 2.19: $P = 7$ com $S = 3$ e $D = 2$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
3	1	2	5	12	30	76	196	512	1353	3610	9713	26324
* 2	1	3	7	18	46	120	316	841	2257	6103	16611	45475
* 1	1	2	6	16	44	120	329	904	2493	6898	19151	53340
* 0		1	3	10	30	89	259	748	2148	6150	17578	50203
* - 1			1	4	15	50	160	496	1509	4530	13474	39808
* - 2				1	5	21	77	265	873	2794	8756	27015
* - 3					1	6	28	112	412	1432	4785	15532
- 4						1	7	35	147	559	1991	6776
$R(n)$	2	6	17	49	141	406	1149	3366	9692	27907	80355	231373
$m_{máx}$	1	2	4	5	7	8	10	11	13	14	16	17
$\Delta\%$			25,00	20,00	28,57	25,00	30,00	27,27	30,77	28,57	31,25	29,41

Tabela 2.20: $P = 7$ com $S = 5$ e $D = 1$

n	1	2	3	4	5	6	7	8	9	10	11	12
d												
* 3	1	2	4	9	19	49	123	315	817	3550	2146	15257
* 2	1	2	5	12	30	74	192	502	1329	3550	9561	25932
* 1		1	3	9	25	69	187	512	1404	3865	10675	29586
* 0			1	4	14	44	133	390	1132	3260	9350	26750
* - 1				1	5	20	70	230	724	2225	6725	20099
* - 2					1	6	27	104	369	1240	4024	12738
* - 3						1	7	35	147	559	1989	6762
* - 4							1	8	43	190	749	2738
$R(n)$	2	5	13	35	94	263	740	2096	5965	17035	48769	139862
$m_{máx}$	1	2	3	5	6	8	9	11	12	14	15	17
$\Delta\%$				20,00	16,67	25,00	22,22	27,27	25,00	28,57	26,67	29,41

Tabela 2.21: $P = 7$ com $S = 7$ e $D = 0$

2.6 Limites Fundamentais de Eficiência[5]

Nas tabelas apresentadas nota-se um comportamento oscilatório da redução de velocidade $\Delta\%$ à medida que n cresce. Por isso é interessante determinar o limite assintótico.

Seja \vec{v}_n o vetor formado pelos elementos da $n^{\text{ésima}}$ coluna da tabela, então:

$$\vec{v}_n = A^{n-1}\vec{v}_1 \quad (2.16)$$

onde A é a matriz de transição e \vec{v}_1 é o vetor formado pelos elementos da 1ª. coluna da tabela.

Para $P = 2$, a matriz A é dada por:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Como A é diagonalizável, existe uma matriz inversível M tal que: $M^{-1}AM = D$, sendo que D é a matriz diagonal dos autovalores de A , isto é:

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Mas como $M^{-1}AM = D$, então $A = MDM^{-1}$ e $A^p = MD^pM^{-1}$. Substituindo este resultado na Eq.(2.16),tem-se:

$$\vec{v}_n = MD^{n-1}M^{-1}\vec{v}_1 \quad (2.17)$$

Como $R(n)$ é combinação linear de elementos de \bar{v}_n , então:

$$R(n) = c_1 \lambda_1^{n-1} + c_2 \lambda_2^{n-1} + c_3 \lambda_3^{n-1}$$

onde c_1 , c_2 e c_3 são constantes.

Os autovalores são dados pela seguinte equação:

$$\det(A - \lambda I) = 0$$

$$\det \begin{vmatrix} 1 - \lambda & 1 & 0 \\ 1 & 1 - \lambda & 1 \\ 0 & 1 & 1 - \lambda \end{vmatrix} = 0$$

$$\Rightarrow (1 - \lambda)(\lambda^2 - 2\lambda) - (1 - \lambda) = 0$$

$$(1 - \lambda)(\lambda^2 - 2\lambda - 1) = 0$$

raízes: $\lambda_1 = 1$
 $\lambda_2 = 1 + \sqrt{2}$
 $\lambda_3 = 1 - \sqrt{2}$

como $R(1) = 2$, $R(2) = 5$ e $R(3) = 12$:

$$\begin{cases} c_1 + c_2 + c_3 = 2 \\ c_1 \lambda_1 + c_2 \lambda_2 + c_3 \lambda_3 = 5 \\ c_1 \lambda_1^2 + c_2 \lambda_2^2 + c_3 \lambda_3^2 = 12 \end{cases}$$

Resolvendo este sistema de equações, tem-se:

$$\begin{aligned} c_1 &= 1 \\ c_2 &= \frac{1}{2} + \frac{3}{2\sqrt{2}} \\ c_3 &= \frac{1}{2} - \frac{3}{2\sqrt{2}} \end{aligned}$$

$$\Rightarrow R(n) = 1 + \left(\frac{1}{2} - \frac{3}{2\sqrt{2}}\right) (1 + \sqrt{2})^{n-1} + \left(\frac{1}{2} + \frac{3}{2\sqrt{2}}\right) (1 - \sqrt{2})^{n-1}$$

Para valores de n grande, o segundo termo da equação acima, correspondente ao maior autovalor, predominará. Portanto, para cada valor de n , os códigos $m_{m\acute{a}x} BnT$ são dados por:

$$2^{m_{m\acute{a}x}} \leq R(n) < 2^{m_{m\acute{a}x}+1} \quad (2.18)$$

$$m_{m\acute{a}x} \leq \log_2 R(n) < m_{m\acute{a}x} + 1 \quad (2.19)$$

Logo, para n grande:

$$m_{m\acute{a}x} \leq \log_2 \left[\left(\frac{1}{2} + \frac{3}{2\sqrt{2}}\right) (1 + \sqrt{2})^{n-1} \right] < m_{m\acute{a}x} + 1 \quad (2.20)$$

Como $m_{m\acute{a}x}$ e n são grandes, os termos contantes podem ser desprezados.

$$m_{m\acute{a}x} \leq n \log_2(1 + \sqrt{2}) < m_{m\acute{a}x} + 1 \quad (2.21)$$

$$\lim_{n \rightarrow \infty} \frac{m_{m\acute{a}x}}{n} = \log_2(1 + \sqrt{2})$$

Deve-se notar que $(1 + \sqrt{2})$ é o maior autovalor, $\lambda_{m\acute{a}x}$, da matriz A . Assim, a máxima redução de velocidade é dada por:

$$\Delta_{m\acute{a}x}\% = \left[1 - \left(\frac{n}{m} \right)_{m\acute{a}x} \right] 100\% \quad (2.22)$$

$$\begin{aligned} \Delta_{m\acute{a}x}\% &= \left[1 - \frac{1}{\log_2(1+\sqrt{2})} \right] 100\% \\ &= 21.36\% \end{aligned}$$

Com este resultado, pode-se concluir que para um desbalanço $P = 2$, a máxima redução de velocidade que se pode conseguir teoricamente, aumentando-se a complexidade do código, é 21.36%. Com o código 15B12T, obtém-se uma redução de 20%, valor bastante próximo do assintótico.

Para o caso de P qualquer, pode se repetir o procedimento anterior para obter a eficiência assintótica para códigos do tipo $mBnT$. Assim, para um determinado P , deve-se:

1 - Determinar a matriz de transição A .

A sua forma genérica é:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 \end{bmatrix}$$

A ordem da matriz é dada por:

$$k = P + 1 \quad (2.23)$$

2 - Para cada matriz $A(P)$, determinar o maior autovalor $\lambda_{m\acute{a}x}(P)$.

3 - O limite assintótico de redução de velocidade é dado por:

$$\Delta_{m\acute{a}x}(P) = \left[1 - \frac{1}{\log_2 \lambda_{m\acute{a}x}^{(P)}} \right] 100\% \quad (2.24)$$

Analisando as tabelas anteriores, verifica-se que para $P = 2$, a máxima redução de velocidade que se consegue, para complexidade até $n = 12$, é 20%. Como foi visto no exemplo de cálculo do limite assintótico, o valor que se obtém com o código $15B12T$ aproxima-se bastante da eficiência assintótica que é 21,36%. Portanto, para este desbalanço, os códigos mais eficientes são: $11B9T$ e $15B12T$.

Para $P = 3$ observa-se que a classe de códigos que apresenta maior eficiência é $12B9T$ ($\Delta\% = 25\%$), mas a sua complexidade ainda é bastante significativa.

Aumentando-se o valor de desbalanço para $P = 4$, verifica-se que as classes de códigos mais eficientes são $14B10T$ e $17B12T$, que ainda são considerados complexos. Para uma complexidade menor, observa-se que os códigos $4B3T$ apresentam-se como opção com redução de velocidade de 25%. Porém, estes códigos não apresentam redundância, já que o número de palavras-código ternárias disponíveis coincide com o número de palavras binárias a serem mapeadas. Portanto, estes códigos podem apresentar problemas de sincronismo, já que a palavra 000 não pode ser excluída. Com $P = 5$, observa-se que este problema pode ser resolvido.

Para uma análise mais detalhada dos valores de eficiência, determinou-se para cada desbalanço P , a matriz de transição A , o autovalor máximo e, finalmente, o limite assintótico.

Para determinar os autovalores de A , basta fazer:

$$\det(A - \lambda I) = 0$$

então:

$$\begin{vmatrix} 1-\lambda & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1-\lambda & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1-\lambda & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1-\lambda & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1-\lambda \end{vmatrix} = 0$$

Se $Q_k(\lambda)$ for o polinômio característico de grau n correspondente à matriz de transição que permite um desbalanço $P = k$, pode-se obter o determinante da matriz A pela regra dos cofatores:

$$Q_k(\lambda) = (1 - \lambda)Q_{k-1}(\lambda) - Q_{k-2}(\lambda) \tag{2.25}$$

E os coeficientes de $Q_k(\lambda)$ são dados por:

$$a_n(k) = -a_{n-1}(k - 1) \tag{2.26}$$

$$a_{n-1}(k) = a_{n-1}(k - 1) - a_{n-2}(k - 1) \tag{2.27}$$

$$a_{n-2}(k) = a_{n-2}(k - 1) - a_{n-3}(k - 1) - a_{n-2}(k - 2) \tag{2.28}$$

\vdots

$$a_1(k) = a_1(k - 1) - a_0(k - 1) - a_1(k - 2) \tag{2.29}$$

Portanto, os coeficientes da equação característica para desbalanço $k \geq 3$ podem ser obtidos, desde que se tenha os coeficientes da equação com desbalanço $k = 0$ e $k = 1$. De acordo com a forma genérica da matriz A , vemos que:

$$Q_0(\lambda) = 1 - \lambda$$

e

$$Q_1(\lambda) = (1 - \lambda)^2 - 1 = \lambda^2 - 2\lambda$$

então:

$$\begin{aligned} a_1(0) &= -1 \\ a_0(0) &= 1 \\ a_2(1) &= 1 \\ a_1(1) &= -2 \\ a_0(1) &= 0 \end{aligned}$$

A partir dos dados acima e das Eqs.(2.26),(2.27) e (2.28) gerou-se a tabela

$P = k$	$Q_k(\lambda)$								
0							-1	1	
1							1	-2	0
2						-1	3	-1	-1
3					1	-4	3	2	-1
4				-1	5	-6	-2	4	0
5			1	-6	10	0	-9	2	1
6		-1	7	-15	5	15	-9	-3	1
7	1	-8	21	-14	-20	24	3	-6	0
	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

P	Equação Característica	$\lambda_{máz}$	$\Delta_{máz} (\%)$
2	$-\lambda^3 + 3\lambda^2 - \lambda - 1 = 0$	2,41	21,20
3	$\lambda^4 - 4\lambda^3 + 3\lambda^2 + 2\lambda - 1$	2,62	28,04
4	$-\lambda^5 + 5\lambda^4 - 6\lambda^3 - 2\lambda^2 + 4\lambda = 0$	2,73	30,98
5	$\lambda^6 - 6\lambda^5 + 10\lambda^4 - 9\lambda^2 + 2\lambda + 1 = 0$	2,80	32,68
6	$-\lambda^7 + 7\lambda^6 - 15\lambda^5 + 5\lambda^4 + 15\lambda^3 - 9\lambda^2 - 3\lambda + 1 = 0$	2,85	33,82
7	$\lambda^8 - 8\lambda^7 + 21\lambda^6 - 14\lambda^5 - 20\lambda^4 + 24\lambda^3 + 3\lambda^2 - 6\lambda = 0$	2,88	34,47

O limite assintótico de $\lambda_{máz}$ é 3 e, portanto, a redução de velocidade que se conseguiria é dada por:

$$1 - \frac{1}{\log_2 3} = 36,9\%$$

A área hachurada indicada na Fig 2.3 é realizável apenas com códigos de complexidade maior que 12, $n > 12$. As outras áreas correspondem às classes de códigos mais simples para um dado desbalanço P . Está indicada também a classe de códigos que não apresentam redução de velocidade, 1B1T.

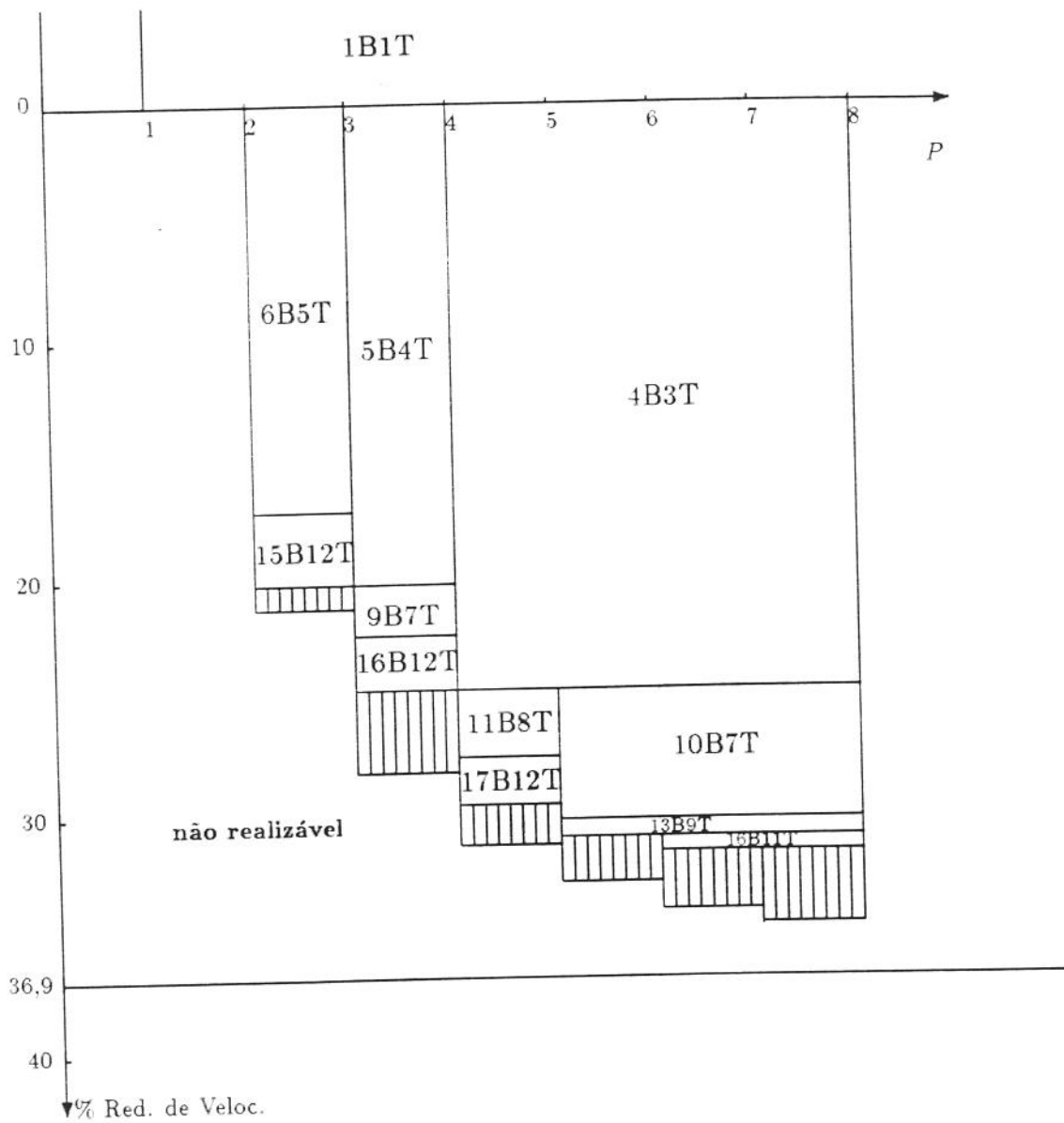


Figura 2.3: Mapa de Códigos Eficientes

Capítulo 3

CÓDIGOS TERNÁRIOS DE FAIXA MÍNIMA

3.1 Introdução

Sabe-se que existem alguns requisitos básicos quando se procura um código de linha apropriado. Dentre eles está a minimização da faixa espectral. Segundo *Nyquist*[1], para um pulso à taxa de transmissão $1/T$, onde T é o período, a faixa teórica mínima para transmiti-lo é $f_n = 1/(2T)$. Para que isto ocorra sem interferência entre símbolos *IES* nos instantes de amostragem, é necessário que o espectro de freqüências do canal seja igual ao do filtro passa-baixas ideal com freqüência de corte em ω_n , conforme a Fig 3.1.

Este filtro é também denominado filtro ideal de Nyquist, e sua resposta impulsiva $g(t)$, mostrada na Fig 3.2, é também conhecida como pulso de Nyquist.

Na prática este filtro é irrealizável e mesmo que não o fosse, o pulso de Nyquist exigiria extrema precisão no sincronismo. Se houver qualquer imprecisão, não se tem mais a condição de *IES* zero, pois a superposição das caudas dos pulsos adjacentes constitui-se

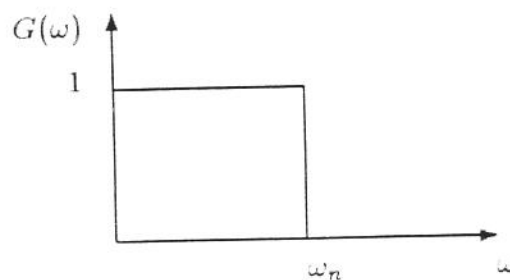


Figura 3.1: Filtro passa-baixas ideal

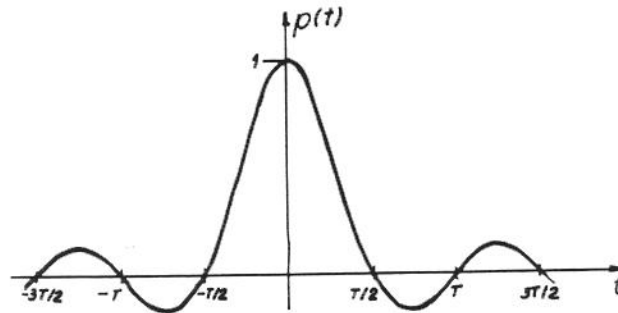


Figura 3.2: Pulso de Nyquist

divergente, podendo causar interpretação incorreta do pulso recebido, mesmo na ausência de ruído.

À medida que se suaviza o corte do filtro ideal, a natureza oscilatória da cauda do pulso é reduzida, pois acentua-se o seu decaimento assintótico. Esta suavização é obtida às custas de um excesso de faixa (roll-off), de valor fracionário β .

Um conjunto de filtros, que além de apresentar uma característica espectral mais suave, satisfaz o critério de Nyquist, são os filtros do tipo cosseno-levantado[11], cuja função de transferência é dada por:

$$H(\omega) = \begin{cases} T & |\omega| \leq \frac{\omega_1}{2(1-\beta)} \\ \frac{T}{2} \{1 - \text{sen}[\frac{T}{2\beta} (|\omega| - \frac{\omega_1}{2})]\} & \frac{\omega_1(1-\beta)}{2} \leq |\omega| \leq \frac{\omega_1(1+\beta)}{2} \\ 0 & \text{caso contrário} \end{cases} \quad (3.1)$$

onde: $\omega_1 = \frac{2\pi}{T}$
 $T =$ período de transmissão de dados
 $\beta =$ excesso de faixa

A sua resposta impulsiva é dada por:

$$h(t) = \frac{\text{sen} \frac{\pi t}{T} \cos \frac{\beta \pi t}{T}}{\frac{\pi t}{T} [1 - \frac{4\beta^2 t^2}{T^2}]} \quad (3.2)$$

Note que para excesso de faixa nulo, $\beta = 0$, $H(\omega)$ é o próprio filtro de Nyquist.

Para excessos de faixa diferentes de zero, a superposição das caudas dos pulsos adjacentes constitui-se numa série convergente, pois $h(t)$ decai assintoticamente segundo $1/t^3$.

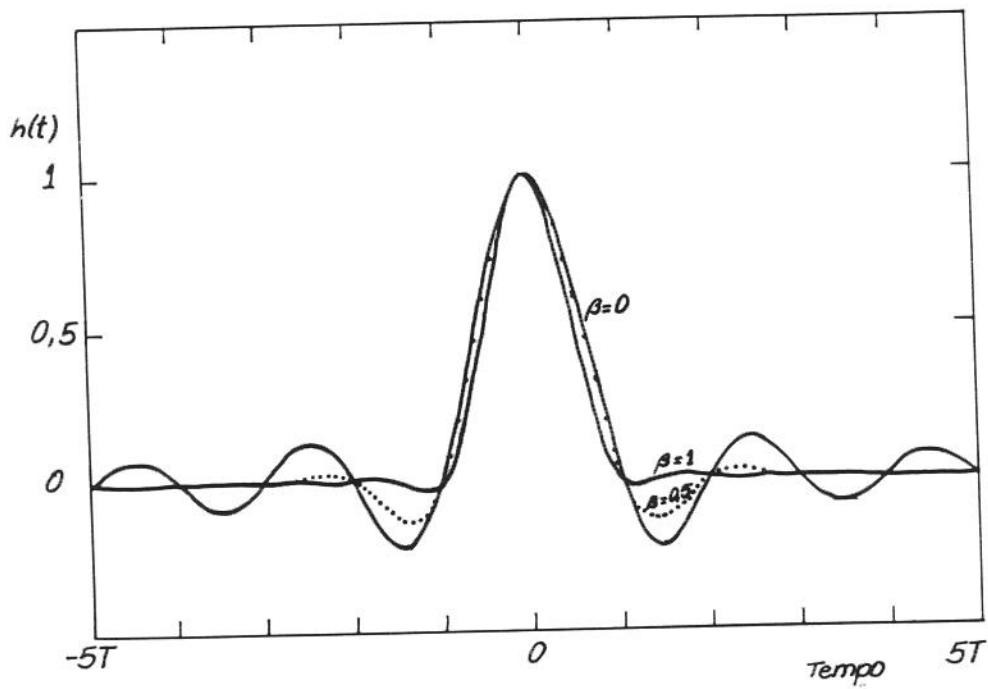
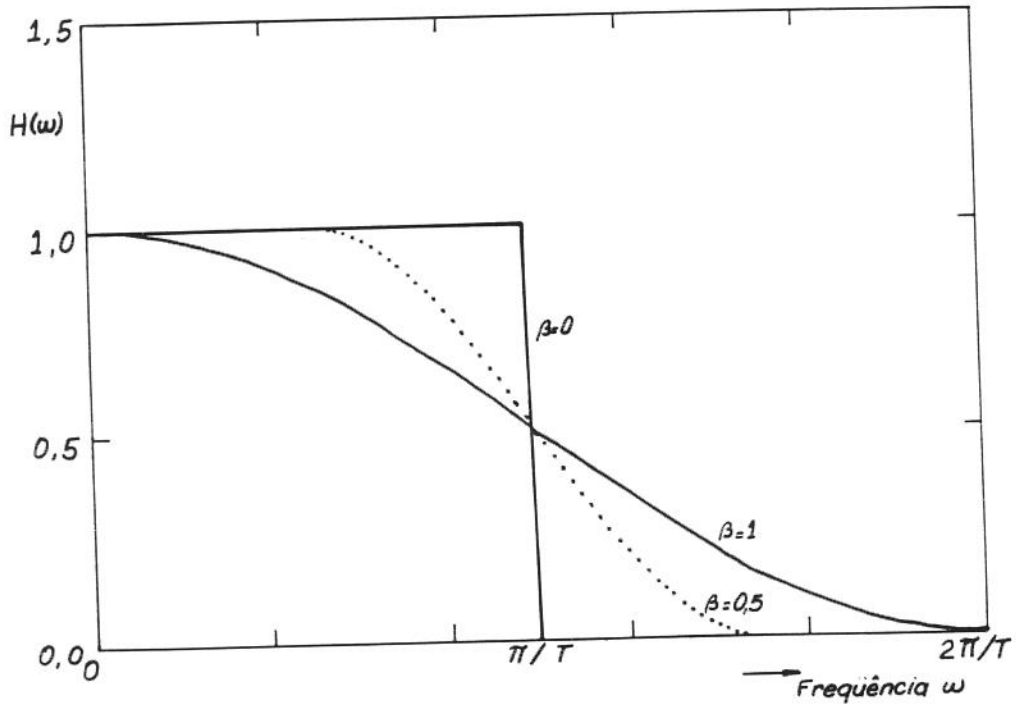


Figura 3.3: Espectro de amplitude do filtro cosseno-levantado para os diversos valores de β

Figura 3.4: Resposta impulsiva associada a cada espectro

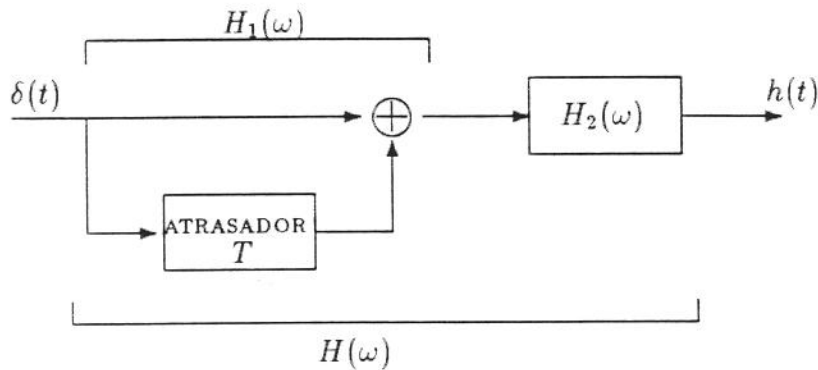


Figura 3.5: Diagrama em blocos para gerar o sinal DB

conforme a Eq.(3.2). Por esta razão o sincronismo não é tão crítico quando se opera com excesso de faixa diferente de zero. Ou seja, mesmo havendo um pequeno desvio no instante de amostragem pode-se garantir decisão correta do pulso recebido. Quanto maior for o excesso de faixa β , mais imune será o sistema às variações das posições dos pulsos de relógio, pois maior será a abertura do diagrama de olho.

Até agora falou-se apenas em formas de onda que produzem IES igual a zero. Dispõe-se de outra alternativa para otimizar a utilização da faixa, introduzida por *Lender*[12], que é a técnica duobinária. Esta técnica consiste em combinar dois pulsos sucessivos para dar origem a um sinal multinível. Este processo introduz IES , mas de maneira controlada, permitindo a sua remoção quando o pulso é recebido. O caso geral em que se combina linearmente n pulsos binários sucessivos, de maneira conhecida e controlada, é conhecido como sistemas de resposta parcial.

O exemplo mais simples de sistemas de resposta parcial é o duobinário, (DB), onde $n = 2$, cuja estrutura para a geração do sinal é mostrada na Fig 3.5.

onde:

$$H_1(\omega) = 1 + e^{-j\omega T} = 2\cos\frac{\omega T}{2}e^{-j\omega T/2} \quad (3.3)$$

$$H_2(\omega) = \begin{cases} T, & -\frac{\pi}{T} \leq \omega \leq \frac{\pi}{T} \\ 0, & \omega > \frac{\pi}{T} \end{cases} \quad (3.4)$$

como $H(\omega) = H_1(\omega)H_2(\omega)$

então:

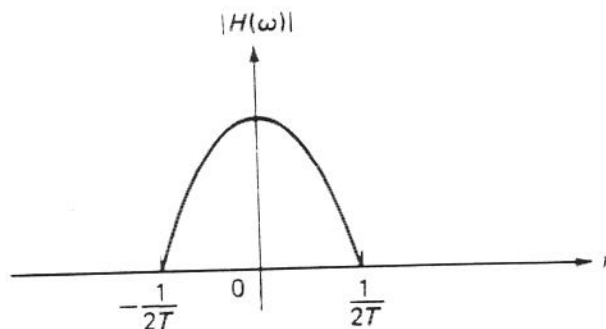


Figura 3.6: Característica espectral do DBM

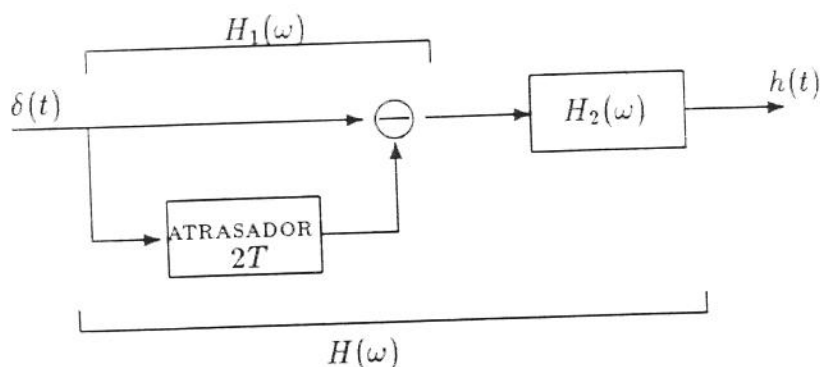


Figura 3.7: Diagrama em blocos para implementação do DBM

$$H(\omega) = \begin{cases} 2T \cos \frac{\omega T}{2} e^{-j\omega T/2} & -\pi/T \leq \omega \leq \pi/T \\ 0 & |\omega| > \pi/T \end{cases} \quad (3.5)$$

O espectro de amplitude $H(\omega)$ possui uma característica cossenoidal que pode ser aproximada na prática e pode ser utilizada para gerar uma seqüência duobinária a partir de uma binária. Observa-se também que a faixa de transmissão é a mínima, ou seja, sem excesso de faixa, com $\beta = 0$.

Outro exemplo de sistemas de resposta parcial que é também bastante conhecido é o duobinário modificado (DBM), cujo esquema básico de implementação é apresentado na Fig 3.6.

onde:

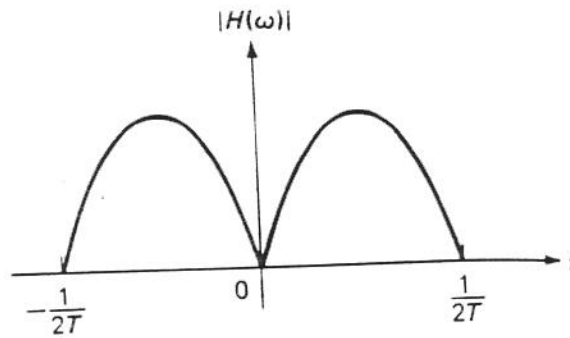


Figura 3.8: Característica espectral do DBM

$$\begin{aligned} H_1(\omega) &= 1 - e^{-2j\omega T} \\ &= 2je^{-j\omega T} \text{sen}\omega T \end{aligned}$$

Logo:

$$H_1(\omega) = 2 \text{sen}\omega T$$

Novamente se $H_2(\omega)$ for o filtro passa-baixas ideal com amplitude T , o espectro de amplitude resultante será:

$$H(\omega) = \begin{cases} 2T \text{sen}\omega T & -\pi/T \leq \omega \leq \pi/T \\ 0 & \omega > \pi/T \end{cases}$$

As duas técnicas apresentadas, que são um caso particular dos sistemas de resposta parcial, possibilitam a transmissão de sinais à velocidade de $1/T$ com uma faixa de $1/(2T)$, que é mínima.

O conceito de códigos de faixa mínima, introduzido por Kim [6], consiste na possibilidade de se operar com excesso de faixa nulo e, ainda assim, manter o diagrama de olho aberto. É uma generalização do zero espectral na frequência de Nyquist. Portanto, os sistemas de resposta parcial são um exemplo típico de códigos de faixa mínima. Kim propõe

em seu trabalho alguns códigos que são variações do Bipolar, do Duobinário e do Duobinário Modificado.

Será discutido, em seguida, um critério geral para a propriedade de faixa mínima aplicável aos códigos de bloco utilizados em transmissão digital.

3.2 Condição Generalizada para Faixa Mínima

Seja o sinal ternário representado pela Eq.(2.2). De maneira análoga à Soma Digital Corrida (*SDC*), será definido um novo parâmetro, a Soma Digital de Interferência entre Símbolos (*SDI*):

$$SDI \triangleq \sum_{n=I}^J (-1)^n y_n \quad (3.6)$$

onde I e J são números inteiros arbitrários

E da mesma forma define-se também a Variação da Soma Digital de Interferência entre símbolos, $VSDI$:

$$VSDI \triangleq \max_{I,J} |SDI| \quad (3.7)$$

Note que comparando as Eqs. (2.2),(3.1) e (3.2), observa-se que a *SDI* e a *VSDI* podem ser expressas da seguinte forma:

$$SDI \triangleq Y_{I,J}(\omega) \Big|_{\omega=\pi/T}$$

$$VSDI \triangleq \max Y_{I,J}(\omega) \Big|_{\omega=\pi/T}$$

Observa-se também que uma *VSDI* finita garante um zero espectral na frequência de Nyquist, ou seja:

$$W_y(\pi/T) = \lim_{N \rightarrow \infty} |Y_N(\pi/T)|^2 \leq \lim_{N \rightarrow \infty} \frac{1}{2NT} VSDI^2 = 0 \quad (3.8)$$

Portanto, se a *SDI* estiver confinada entre dois valores finitos, certamente existirá um zero espectral na frequência de Nyquist, $f = \pi/T$.

Após a introdução dos novos parâmetros *SDI* e *VSDI*, será estabelecida uma condição para códigos de linha de faixa mínima.

Proposição:

Em sistemas ternários de faixa mínima que operam com excesso de faixa nulo, a abertura dos olhos será diferente de zero (olhos abertos) se a *VSDI* do código for finita.

Prova:

Seja $s(t)$ o sinal de linha dado por:

$$\begin{aligned} s(t) &= \sum_{n=-\infty}^{\infty} y_n \frac{\text{sen} \left| \frac{\pi(t-nT)}{T} \right|}{\frac{\pi(t-nT)}{T}} = \sum_{n=-\infty}^{\infty} y_n \frac{\text{sen}(\pi t/T - n\pi)}{\pi(t-nT)/T} = \\ &= \sum_{n=-\infty}^{\infty} y_n (-1)^n \frac{\text{sen} \frac{\pi t}{T}}{\frac{\pi(t-nT)}{T}} = \frac{T}{\pi} \text{sen} \frac{\pi t}{T} \sum_{n=-\infty}^{\infty} (-1)^n \frac{y_n}{t-nT} \end{aligned}$$

Os olhos correspondentes ao caso ternário são mostrados na Fig.3.8.

Seja $VSDI = M$ e $0 \leq t < T/2$.

Sejam também $U_1(t)$ e $L_1(t)$ o teto e o piso do olho, respectivamente e sejam ainda:

$$z_n \triangleq (-1)^n y_n \quad (3.9)$$

$$v_n(t) \triangleq \frac{z_n}{t-nT} \quad (3.10)$$

então:

$$s(t) = \frac{T}{\pi} \text{sen} \frac{\pi t}{T} \sum_{n=-\infty}^{\infty} v_n(t) \quad (3.11)$$

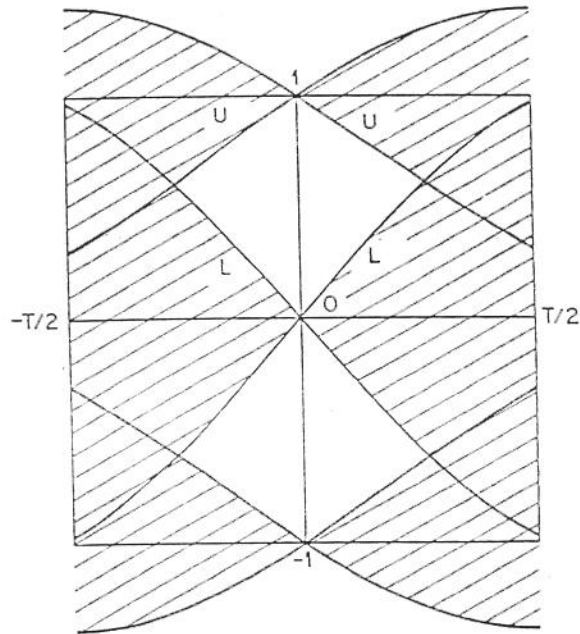


Figura 3.9: Diagrama de olho de um código ternário

Observando-se a Fig.3.1, nota-se que $U_1(t)$ começa em $s(0) = 1$ e $L_1(t)$ em $S(0) = 0$.

$$U_1(t) = \frac{T}{\pi t} \text{sen}\left(\frac{\pi t}{T}\right) + \min[D(t)] \quad (3.12)$$

$$L_1(t) = \text{máx}[D(t)] \quad (3.13)$$

onde:

$$D(t) = \frac{T}{\pi} \text{sen}(\pi t/T) \sum' v_n(t)$$

$$e \sum' = \sum_{n=-\infty}^{\infty}, \text{ com } n \neq 0$$

Portanto, para determinar $L_1(t)$, basta maximizar $D(t)$ com $y_0 = 0$. Obtém-se o valor máximo de $D(t)$ quando se tem uma seqüência de pulsos $z_n = -1$ a partir de $n = 1, n \geq 1$. O tamanho desta seqüência é limitada pela $VSDI$. Como $VSDI = M$, o número máximo de pulsos $z_n = -1$ consecutivos que se pode ter é M , de $n = 1$ até $n = M$. Para $n = M + 1$ z_n tem que ser necessariamente diferente de -1 , havendo portanto, para $n > M$ mais pulsos $z_n = 1$ do que $z_n = -1$, caso contrário, a $VSDI$ seria maior que M .

Se os termos dos pulsos diferentes de zero em $n > M$ forem reindexados para:

$$m_1, m_2, m_3, \dots, m_J, \text{ se } z_n = 1$$

$$n_1, n_2, n_3, \dots, n_J, \text{ se } z_n = -1$$

então, pela restrição da $VSDI$, onde $0 \leq SDI[M + 1, k] \leq M$, para $k \geq M + 1$ e $I \geq J$, tem-se:

$$m_1 < n_1, m_2 < n_2, \dots, m_J < n_J$$

Portanto, no intervalo $[M + 1, k]$:

$$\begin{aligned} \sum_{n=M+1}^k v_n(t) &= \sum_{i=1}^I \frac{1}{t - m_i T} - \sum_{i=1}^J \frac{1}{1 - n_i T} = \\ &= \sum_{i=1}^J \left[\frac{1}{t - m_i T} - \frac{1}{1 - n_i T} \right] - \sum_{i=J+1}^I \frac{1}{t - m_i T} \end{aligned}$$

O primeiro termo da equação acima é negativo, pois $m_i < n_i$ e o intervalo de tempo considerado é $0 \leq t < T/2$ e o segundo termo também é negativo, pois $t - m_i T < 0$ ($0 \leq t < T/2$ e $m_i > 1$). Como a contribuição total dos termos de $v_n(t) < 0$ ($z_n = 1$), para $n \geq M + 1$, excede em valor absoluto a dos termos de $v_n(t) > 0$ ($z_n = -1$), então qualquer pulso diferente de zero diminuiria o valor de $D(t)$. Conclue-se, portanto, que para $n > 0$ o somatório de $D(t)$ é minimizado quando:

$$(-1)^n y_n = \begin{cases} -1 & 1 \leq n \leq M \\ 0 & n \geq M + 1 \end{cases} \quad (3.14)$$

Analogamente, o somatório de $D(t)$, para $n < 0$ é maximizado quando:

$$(-1)^n y_n = \begin{cases} 1 & -M \leq n \leq -1 \\ 0 & n \leq -M - 1 \end{cases} \quad (3.15)$$

Com as Eqs. (3.14) e (3.15) acima obtidas, para $0 \leq t < T/2$, tem-se:

$$L_1(t) = \frac{T}{\pi} \operatorname{sen} \frac{\pi t}{T} \left[\sum_{n=1}^M \frac{1}{nT + t} + \sum_{n=1}^M \frac{1}{nT - t} \right] \quad (3.16)$$

$$U_1(t) = \frac{T}{\pi} \operatorname{sen} \frac{\pi t}{T} \left[1/T - \sum_{n=1}^M \frac{1}{nT + t} - \sum_{n=1}^{M-1} \frac{1}{nT - t} \right] \quad (3.17)$$

Verifica-se que $L_1(t)$ é monotonicamente crescente e $U_1(t)$ é monotonicamente decrescente. Portanto, em algum t_0 , $0 < t_0 < T/2$:

$$L_1(t_0) = U_1(t_0)$$

Isto significa também que o olho está aberto no intervalo $[0, t_0)$.

Para obter o lado esquerdo do olho, ou seja, para $-T/2 < t \leq 0$, basta repetir o procedimento anterior para se obter:

$$L_1(t) = -\frac{T}{\pi} \operatorname{sen} \frac{\pi t}{T} \left[\sum_{n=1}^M \frac{1}{nT + t} + \sum_{n=1}^M \frac{1}{nT - t} \right] \quad (3.18)$$

$$U_1(t) = -\frac{T}{\pi} \operatorname{sen} \frac{\pi t}{T} \left[1/T - \sum_{n=1}^M \frac{1}{nT + t} - \sum_{n=1}^{M-1} \frac{1}{nT - t} \right] \quad (3.19)$$

Pelas Eqs. (3.16), (3.17), (3.18) e (3.19), nota-se a simetria do olho em relação à origem do tempo, quando se considera o intervalo $-T/2 < t < T/2$.

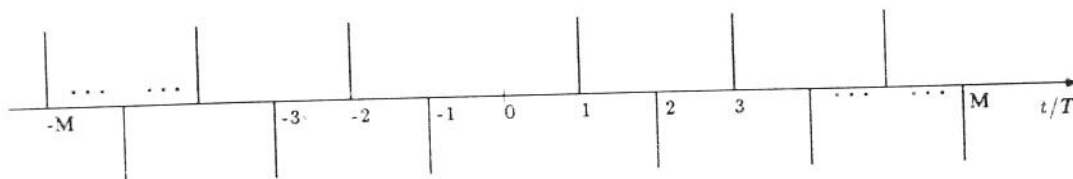


Figura 3.10: Seqüência de pior caso para $L_1(t), 0 \leq t < T/2$

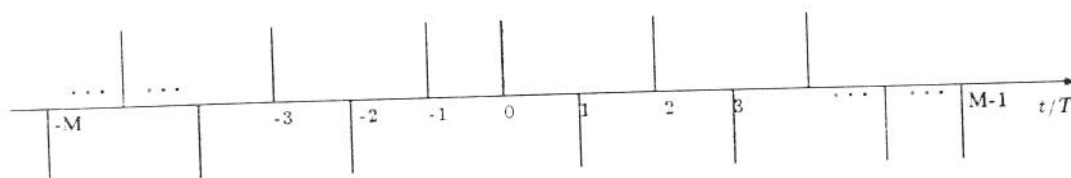


Figura 3.11: Seqüência de pior caso para $U_1(t), 0 \leq t < T/2$

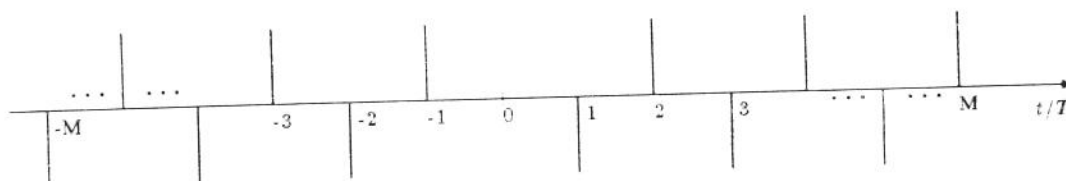


Figura 3.12: Seqüência de pior caso para $L_1(t), -T/2 < t \leq 0$

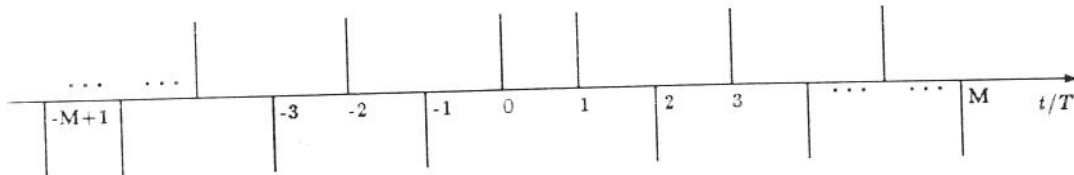


Figura 3.13: Sequência de pior caso para $U_1(t)$, $-T/2 < t \leq 0$

Aplicando-se exatamente o mesmo procedimento para o olho inferior, nota-se que há simetria em relação ao eixo horizontal do olho inferior em relação ao superior. Uma vez obtido os olhos do sistema ternário com $VSDI = M$, idealmente limitado em banda à frequência de Nyquist, verificou-se que eles estão abertos com largura $2t_0$. Finalmente conclue-se que tal sistema é um sistema de faixa mínima.
c.q.d.

Tomando-se como exemplo o pulso de Nyquist, o diagrama de olho confunde-se com o próprio pulso nos instantes $t = kT$. Portanto, quando calcula-se a derivada do pulso no tempo nestes mesmos instantes tem-se uma boa noção da abertura do olho, pois tal derivada é finita e é numericamente igual à tangente do pulso no instante em questão. Um ângulo entre a reta tangente em $t = kT$ e o eixo dos tempos, menor que 90° representa um diagrama de olho aberto.

$$s'(t) = \cos(\pi t/T) \sum' (-1)^n \frac{a_n}{t - nT} + t/\pi \operatorname{sen}(\pi t/T) \frac{d}{dt} \sum' (-1)^n \frac{a_n}{t - nT}$$

$$s'(t) |_{t=0} = \sum' (-1)^n \frac{a_n}{-nT} = \frac{-1}{T} \sum' (-1)^n \frac{a_n}{n} \tag{3.20}$$

Nos casos em que os pulsos centrais forem -1 e 0 , $s'(t) |_{t=0}$ fornece uma noção de $U_1(t)$ e $L_1(t)$, respectivamente. Assim, se a $VSDI$ for limitada, $s'(0)$ também o será e, conseqüentemente o ângulo θ será menor que 90° e o olho estará aberto.

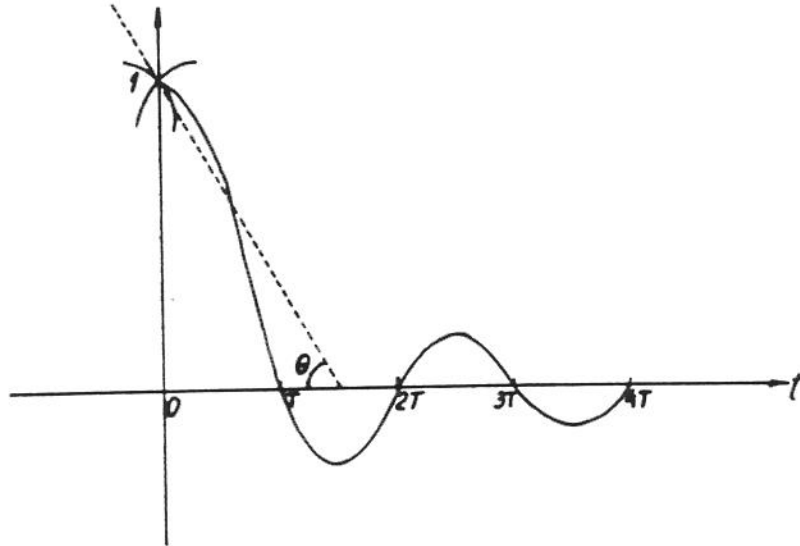


Figura 3.14: Abertura do diagrama de olho

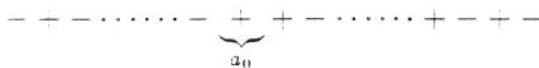
Como $VSDI = M$, na Eq.(3.20) tem-se no máximo M parcelas a mais de uma polaridade do que de outra, ou seja, se o pulso central for positivo, tem-se no máximo $(M - 1)$ parcelas consecutivas no somatório em $(-\infty, -1]$ e, no máximo M parcelas negativas em $[1, \infty)$.

Se o pulso for zero tem-se no máximo M parcelas positivas(negativas) consecutivas em $(-\infty, -1]$ e no máximo M parcelas negativas(positivas) em $[1, \infty)$.

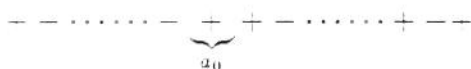
De acordo com as Figs.3.14 e 3.15, conclue-se que as seqüências que maximizam $s'(0)$ são:

- $a_0 = +$

1. M par



2. M ímpar



Uma vez determinada a seqüência que maximiza $s'(0)$, determinou-se o seu valor máximo.

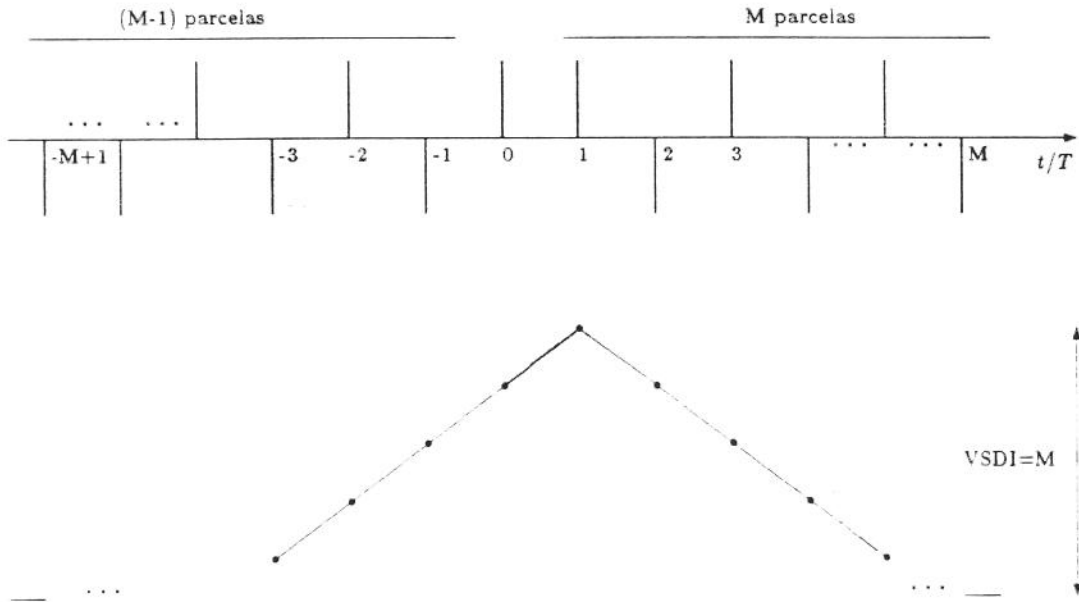


Figura 3.15: Seqüência de pior caso para $U(t)$. O pulso analisado é positivo

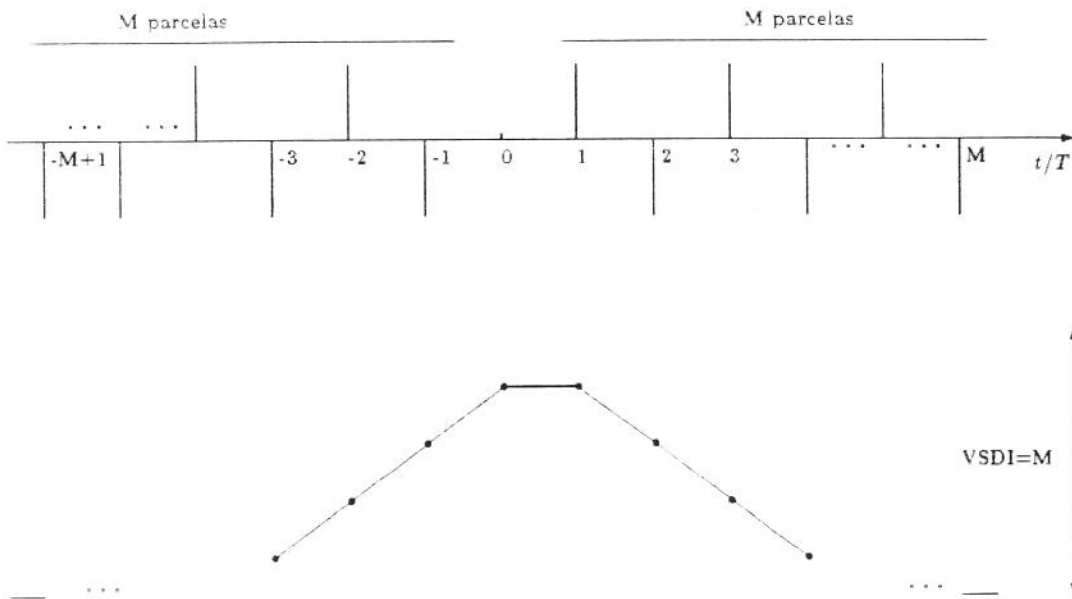


Figura 3.16: Seqüência de pior caso para $L(t)$. O pulso analisado é zero

$$\begin{aligned} \max |s'(0)| &= \frac{1}{T} \left[\frac{1}{M-1} + \dots + \frac{1}{3} + \frac{1}{2} + 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{M} \right] \\ \Rightarrow \max |s'(0)| &= \frac{2}{T} \left[1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{M-1} \right] + \frac{1}{MT} \end{aligned}$$

- $a_0 = 0$

1. M par

$$\text{---} \dots \text{---} \underbrace{0}_{a_0} \text{---} \dots \text{---}$$

2. M ímpar

$$\text{---} \dots \text{---} \underbrace{0}_{a_0} \text{---} \dots \text{---}$$

Para $a_0 = 0$, o valor máximo de $s'(0)$ será:

$$\begin{aligned} \max |s'(0)| &= \frac{1}{T} \left[\frac{1}{M} + \dots + \frac{1}{3} + \frac{1}{2} + 1 - \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{M} \right] \\ \Rightarrow \max |s'(0)| &= \frac{2}{T} \left[1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{M} \right] \end{aligned}$$

Como já mencionado no início deste capítulo, o DB e o DBM são os códigos de faixa mínima mais conhecidos. Além destes existem também os códigos propostos por *Kim*[6], que são variações dos códigos Bipolar, DB e DBM.

Na Tabela 3.1 encontram-se resumidas as principais características de cada um destes códigos. Nota-se que apenas o Bipolar não é de faixa mínima, porém é balanceado, enquanto que o DB é de faixa mínima, mas não é balanceado. Portanto, o único código na tabela que é balanceado e de faixa mínima é o DBM.

Código	Função do Sistema $F(D^*)$	Característica de Transferência		Resposta temporal $h(t)$		Espectro de potência $W_x(\omega)$	
		Resposta em frequência $ H(\omega) $					
Bipolar (BP)	$1 - D$	$2T \text{sen} \frac{\omega T}{2}$ 	$\text{sinc}(t/T) - \text{sinc}(t/T - 1)$ 		$T \text{sen}^2 \frac{\omega T}{2}$ 		
Duobinário (DB)	$1 + D$	$2T \cos \frac{\omega T}{2}$ 	$\text{sinc}(t/T) + \text{sinc}(t/T - 1)$ 		$T \cos^2 \frac{\omega T}{2}$ 		
Duobinário Modificado (DBM)	$1 - D^2$	$2T \text{sen} \omega T$ 	$\text{sinc}(t/T) - \text{sinc}(t/T - 2)$ 		$T \text{sen}^2 \omega T$ 		

Tabela 3.1: Características

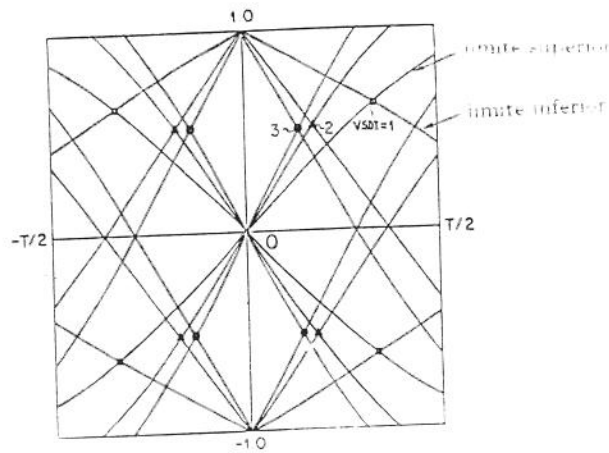


Figura 3.17: Diagrama de olho

A Fig.3.17 mostra o diagrama de olho para os diversos valores de $VSDI$ e a Tabela 3.2 lista $Ts'_0(0)$ e a largura do olho para cada $VSDI$. É interessante observar que estes valores são os de pior caso para uma determinada $VSDI = M$ e que qualquer sistema ternário de faixa mínima terá um olho mais aberto ou pelos menos de mesma abertura para a mesma $VSDI$.

Deve-se notar também que a largura do olho para $VSDI = 1$ é igual à largura do DB e para $VSDI = 2$ é igual à do DBM. Isto significa que o DB e o DBM são os códigos de faixa mínima que apresentam a largura de olho mínima para $VSDI = 1$ e $VSDI = 2$, respectivamente.

Assim como no Capítulo 2, é possível determinar-se os limites de eficiência para cada valor de SDI . O procedimento é idêntico ao do capítulo anterior, já que para se obter a $VSDI$ a partir da $VSDC$, basta inverter os bits ímpares. Por isso esse processo não será repetido aqui, pois o mapa Redução de velocidade $\times VSDI$ será idêntico ao já determinado, bastando apenas substituir no eixo das abscissas $VSDC$, ou P por $VSDI$.

A Fig.3.18 mostra o mapa de códigos eficientes quanto à SDI e, na parte superior

VSDI	$Ts'_-(0)$	$Ts'_0(0)$	largura do olho (T)
1	1	2	0,667
2	2,5	3	0,357
3	3,334	3,667	0,282
4	3,917	4,167	0,245
5	4,367	4,567	0,222
6	4,534	4,701	0,207
7	4,844	4,987	0,195
8	5,112	5,237	0,185
∞	∞	∞	0

Tabela 3.2: Abertura do olho

estão representados os valores de largura do olho variando com a $VSDI$, $1 \leq VSDI \leq 8$, conforme mostra a Tab.3.2.

Sabendo-se que a largura do olho para o código DB é 0,667 e para o DBM é 0,357[2], conclue-se que o DB constitui-se no código que produz a pior abertura de olho para $VSDI = 1$ e, o DBM para $VSDI = 2$. Ou seja, qualquer código com a mesma $VSDI$, como por exemplo, $VSDI = 2$, possuirá diagrama de olho pelo menos tão aberto quanto o do DBM. Portanto, existem códigos tão eficientes quanto os de resposta parcial no que diz respeito à ocupação da faixa espectral e que apresentam as seguintes vantagens adicionais:

- No caso do DB, $VSDI = 1$, *não existe* possibilidade de redução de velocidade sobre o código *1BIT*, que é o próprio DB, mantendo-se a abertura do olho;
- No caso do DBM, que tem $VSDI = 2$, pode-se reduzir a velocidade em até 21.36%, com sacrifícios crescentes no grau de balanceamento, mas sem degradação na abertura do diagrama de olho.

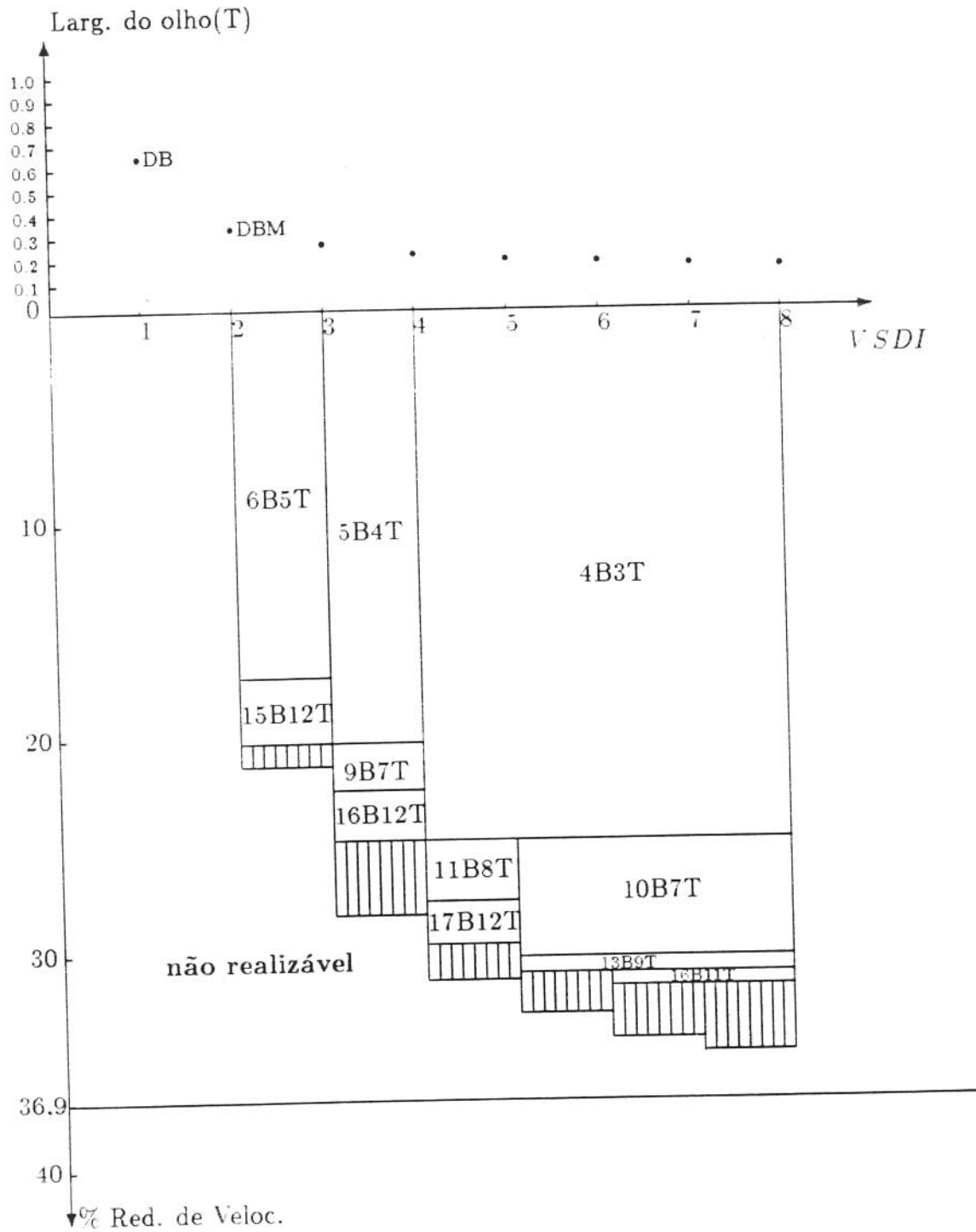


Figura 3.18: Mapa de Códigos Eficientes

Capítulo 4

CÓDIGOS TERNÁRIOS BALANCEADOS DE FAIXA MÍNIMA

4.1 Condição de Balanceamento e Faixa Mínima

Nos capítulos anteriores foram estabelecidas condições suficientes para códigos ternários balanceados e de faixa mínima, separadamente. Constatou-se que não existe relação alguma entre o código ser balanceado e de faixa mínima, ou seja, existem códigos balanceados que não são de faixa mínima, existem os que são de faixa mínima não balanceados e, finalmente os balanceados de faixa mínima.

A proposta deste capítulo é estabelecer uma condição suficiente para o projeto de um código que possua as duas propriedades: balanceamento e faixa mínima. Dos capítulos anteriores, conclui-se que tal condição é estabelecida se for garantido o confinamento da $VSDC$ e $VSDI$, isto é:

$$\left| \sum_{n=1}^J y_n \right| \leq M \quad (4.1)$$

$$\left| \sum_{n=1}^J (-1)^n y_n \right| \leq M \quad (4.2)$$

Se as Eqs.(4.3) e (4.4) forem somadas e subtraídas, obtém-se as condições de balanceamento e de faixa mínima dada pelas Eqs.(4.1) e (4.2).

$$\sum_{n\text{ ímpar}} y_n \leq \frac{M}{2} \quad (4.3)$$

$$\left| \sum_{n \text{ par}} y_n \right| \leq \frac{M}{2} \quad (4.4)$$

Portanto, para gerar uma seqüência balanceada de faixa mínima, basta garantir que a seqüência de dígitos pares e ímpares sejam balanceadas separadamente.

$$SDC = \sum_{n=1}^J y_n = \sum_{n \text{ par}} y_n - \sum_{n \text{ ímpar}} y_n \quad (4.5)$$

$$SDI = \sum_{n=1}^J (-1)^n y_n = \sum_{n \text{ par}} y_n - \sum_{n \text{ ímpar}} y_n \quad (4.6)$$

Baseado nas Eqs.(4.5) e (4.6), a Soma Digital Vetorial é definida como:

$$S\bar{D}V = (SDCI, SDCP) \quad (4.7)$$

onde:

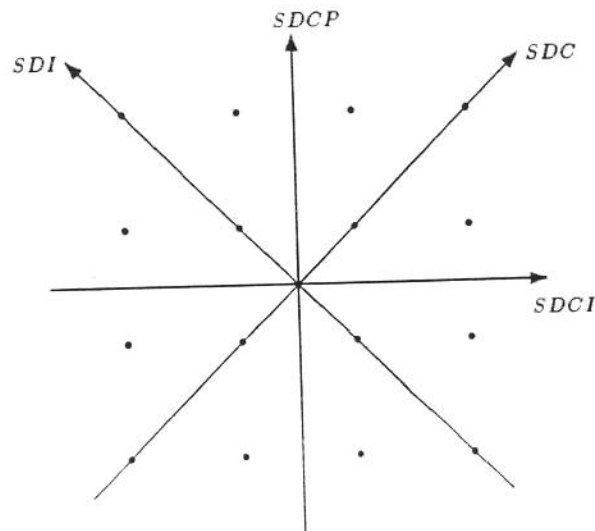
$$SDCI = \sum_{n=1}^J y_n, n \text{ ímpar}$$

$$SDCP = \sum_{n=1}^J y_n, n \text{ par}$$

Portanto:

$$SDC = SDCI + SDCP \quad (4.8)$$

$$SDI = -SDCI + SDCP \quad (4.9)$$

Figura 4.1: Representação dos estados no plano SDV

Então, os estados podem ser representados por pontos no plano da SDV , conforme a Fig.4.1.

Uma maneira bastante simples de gerar seqüências balanceadas de faixa mínima é intercalar duas seqüências balanceadas. Considerou-se, por exemplo, o código $4B3T$ balanceado, apresentado na Fig.4.2. Na saída do intercalador, tem-se uma seqüência c_n balanceada de faixa mínima. Entretanto, a seqüência resultante não se constitui num código de bloco $4B3T$.

O exemplo mais simples e mais conhecido desta classe de códigos é o duobinário modificado, cuja disposição de estados é mostrada na Fig.4.3. O código DBM é o tipo $1B1T$ e que, portanto, não apresenta redução na taxa de símbolos. A $VSDC$ é igual à $VSDI$ que é igual a 2, podendo a SDC e a SDI variar entre -1 e 1 . Outro exemplo de código balanceado e de faixa mínima é o $VMDB$, [6], que é uma variação do DBM e que também não proporciona redução de velocidade, mas viabiliza a proibição de longas seqüências de zeros através da substituição de padrões.

Para otimizar a utilização da faixa de transmissão é interessante investigar-se a existência de códigos que, além de possuírem as propriedades mencionadas anteriormente, produzam redução de velocidade.

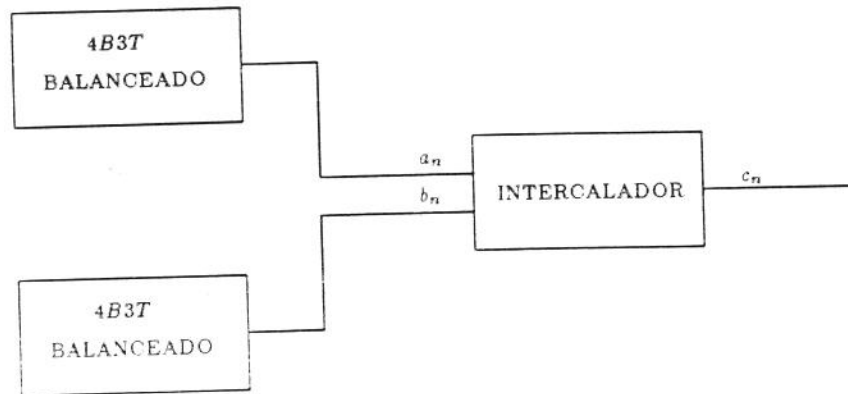


Figura 4.2: Esquema para gerar uma seqüência balanceada de faixa mínima

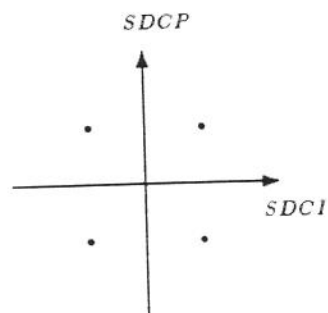


Figura 4.3: Diagrama de estados do DBM

4.2 Existência de Códigos Balanceados e de Faixa Mínima

De acordo com o mapa de Redução de velocidade versus Desbalanço, mostrado na Fig.2.3, observa-se que o código de menor complexidade que apresenta redução de velocidade é o $4B3T$. Nesta secção será analisada a seguinte questão: É possível construir um código $4B3T$ balanceado de faixa mínima? Se não, qual é o código de menor complexidade com essas propriedades?

Analisando-se os códigos do tipo $mB3T$, tem-se:

$$\text{com 3 bits ternários} \begin{cases} 2 & \text{ímpares} \\ 1 & \text{par} \end{cases} \quad \text{ou vice-versa}$$

Para os dígitos ímpares obtém-se 6 combinações possíveis de disparidade não negativa:

$$00, 0-, -0, --, ++, ---$$

e para os pares:

$$0, -$$

Desta forma, dispõe-se de $2 \times 6 = 12$ combinações de três dígitos ternários que atendem as restrições necessárias. Mas $12 < 16$, que seria o número de palavras exigido para mapear 4 dígitos binários a fim de se obter o código $4B3T$. Portanto, é impossível gerar um código $4B3T$ balanceado de faixa mínima, com quatro alfabetos, um para cada quadrante da Soma Digital Vetorial.

Verificar-se-á, então para $n = 4$.

$$\text{com 4 bits ternários} \begin{cases} 2 & \text{ímpares} \\ 2 & \text{pares} \end{cases}$$

Neste caso dispõe-se de 36 palavras ternárias de comprimento $n = 4$ para cada quadrante. Como $36 > 32 = 2^5$, existe uma classe de códigos $5B4T$ balanceados de faixa mínima e que se constitui no código de bloco menos complexo, já que continuando esta investigação descobriu-se que o $6B5T$ e $8B6T$ também o são. Constatou-se também que o $6B5T$ além de ser mais complexo, proporciona menor redução de velocidade.

Para a classe $5B4T$, dispõe-se de 81 palavras ternárias, das quais apenas 36 palavras podem ser utilizadas para cada um dos quatro alfabetos para obter-se um código balanceado de faixa mínima.

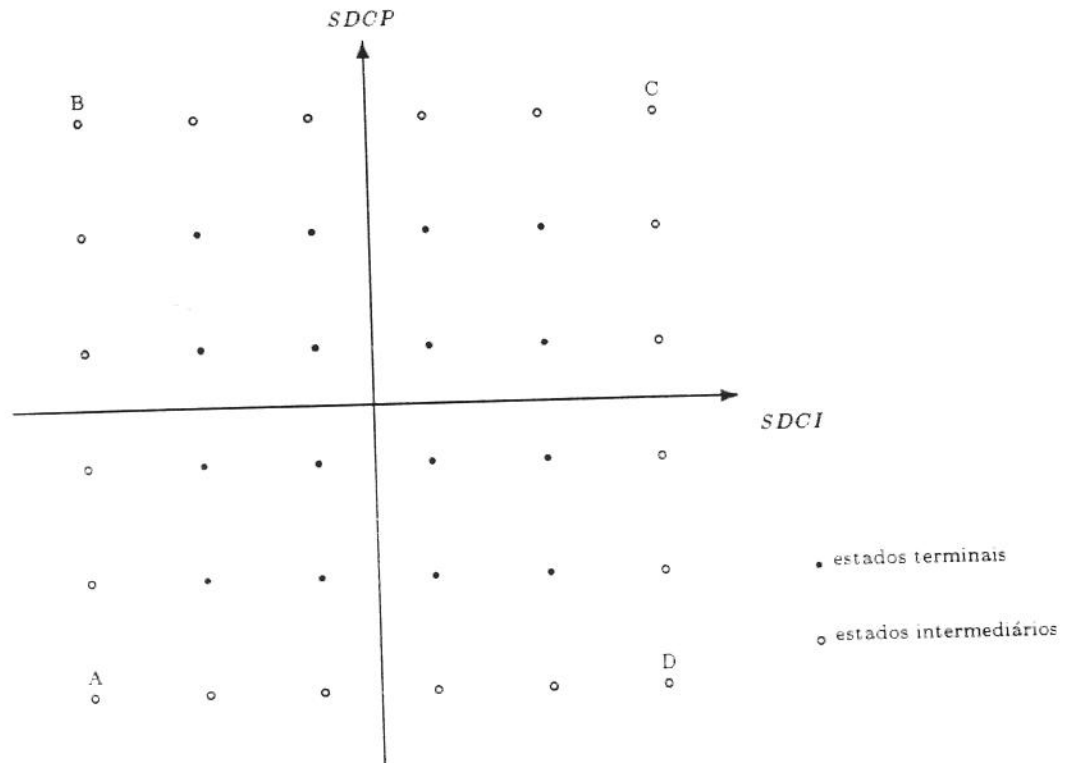


Figura 4.4: Diagrama de estados do 5B4T

Se não houver qualquer tipo de restrição às palavras, o diagrama de estados no plano da SDC é apresentado na Fig.4.4.

Conforme a Fig.4.4 a $VSDC = VSDI = 10$. Se as palavras $--++$, $-++-$, $++--$ e $+- --$ forem proibidas nos alfabetos usados a partir do primeiro, segundo, terceiro e quarto quadrantes da SDV , respectivamente os estados extremos A, B, C e D serão evitados, fazendo com que $VSDC = VSDI = 8$. Dispõe-se ainda de 35 palavras para cada alfabeto, ou seja, pode-se excluir 3 palavras de cada um deles, pois são necessárias apenas 32 palavras.

Pode-se confinar ainda mais a SDC e a SDI se forem considerados apenas doze estados terminais com doze alfabetos, um alfabeto para cada estado.

A configuração mostrada na Fig.4.5 corresponde à melhor condição de confinamento da SDC e SDI , que são iguais a 6, fixando o tamanho do bloco ternário em $n = 4$. Porém, a complexidade é maior em relação à configuração da Fig.4.4, pois há a necessidade de 12 alfabetos (um alfabeto para cada estado terminal).

Para uma análise mais detalhada do limite de eficiência, determinou-se a matriz de transição, considerando movimentos de dois dígitos, que é dada por:

$$A = \begin{pmatrix} 4 & 4 & 1 \\ 2 & 3 & 1 \\ 1 & 2 & 1 \end{pmatrix}$$

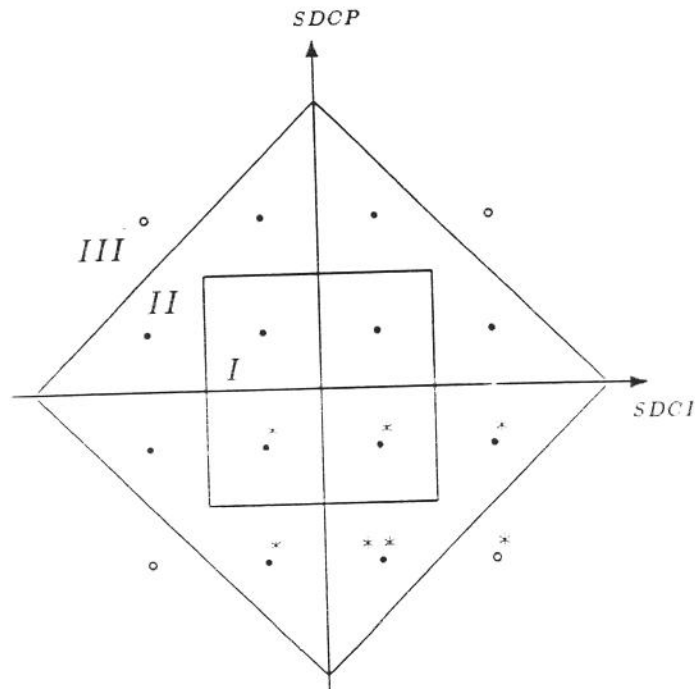


Figura 4.5: Diagrama de estados dos códigos 5B4T com 12 estados terminais

A matriz A foi obtida da seguinte maneira: observando a Fig.4.5 nota-se que os estados podem ser divididos na forma indicada. Isto é possível, pois o número de palavras que chegam aos estados do tipo I , tanto vindos dos estados do tipo II como dos intermediários ou mesmo vindos deles mesmos, é igual. O mesmo ocorre com os estados do tipo II e os intermediários foram considerados como um outro tipo. Por exemplo, tomou-se como estado inicial o mais externo, indicado por $**$ que é o caso mais crítico. Em movimentos de dois dígitos existe a possibilidade de se estar em qualquer um dos seis estados assinalados com um (*), sendo que três delas vão para estados do mesmo tipo, duas vão para os do tipo I e apenas uma vai para o do tipo III . Assim é obtida a segunda linha da matriz de transição, sendo que para obter as outras basta repetir o processo.

Se a matriz A for multiplicada por ela mesma, obtém-se o número de palavras disponíveis para esta configuração da Fig.4.5 para códigos do tipo 4T.

$$A^2 = B_4 = \begin{pmatrix} \mathbf{25} & \mathbf{30} & 9 \\ \mathbf{15} & \mathbf{19} & 6 \\ 9 & 12 & 4 \end{pmatrix}$$

Somando os números destacados em negrito em cada linha, o menor número obtido foi $15 + 19 = 34 > 32 = 2^5$. Mesmo proibindo a palavra 000, ainda restam 33 palavras para mapear as 32 palavras binárias. Os outros números da matriz não foram considerados porque os estados do tipo III não são terminais.

Uma vez determinada a matriz de transição A , a equação característica é dada por:

$$\det \begin{bmatrix} 4 - \lambda & 4 & 1 \\ 2 & 3 - \lambda & 1 \\ 1 & 2 & 1 - \lambda \end{bmatrix} = 0$$

$$\begin{aligned} (4 - \lambda)(\lambda^2 - 4\lambda + 1) - 4(1 - 2\lambda) + (1 + \lambda) &= 0 \\ -\lambda^3 + 8\lambda^2 - 8\lambda + 1 &= 0 \\ -\lambda^3 + 8\lambda^2 + 8\lambda - 1 &= 0 \\ (\lambda - 1)(-\lambda^2 + 7\lambda - 1) &= 0 \end{aligned}$$

$$\Rightarrow \lambda_{m\acute{a}x} = 6.85$$

$$\left(\frac{m}{n}\right)_{m\acute{a}x} = \frac{1}{2} \log_2 \lambda_{m\acute{a}x} = 1.388$$

$$\Delta\% = \left[1 - \frac{2}{\log_2 \lambda_{m\acute{a}x}}\right] 100\% = 27.96\%$$

Portanto, o limite assintótico de eficiência para a configuração da Fig.4.5 é de 27.96%. sendo que a redução para a classe 5B4T proposta é de 20%. Com o uso de blocos mais longos ($n > 4$) a redução poderia se aproximar bem mais do limite assintótico, mas às custas de maior complexidade do código.

Outro tipo de análise que se pode fazer é, a partir de uma dada configuração, investigar qual o código mais simples que proporciona alguma compressão da velocidade de transmissão. Para melhor compreensão tomou-se como exemplo, a configuração mostrada na Fig.4.6 que apresenta $VSDI$ igual à do DBM , o que significa que o diagrama de olho correspondente deve ser semelhante ao do DBM .

O número de palavras que saem de cada estado pode ser determinado a partir das matrizes de transições quando ocorre um bit ímpar ou par. T_i e T_p , respectivamente.

$$T_i = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ e } T_p = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

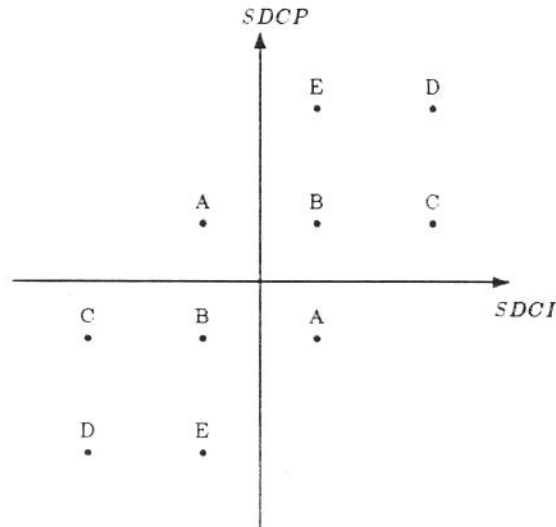


Figura 4.6: $VSDC = 6$ e $VSDI = 2$

Para dois bits tem-se:

$$T_2 = T_i T_p = \begin{pmatrix} 2 & 2 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow \bar{n}_2 = \begin{pmatrix} 5 \\ 7 \\ 5 \\ 4 \\ 4 \end{pmatrix} \Rightarrow 2B2T$$

Para quatro bits:

$$T_4 = T_2 T_2 = \begin{pmatrix} 8 & 9 & 3 & 5 & 3 \\ 9 & 11 & 5 & 7 & 5 \\ 5 & 7 & 4 & 5 & 4 \\ 3 & 5 & 4 & 4 & 4 \\ 3 & 5 & 4 & 4 & 4 \end{pmatrix} \Rightarrow \bar{n}_4 = \begin{pmatrix} 28 \\ 37 \\ 25 \\ 20 \\ 20 \end{pmatrix} \Rightarrow 4B4T$$

Neste caso, se apenas os estados A e B forem considerados como estados terminais, tem-se, no mínimo, 17 palavras, viabilizando um código $4B4T$ que permite a proibição da palavra 0000, o que faz com que se evite longas seqüências de palavras sem transição de nível. Esta é uma maneira alternativa para se limitar o número de zeros consecutivos, a outra maneira, já comentada, é a de substituição de padrões, sugerida por Kim.

Para seis bits:

$$T_6 = T_4 T_2 = \begin{pmatrix} 37 & 45 & 20 & 28 & 20 \\ 45 & 57 & 28 & 37 & 28 \\ 28 & 37 & 20 & 25 & 20 \\ 20 & 28 & 17 & 20 & 17 \\ 20 & 28 & 17 & 20 & 17 \end{pmatrix} \Rightarrow \vec{n}_6 = \begin{pmatrix} 150 \\ 195 \\ 130 \\ 102 \\ 102 \end{pmatrix} \Rightarrow 6B6T$$

Novamente, se apenas A e B forem considerados estados terminais, tem-se 82 palavras ternárias, das quais apenas 64 são necessárias. A redundância neste caso é bem mais significativa do que no caso anterior, o que permite não só a exclusão da palavra 000000 como a de outras 17 que forem convenientes. Portanto, é possível a obtenção de um código do tipo $6B6T$ com apenas quatro estados terminais.

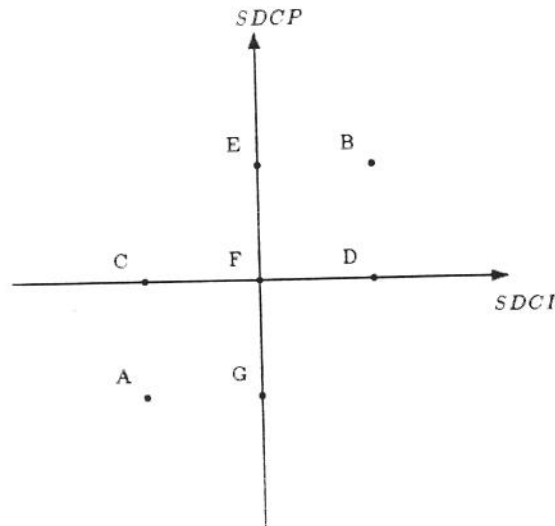
Para oito bits:

$$T_8 = T_6 T_2 = \begin{pmatrix} 184 & 232 & 113 & 150 & 113 \\ 232 & 297 & 150 & 195 & 150 \\ 150 & 195 & 102 & 130 & 102 \\ 113 & 150 & 82 & 102 & 82 \\ 113 & 150 & 82 & 102 & 82 \end{pmatrix} \Rightarrow \vec{n}_8 = \begin{pmatrix} 792 \\ 1024 \\ 579 \\ 529 \\ 529 \end{pmatrix} \Rightarrow 9B8T$$

Desta forma verifica-se que é possível obter um código que apresente diagrama de olho semelhante ao do DBM e ainda com a vantagem de reduzir a velocidade. Este código é o $9B8T$ com um alfabeto por estado, isto porque os estados mais externos D e E da Fig.4.6, que são os mais críticos, dispõem de $529 (> 2^9 = 512)$ palavras.

O método utilizado para determinar o número de palavras disponíveis é apenas um dos vários processos possíveis. Outro método para se obter o mesmo resultado é semelhante ao diagrama pantográfico[3], que baseia-se no triângulo de Pascal. Este consiste em traçar o diagrama de transições de estados, tanto nas transições de bits pares como de ímpares e após cada uma delas, verificar se o número de palavras, que é a soma do número de palavras que chegam em cada um dos estados, é suficiente para proporcionar uma redução de velocidade. Tomou-se, como exemplo, um diagrama de estados, cuja $VSDC = 4$ e $VSDI = 2$, conforme a Fig.4.7.

As sub-transições que estão representadas em pontilhado na Fig.4.8, indicam a permutação de estados que são semelhantes, no que se refere às transições devido à simetria entre eles. Por exemplo, os estados E e D comportam-se de maneira semelhante em relação aos estados B e F , lembrando apenas que o estado E interage com B nas transições de bits ímpares e com F nas de bits pares. Com D ocorre justamente o contrário, ou seja, ele interage com F nas ímpares e com B nas pares. Percebendo esta simetria entre os estados, a cada transição trocam-se as posições dos estados E e D e de C e G , obtendo-se as "borboletas".

Figura 4.7: $VSDC = 4$ e $VSDI = 2$

Verificou-se também que o código mais simples com $VSDC = 4$ e $VSDI = 2$ que proporciona redução de velocidade é o $10B9T$. Este método foi apresentado apenas a título de ilustração, pois como se pode verificar não é muito prático.

Assim, investiu-se no primeiro método que envolve multiplicação de matrizes e que, computacionalmente é de fácil realização. Para tal, desenvolveu-se o *Programa*¹ *I* que calcula as matrizes de transições, a partir das matrizes de transição de bit ímpar e par, e verifica a cada transição se há redução de velocidade, considerando todos os estados como sendo estados terminais. Este programa foi utilizado para obter códigos úteis para $VSDC = 2, 4, 6$ e 8 com $VSDI = 2$. Para avaliar a redução de velocidade de cada código útil obtido, é necessário que se calcule o limite assintótico correspondente a cada $VSDC$. Para tal, utilizou-se o método proposto no capítulo 2, onde determinava-se o maior autovalor da matriz de transição, de forma que quando se considera dois bits, o valor assintótico é dado por:

$$\Delta\% = \left[1 - \frac{2}{\log_2 \lambda_{m\acute{a}x}} \right] 100\% \quad (4.10)$$

O autovalor $\lambda_{m\acute{a}x}$ correspondente a cada matriz de transição foi obtido com o *Programa*² *II* que calcula o determinante da matriz. Para se identificar $\lambda_{m\acute{a}x}$, basta verificar qual $\lambda_{m\acute{a}x}$ é capaz de anular o determinante.

Portanto, com os dois programas mencionados determina-se os códigos úteis e os limites assintóticos para cada configuração de estados proposta.

¹ Apêndice I² Apêndice 2

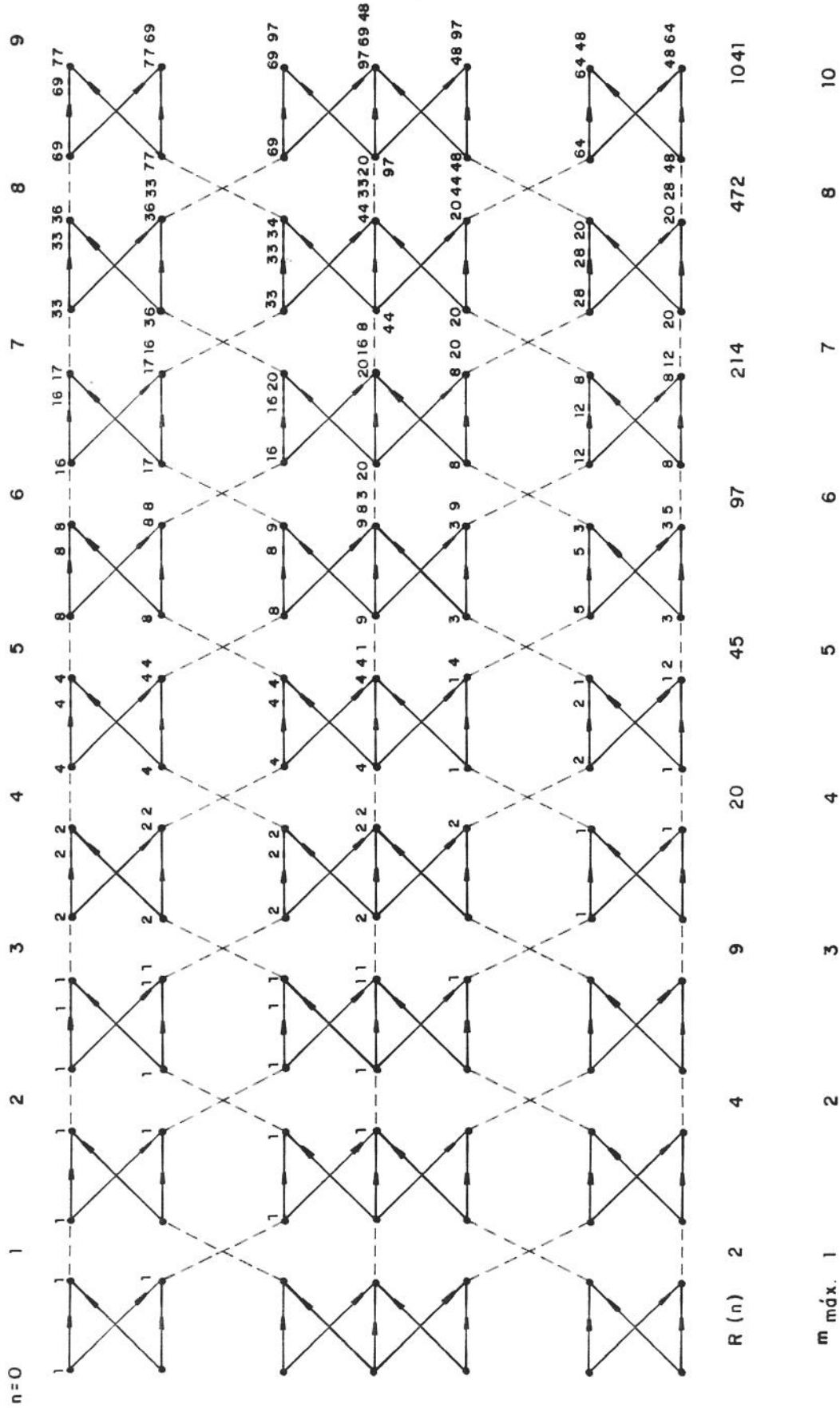


FIGURA 4.8: MÉTODO GRÁFICO

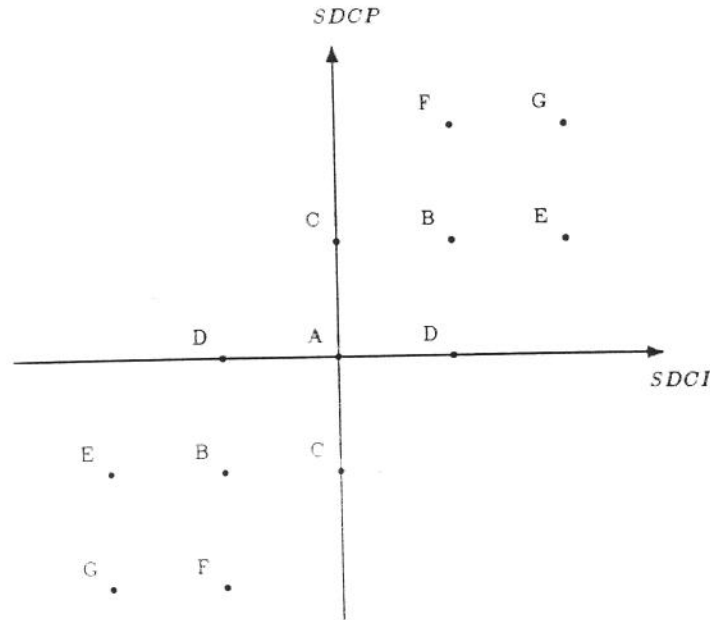


Figura 4.9: $VSDC = 8$ e $VSDI = 2$

À medida em que se aumenta a $VSDC$ a situação aproxima-se da situação discutida no Capítulo 3, para a qual $VSDC = \infty$, onde a condição de balanceamento não é satisfeita. Nesta condição, baseado na Fig.3.17 a classe de códigos $6B5T$ surge como a mais simples, com $VSDI = 2$, proporcionando uma redução de velocidade de 16,67 %.

Deve-se notar que o $6B5T$ pode ser construído com apenas três alfabetos. O primeiro para estados cuja $SDI = 1$, o segundo, que é simétrico ao primeiro, para estados cuja $SDI = -1$ e por último para $SDI = 0$. O diagrama de estados é mostrado na Fig.4.10.

Novamente, baseado no triângulo de Pascal, obtém-se o número de palavras disponíveis após cinco transições e verifica-se que para os estados mais externos, no caso de $SDI = 1$ ou $SDI = -1$, este número é suficiente.

SDI	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
1	1	2	4	9	21
0	1	2	5	12	29
-1		1	3	8	20
				70	

Portanto, mesmo excluindo a palavra 00000 dos três alfabetos, restam ainda 69 palavras das 64 necessárias para cada alfabeto.

Deve-se observar também a possibilidade de minimização do número de alfabetos via algoritmo de eliminação de estados terminais na matriz de transição. No caso de $VSDC = 8$, para o $9B8T$ se forem considerados apenas sete estados terminais, o número de

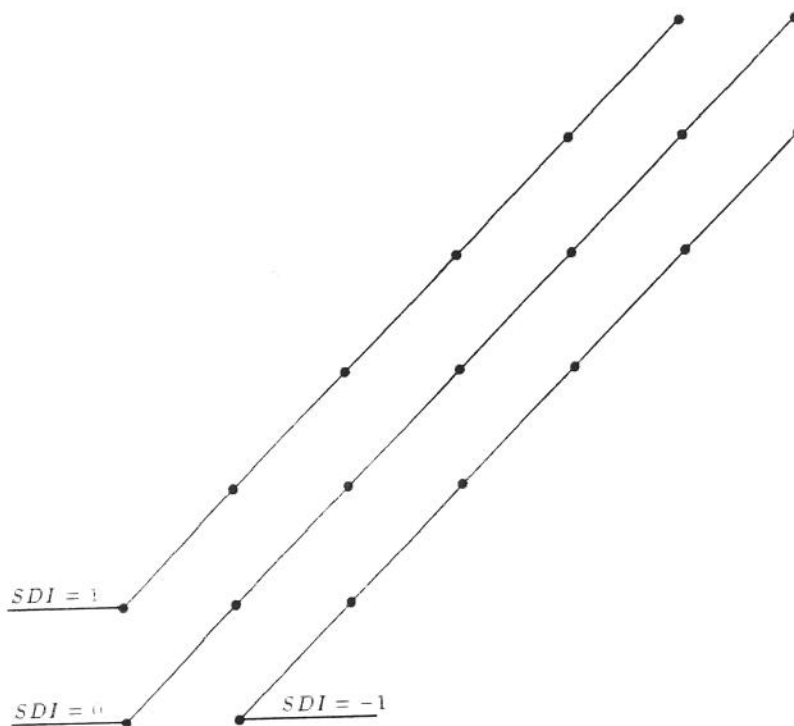


Figura 4.10: Diagrama de estados do 6B5T com $VSDC = \infty$

alfabetos necessários diminui para sete.

$$M_8 = \begin{pmatrix} 171* & 262* & 246* & 220* & 106 & 42 & 218 \\ 131* & 224* & 187* & 176* & 110 & 48 & 206 \\ 110* & 176* & 158* & 144* & 77 & 32 & 152 \\ 123* & 187* & 177* & 158* & 74 & 29 & 154 \\ 77 & 144 & 109 & 107 & 81 & 37 & 143 \\ 53 & 110 & 74 & 77 & 70 & 33 & 118 \\ 53 & 110 & 74 & 77 & 70 & 33 & 118 \end{pmatrix}$$

No caso, os estados considerados terminais foram: A, B, C e D, conforme a Fig.4.9. Se os elementos acompanhados de um *, que correspondem aos estados citados, forem somados em cada linha, verificar-se-á que o menor valor encontrado ainda será maior que 512. De fato pode-se obter um código 9B8T com apenas sete alfabetos.

Com os dados obtidos, montou-se uma tabela em que constam a redução de velocidade, o limite assintótico, o número de alfabetos em função do desbalanço, sempre considerando $VSDI = 2$. Desta forma, espera-se que os códigos que satisfazem tais condições apresentarão diagrama de olho pelo menos tão aberto quanto o do DBM, mas com a vantagem adicional de proibir longas seqüências de zeros e proporcionar redução de velocidade.

VSDC	CÓDIGOS ÚTEIS	$\Delta\%$	LIMITE ASSINTÓTICO	NÚMERO DE ALFABETOS
2	DBM	0	0	4
4	10B9T	10,00	12,37	7
	18B16T	11,11		7
6	9B8T	11,11	16,28	10
	18B16T	11,11		4
	15B13T	13,33		10
8	9B8T	11,11	17,81	7
	14B12T	14,29		7
∞	6B5T	16,67	21,36	3
	15B12T	20,00		3

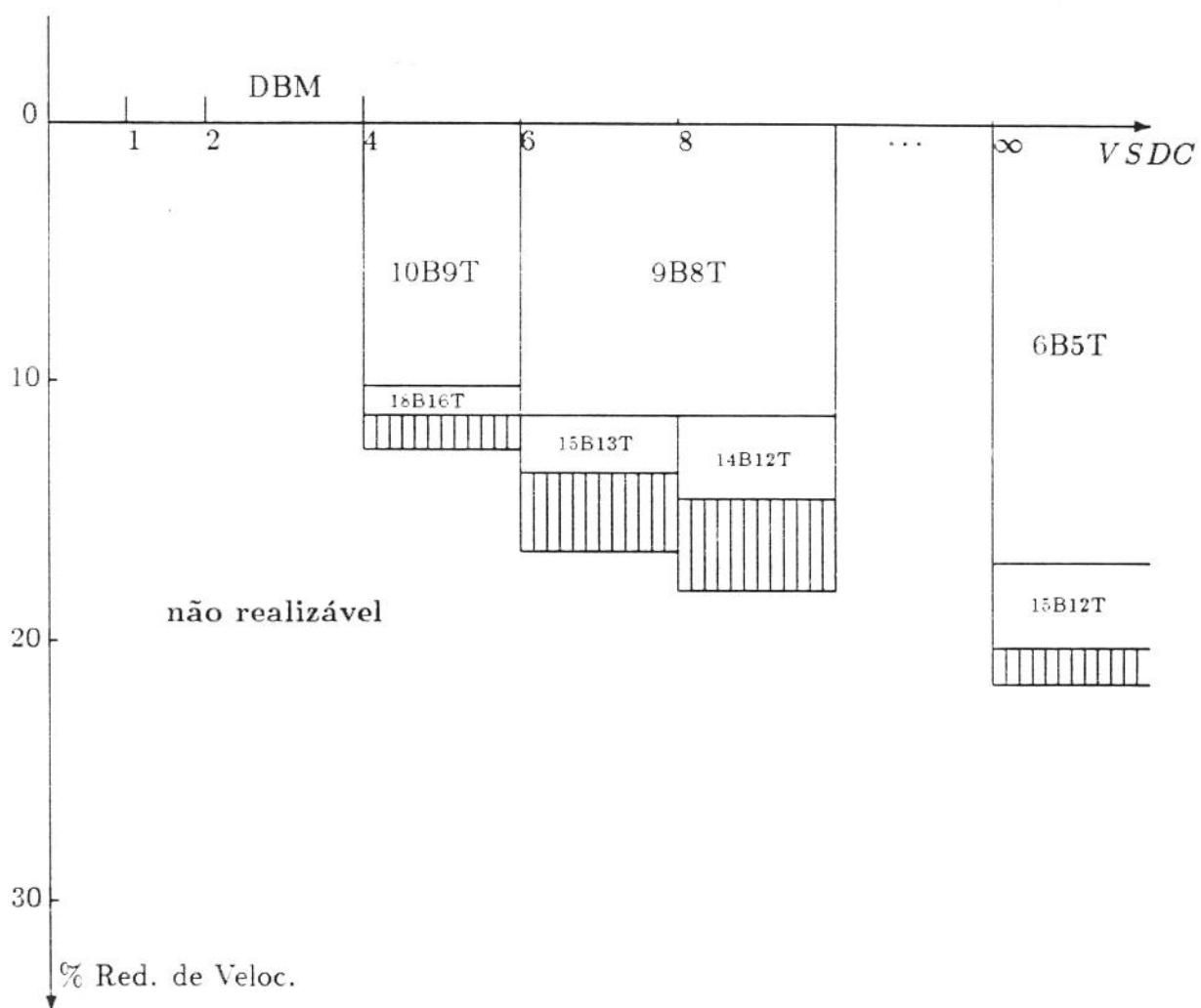


Figura 4.11: Mapa de Códigos Eficientes para $VSDI = 2$

Capítulo 5

CÁLCULO DO DIAGRAMA DE OLHO

5.1 Introdução

O diagrama de olho é uma medida de desempenho bastante conhecida e utilizada, na prática, em sistemas de transmissão digital[11]. Como a maioria dos sistemas de transmissão em banda-base utilizam codificação de linha, é comum o uso do diagrama de olho para comparar os diversos tipos de códigos de linha. Em particular, neste trabalho ele será útil para comprovar a propriedade de faixa mínima de alguns códigos propostos.

É possível observá-lo no osciloscópio quando o sinal digital é sincronizado com a taxa de símbolos. Na prática ele é obtido *a posteriori*, isto é, quando já se tem em mãos um protótipo do sistema digital. O objetivo deste capítulo é de apresentar um programa que calcule o diagrama de olho, permitindo a obtenção do mesmo *a priori*, ou seja, durante o projeto do sistema. Este programa utiliza um método computacional rápido[13] baseado no Algoritmo de Viterbi (AV)[8].

5.2 O Algoritmo de Viterbi

A princípio ele foi proposto como um método de decodificação de códigos convolucionais[14]. Desde então, ele tem se revelado, na sua forma mais geral, como uma solução para o problema de estimação da máxima probabilidade *a posteriori*, MAP. O AV é uma solução recursiva ótima para o problema de estimação da seqüência de estados de um Processo de Markov de estados finitos e discretos no tempo, submetidos a um canal sem memória de ruído.

Um processo é dito Markoviano, se a probabilidade $P(x_{k+1}|x_0, x_1, \dots, x_k)$ de estar no estado x_{k+1} no instante $k + 1$, dado todos os estados anteriores até k , depender somente do estado imediatamente anterior x_k , ou seja[15]:

$$P(x_{k+1}|x_0, x_1, \dots, x_k) = P(x_{k+1}|x_k)$$

A Fig 5.1 ilustra o AV, considerando uma treliça com quatro estados e cinco transições. Neste exemplo o problema proposto foi o do caminho mais curto. A cada transição determinam-se os estados sobreviventes, que são aqueles associados ao custo mínimo. Cada transição corresponde a um passo no algoritmo, de modo que no fim dos k passos recursivos, o algoritmo determina o caminho mais curto do nó inicial até o final. O número máximo de sobreviventes que se pode ter é igual ao número de estados existentes.

Se o codificador de linha for representado por uma máquina sequencial síncrona, pode-se obter um algoritmo geral para o cálculo do diagrama de olho. Portanto, este conceito pode ser estendido para códigos, pois a lei de formação da treliça depende dos alfabetos e as IES a que é submetida a palavra, devido às interferências causadas pelas palavras anteriores e posteriores ao instante de tempo em questão.

5.3 Cálculo do Diagrama de olho

Para se calcular o diagrama de olho, basta adaptar o AV adequadamente. Esta adaptação foi realizada para o caso binário[3]. Nesta seção o faremos para o caso ternário. O programa consiste, basicamente, dos seguintes passos:

- P-1 Gerar palavras-código nT ;
- P-2 Escolher a forma de onda do pulso $p(t)$;
- P-3 Variar a fase de 1 a n , $\nu = 1, n$;
- P-4 Escolher $t_0 \in [-T/2, T/2]$;
- P-5 Para coeficiente $NCO = 0, 1$;
- P-6 Para $NI = -U, U$; $2U$ é o número total de interferentes
- P-7 Para $L = 1, NPC$; NPC é o número de palavras-código
- P-8 Se $NI = 0$, verificar se o bit correspondente à fase da palavra é igual ao NCO; se for, vá para P-8. Se não for, volte para P-6.
- P-9 Calcular a IES e associá-la ao custo de comutação de um estado inicial i para outro terminal j ;
- P-10 Calculadas as IES de todas as palavras do alfabeto, escolher o máximo ou o mínimo que chega em cada um dos estados terminais. Se $NCO = 0$, escolhe-se o máximo, caso contrário, escolhe-se o mínimo, obtendo-se assim, os sobreviventes;
- P-11 Acumular os custos para cada transição somando-os com os valores obtidos na transição anterior. O número de transições NT é dado por: $NT = 2NI/n$;
- P-12 Repetir os passos de P-7 a P-11 para as NT transições;

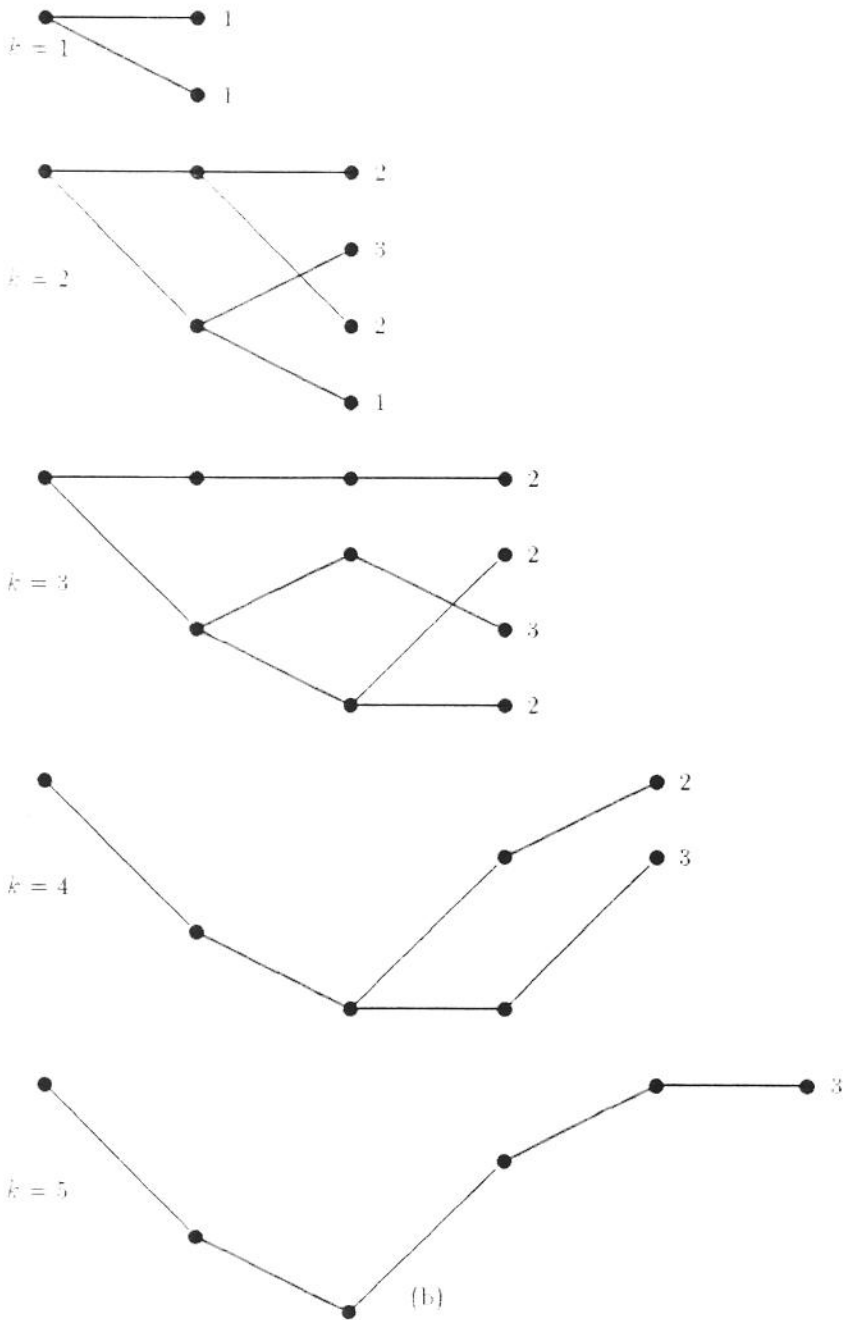
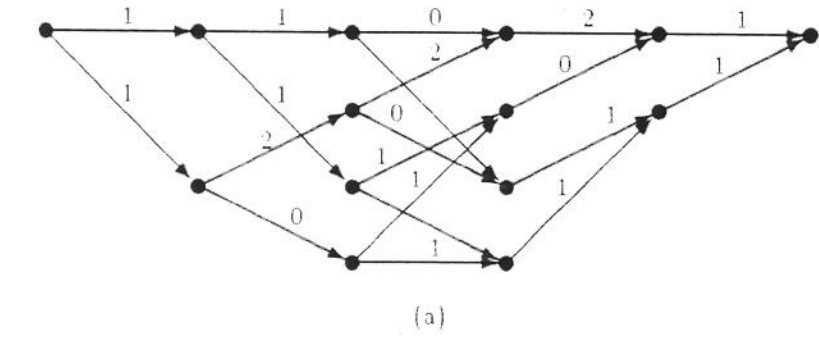


Figura 5.1: Treliça com 4 estados terminais, 5 transições e a determinação recursiva do caminho mais curto usando o AV

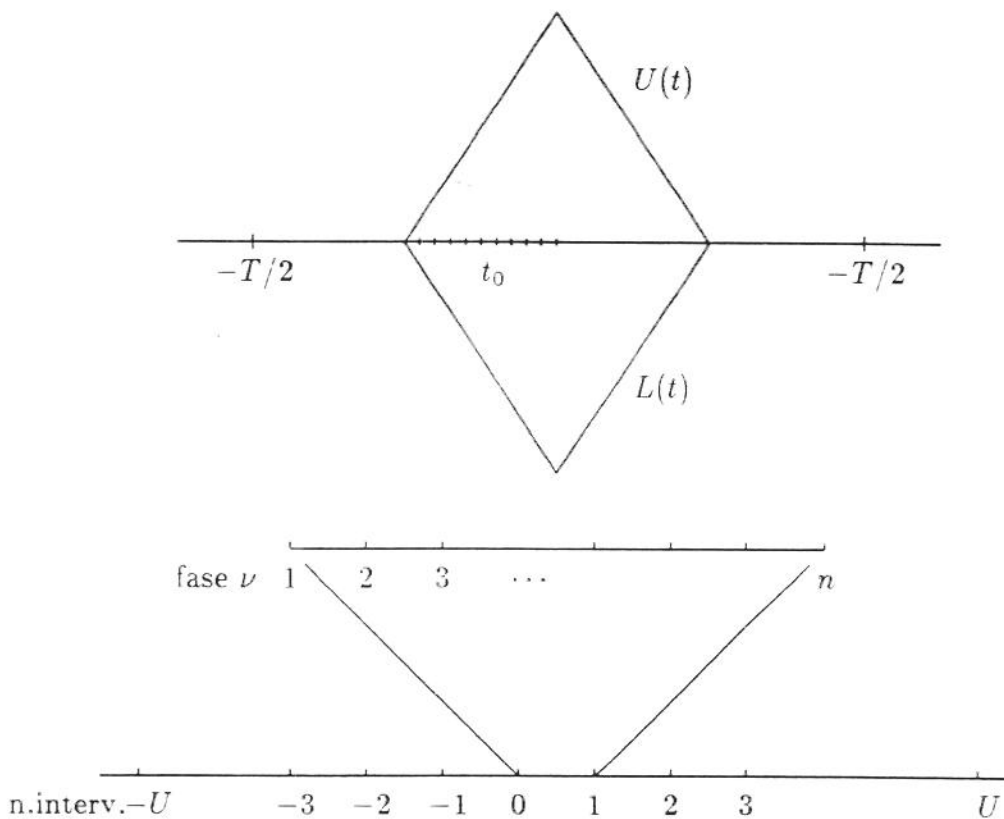


Figura 5.2: Construção do diagrama de olho

P-13 Ao final das NT transições, escolher os valores dos custos máximo e mínimo para t_0 , que correspondem ao piso e teto do olho naquele determinado instante;

P-14 Repetir os passos de P-5 a P-13 para os diversos valores de t_0 ;

P-15 Repetir os passos de P-4 a P-14 para obter os n olhos correspondentes a cada uma das fases:

Este procedimento é utilizado para calcular tanto o teto do olho, $U(t)$ como o piso, $L(t)$, dependendo do coeficiente que está sendo analisado. Se o coeficiente for 0, no passo no. 3. escolho o custo máximo para determinar o piso; e se for 1, escolho o custo mínimo para obter o teto.

A Fig.5.2 ilustra com mais detalhes a construção do olho.

Vale notar que o número de operações associadas de P-7 a P-11 é fixo e depende apenas do código, enquanto que a parte externa, ou seja, os passos anteriores a P-7 e posteriores a P-11, é repetida $2Un$, onde $2U$ é o número total de interferentes e n é o tamanho da palavra. Conclue-se, portanto, que a complexidade computacional cresce linearmente com

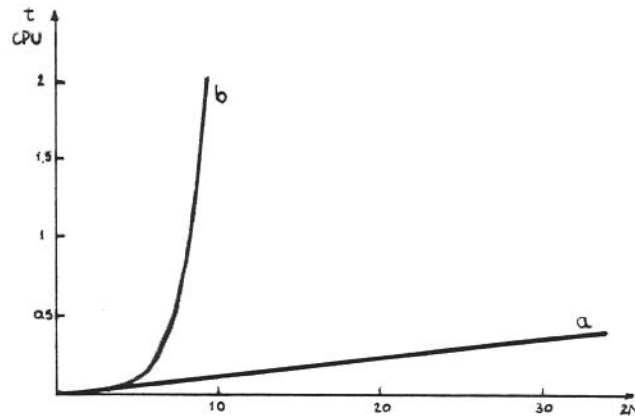


Figura 5.3: Tempo de CPU necessário para computar o diagrama de olho usando o AV (curva a) e o método exaustivo (curva b) versus o número de interferentes

o número de interferentes, ao contrário do que acontece com o método exaustivo, conforme pode-se observar na Fig.5.3[13].

A seguir será apresentado o programa codificado na linguagem *FORTRAN IV* e foi executado no sistema *VAX 11/750 da Digital Equipment Corporation*.

O primeiro programa foi implementado para códigos do tipo *5B4T*, cujo diagrama de estados é mostrado na Fig.4.4, onde os estados A, B, C e D são eliminados. O pulso considerado foi o de Nyquist, que é o de pior caso.

O segundo programa é para códigos do tipo *9B8T*, cujo diagrama de estados é mostrado na Fig.4.6. Outra vez, o pulso considerado foi o de Nyquist, o que significa que se outro pulso qualquer for transmitido, como por exemplo, os do tipo cosseno-levantado, o desempenho será melhor. Em outras palavras, o diagrama de olho obtido com qualquer pulso que tenha um excesso de faixa diferente de zero, será mais aberto do que o obtido com estes programas.

5B4T

```

FUNCTION P(T)      !PULSO DE NYQUIST
PI=4.*ATAN(1.)
IF(T)1,2,1
1  P=(SIN(PI*T))/(PI*T)
RETURN
2  P=1
RETURN
END

```

```

SUBROUTINE CALCTRE(CUSMAX, CUSMIN, MA, MI, VAUXM, VAUXMI)
DIMENSION CUSMAX(-2:2, -2:2), CUSMIN(-2:2, -2:2), MA(-2:2, -2:2),
1  MI(-2:2, -2:2), VAUXM(-2:2, -2:2), VAUXMI(-2:2, -2:2)
INTEGER*4 DP, DI
COMMON L, DP, DI, NC, SUM, NE
IF(VAUXM(DI, DP)+35)2, 1, 2
1  IF(VAUXMI(DI, DP).EQ.35)THEN
    CUSMAX(DI, DP)=SUM
    CUSMIN(DI, DP)=SUM
    MA(DI, DP)=L
    MI(DI, DP)=L
2  ELSE
    CUSMAX(DI, DP)=AMAX1(SUM, VAUXM(DI, DP))
    CUSMIN(DI, DP)=AMIN1(SUM, VAUXMI(DI, DP))
    IF (CUSMAX(DI, DP).NE. VAUXM(DI, DP))THEN
        MA(DI, DP)=L
    END IF
    IF (CUSMIN(DI, DP).NE. VAUXMI(DI, DP))THEN
        MI(DI, DP)=L
    END IF
END IF
VAUXM(DI, DP)=CUSMAX(DI, DP)
VAUXMI(DI, DP)=CUSMIN(DI, DP)
RETURN
END

```

```

SUBROUTINE ATRB(I1, I2, J1, J2, CM, CMIN, MAS, MIS, CSMAX, CSMIN, MA, MI)
DIMENSION CM(16, 16), CMIN(16, 16), MAS(16, 16), MIS(16, 16), NE(4, 4)
DIMENSION MA(-2:2, -2:2), MI(-2:2, -2:2), CSMAX(-2:2, -2:2), CSMIN(-2:2, -2:2)
INTEGER*4 DI, DP
COMMON L, DP, DI, NC, SUM, NE
DO 10 I=I1, I2

```

```

DO 10 J=J1,J2
CM(NE(I,J),NE(I+DI,J+DP))=CSMAX(DI,DP)
CMIN(NE(I,J),NE(I+DI,J+DP))=CSMIN(DI,DP)
MAS(NE(I,J),NE(I+DI,J+DP))=MA(DI,DP)
MIS(NE(I,J),NE(I+DI,J+DP))=MI(DI,DP)
RETURN
END

DIMENSION CMAX(16,16), CMIN(16,16), MAXS(16,16), MINS(16,16), MASQ(16),
1 MISQ(16), M(25), NE(4,4), CUSMAX(-2:2,-2:2), CUSMIN(-2:2,-2:2)
DIMENSION CMAT(16), CMIT(16), CMI(16), CMA(16), VAUXM(-2:2,-2:2),
1 VAUXMI(-2:2,-2:2), CLT(16), CST(16), VALMAX(4), VALMIN(4)
DIMENSION MA(-2:2,-2:2), MI(-2:2,-2:2)
INTEGER*4 C(81,4), COEF(3) A1(36,4), A2(36,4), A3(36,4), A4(36,4), DI, DP
COMMON L, DP, DI, NC, SUM, NE
COEF(1)=0
COEF(2)=1
COEF(3)=-1
DO 5 K=1,3 !GERACAO DAS PALAVRAS 4T
DO 5 L=1,27
L0=L+27*(K-1)
5 C(L0,1)=COEF(K)
NF=0
DO 15 MA1=1,3
DO 15 K=1,3
DO 10 L=1,9
L0=L+NF
10 C(L0,2)=COEF(K)
15 NF=NF+9
NF=0
DO 25 MA1=1,9
DO 25 K=1,3
DO 20 L=1,3
L0=L+NF
20 C(L0,3)=COEF(K)
25 NF=NF+3
DO 30 L=1,27
DO 30 K=1,3
L0=K+3*(L-1)
30 C(L0,4)=COEF(K)

K=0
DO 35 I=1,4
DO 35 J=1,4
K=K+1
35 NE(I,J)=K
NO=4 !NO E O NUMERO DE OLHOS

```

```

DO 190 NFA=1.NO      !NFA E A FASE
TI=NFA-1-0.110
TF=NFA-1+0.110
DO 185 T1=TI,TF,0.005
DO 175 NC=0.1      !NC E O COEFICIENTE DA PALAVRA CENTRAL
DO 40 I=1,16
CMAT(I)=0
40  CMIT(I)=0
DO 145 NI=-32,32,4  !NI E O NUMERO DE INTERVALOS VEZES 4
DO 45 I=1,16
DO 45 J=1,16
CMAX(I,J)=-50
45  CMIN(I,J)=50
L1=0
L2=0
L3=0
L4=0
DO 50 I=-2.2
DO 50 J=-2.2
VAUXM(I,J)=-35
50  VAUXMI(I,J)=35
DO 140 L=1.81      !CALCULO DOS CUSTOS ASSOCIADOS A CADA
SUM=0              !PALAVRA EM CADA TRANSICAO
DO 55 J=1.4
T0=-NI+1-J-T1
55  SUM=SUM+C(L,J)*P(T0)
IF(L.EQ.1)GO TO 140
IF(L.EQ.51)GO TO 140
IF(L.EQ.71)GO TO 140
IF(NI)65.60.65
60  IF(C(L,NFA).NE.NC)GO TO 500
65  DI=C(L,1)-C(L,3)
DP=C(L,2)+C(L,4)
CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
IF(DI.GT.0)THEN
  IF(DP.GT.0)THEN
    CALL ATRB(1,2,1,2, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
    L1=L1+1
    DO 70 J=1,4
      A1(L1,J)=C(L,J)
70  ELSE
    IF(DP.NE.0)THEN
      CALL ATRB(1,2,3,4, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN,MA,MI)
      L2=L2+1
      DO 75 J=1,4
        A2(L2,J)=C(L,J)
75  ELSE
    CALL ATRB(1,2,1,4, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN,MA,MI)
    L1=L1+1
    L2=L2+1
    DO 80 J=1,4

```

```

      A1(L1,J)=C(L,J)
      A2(L2,J)=C(L,J)
80      END IF
      END IF
      ELSE
      IF(DI.NE.0)THEN
      IF(DP.LT.0)THEN
      CALL ATRB(3,4,3,4, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN,MA,MI)
      L3=L3+1
      DO 85 J=1,4
85      A3(L3,J)=C(L,J)
      ELSE
      IF(DP.NE.0)THEN
      CALL ATRB(3,4,1,2, CMAX, CMIN,MAXS,MINS, CUSMAX, 1 CUSMIN,MA,MI)
      L4=L4+1
      DO 90 J=1,4
90      A4(L4,J)=C(L,J)
      ELSE
      CALL ATRB(3,4,1,4, CMAX, CMIN,MAXS,MINS, CUSMAX, 1 CUSMIN,MA,MI)
      L3=L3+1
      L4=L4+1
      DO 95 J=1,4
95      A3(L3,J)=C(L,J)
      A4(L4,J)=C(L,J)
      END IF
      END IF
      ELSE
      IF(DP.EQ.0)THEN
      IF(L.EQ.45.OR.L.EQ.53.OR.L.EQ.69.OR.L.EQ.77)
1      GO TO 105
      CALL ATRB(1,4,1,4, CMAX, CMIN,MAXS,MINS, CUSMAX,
1      CUSMIN,MA,MI)
      L1=L1+1
      L2=L2+1
      L3=L3+1
      L4=L4+1
      DO 100 J=1,4
      A1(L1,J)=C(L,J)
      A2(L2,J)=C(L,J)
      A3(L3,J)=C(L,J)
100      A4(L4,J)=C(L,J)
105      IF(L.EQ.45)THEN
      CALL ATRB(1,4,1,2, CMAX, CMIN,MAXS,MINS, CUSMAX,
1      CUSMIN,MA,MI)
      CALL ATRB(1,2,3,4, CMAX, CMIN,MAXS,MINS, CUSMAX,
1      CUSMIN,MA,MI)
      L1=L1+1
      L2=L2+1
      L4=L4+1
      DO 110 J=1,4
      A1(L1,J)=C(L,J)

```

```

110          A2(L2,J)=C(L,J)
            A4=(L4,J)

            ELSE
              IF(L.EQ.53)THEN
                CALL ATRB(1,2,1,4, CMAX, CMIN,MAXS.MINS,
1              CUSMAX, CUSMIN,MA,MI)
                CALL ATRB(3,4,3,4, CMAX, CMIN,MAXS.MINS,
1              CUSMAX, CUSMIN,MA,MI)
                L1=L1+1
                L2=L2+1
                L3=L3+1
                DO 115 J=1,4
                  A1(L1,J)=C(L,J)
                  A2(L2,J)=C(L,J)
115          A3(L3,J)=C(L,J)

              ELSE
                IF(L.EQ.69)THEN
                  CALL ATRB(1,4,1,2, CMAX, CMIN,
1              MAXS.MINS, CUSMAX, CUSMIN, MA,MI)
                  CALL ATRB(3,4,3,4, CMAX, CMIN,
1              MAXS.MINS, CUSMAX, CUSMIN, MA,MI)
                  L1=L1+1
                  L3=L3+1
                  L4=L4+1
                  DO 120 J=1,4
                    A1(L1,J)=C(L,J)
                    A3(L3,J)=C(L,J)
120          A4(L4,J)=C(L,J)

                ELSE
                  IF(L.EQ.77)THEN
                    CALL ATRB(1,4,3,4, CMAX, CMIN,
1              MAXS.MINS, CUSMAX, CUSMIN,MA,MI)
                    CALL ATRB(3,4,1,2, CMAX, CMIN,
1              MAXS.MINS, CUSMAX, CUSMIN,MA,MI)
                    L2=L2+1
                    L3=L3+1
                    L4=L4+1
                    DO 125 J=1,4
                      A2(L2,J)=C(L,J)
                      A3(L3,J)=C(L,J)
125          A4(L4,J)=C(L,J)

                  END IF
                END IF
              END IF
            END IF
          ELSE
            IF(DP.GT.0)THEN
              CALL ATRB(1,4,1,2, CMAX, CMIN,MAXS.MINS, CUSMAX,

```

```

1      CUSMIN,MA,MI)
      L1=L1+1
      L4=L4+1
      DO 130 J=1,4
130     A1(L1,J)=C(L,J)
      A4(L4,J)=C(L,J)
      ELSE
1      CALL ATRB(1,4,3,4, CMAX, CMIN,MAXS,MINS, CUSMAX,
      CUSMIN,MA,MI)
      L2=L2+1
      L3=L3+1
      DO 135 J=1,4
135     A2(L2,J)=C(L,J)
      A3(L3,J)=C(L,J)
      END IF
      END IF
      END IF
140  CONTINUE
      OPEN(UNIT=23,FILE='DADOS.OUT',STATUS=' UNKNOWN')
      IF(NC.EQ.1)THEN
      DO 145 IEF=1,16
      DO 145 IEI=1,16
      CST(IEI)=CMIT(IEI)-CMIN(IEI, IEF)
      IF(IEI.EQ.1)THEN
      CMI(IEF)=CST(IEI)
      MISQ(IEF)=MINS(IEI, IEF)
      ELSE
      CMI(IEF)=AMIN1(CST(IEI), CAMIN)
      IF(CMI(IEF).EQ.CST(IEI))THEN
      MISQ(IEF)=MINS(IEI, IEF)
      END IF
      END IF
      CAMIN=CMI(IEF)
145  CONTINUE
      ELSE
      DO 150 IEF=1,16
      DO 150 IEI=1,16
      CLT(IEI)=CMAT(IEI)+CMAX(IEI, IEF)
      IF(IEI.EQ.1)THEN
      CMA(IEF)=CLT(IEI)
      MASQ(IEF)=MAXS(IEI, IEF)
      ELSE
      CMA(IEF)=AMAX1(CLT(IEI), CAMAX)
      IF(CMA(IEF).EQ.CLT(IEI))THEN
      MASQ(IEF)=MAXS(IEI, IEF)
      END IF
      END IF
      CAMAX=CMA(IEF)
150  CONTINUE
      END IF
      DO 165 IEF=1,16

```

```
CMAT(IEF)=CMA(IEF)
CMIT(IEF)=CMI(IEF)
IF(NC.EQ.0)THEN
  WRITE(23,155)T1,NI,IEF,CMA(IEF),MASQ(IEF)
155   FORMAT(X,'T='F6.4,2X,'NI=' I3,2X,'EF=' I2,2X,
1   'MAX='F10.5,2X,'SOBMA=' I2)
ELSE
  WRITE(23,160)T1,NI,IEF,CMI(IEF),MISQ(IEF)
160   FORMAT(X,'T='F6.4,2X,'NI=' I3,2X,'EF=' I2,2X,
1   'MIN='F10.5,2X,'SOBMI=' I2)
END IF
165 CONTINUE
VMX1=AMAX1(CMAT(1),CMAT(2),CMAT(3),CMAT(4),CMAT(5),
1   CMAT(6),CMAT(7),CMAT(8))
VMX2=AMAX1(CMAT(9),CMAT(10),CMAT(11),CMAT(12),CMAT(13),
1   CMAT(14),CMAT(15),CMAT(16))
VMN1=AMIN1(CMIT(1),CMIT(2),CMIT(3),CMIT(4),CMIT(5),
1   CMIT(6),CMIT(7),CMIT(8))
VMN2=AMIN1(CMIT(9),CMIT(10),CMIT(11),CMIT(12),CMIT(13),
1   CMIT(14),CMIT(15),CMIT(16))
VMX=AMAX1(VMX1,VMX2)
VMN=AMIN1(VMN1,VMN2)
170 CONTINUE
175 CONTINUE
OPEN(UNIT=25,FILE='DOLH4.DAT',STATUS=' UNKNOWN')
WRITE(25,180)VMX,VMN
180   FORMAT(X,F10.5,3X,F10.5)
185 CONTINUE
190 CONTINUE
CLOSE(UNIT=23)
CLOSE(UNIT=25)
END
```

9B8T

```

FUNCTION P(T)      !PULSO DE NYQUIST
PI=4.*ATAN(1.)
IF(T)1,2,1
1  P=(SIN(PI*T))/(PI*T)
RETURN
2  P=1
RETURN
END

SUBROUTINE CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
DIMENSION CUSMAX(-2:2,-2:2), CUSMIN(-2:2,-2:2),MA(-2:2,-2:2),
1  MI(-2:2,-2:2),VAUXM(-2:2,-2:2),VAUXMI(-2:2,-2:2)
INTEGER*4 DP,DI
COMMON L,DP,DI,NC,SUM,NE
IF(VAUXM(DI,DP)+35)2,1,2
1  IF(VAUXMI(DI,DP).EQ.35)THEN
    CUSMAX(DI,DP)=SUM
    CUSMIN(DI,DP)=SUM
    MA(DI,DP)=L
    MI(DI,DP)=L
ELSE
2  CUSMAX(DI,DP)=AMAX1(SUM,VAUXM(DI,DP))
    CUSMIN(DI,DP)=AMIN1(SUM,VAUXMI(DI,DP))
    IF (CUSMAX(DI,DP).NE.VAUXM(DI,DP))THEN
        MA(DI,DP)=L
    END IF
    IF (CUSMIN(DI,DP).NE.VAUXMI(DI,DP))THEN
        MI(DI,DP)=L
    END IF
END IF
VAUXM(DI,DP)=CUSMAX(DI,DP)
VAUXMI(DI,DP)=CUSMIN(DI,DP)
RETURN
END

SUBROUTINE ATRB(I1, I2,J1,J2, CM, CMIN,MAS,MIS, CSMAX, CSMIN,MA,MI)
DIMENSION CM(9,9), CMIN(9,9),MAS(9,9),MIS(9,9),NE(3,3)
DIMENSION MA(-2:2,-2:2),MI(-2:2,-2:2), CSMAX(-2:2,-2:2), CSMIN(-2:2,-2:2)
INTEGER*4 DI,DP
COMMON L,DP,DI,NC,SUM,NE
DO 10 I=I1, I2
DO 10 J=J1,J2
CM(NE(I,J),NE(I+DI,J+DP))=CSMAX(DI,DP)
CMIN(NE(I,J),NE(I+DI,J+DP))=CSMIN(DI,DP)
MAS(NE(I,J),NE(I+DI,J+DP))=MA(DI,DP)
10 MIS(NE(I,J),NE(I+DI,J+DP))=MI(DI,DP)

```



```

RETURN
END

```

```

DIMENSION CMAX(9,9), CMIN(9,9),MAXS(9,9),MINS(9,9),MASQ(9),
1 MISQ(9),M(25),NE(3,3), CUSMAX(-2:2,-2:2), CUSMIN(-2:2,-2:2)
DIMENSION CMAT(9), CMIT(9), CMI(9), CMA(9),VAUXM(-2:2,-2:2),
1 VAUXMI(-2:2,-2:2), CLT(9), CST(9)
DIMENSION MA(-2:2,-2:2),MI(-2:2,-2:2),ND(2000,8),NDL(2000,8),
1 NBL(2000,8),NC(2000,8),NCL(2000,8),NA(2000,8)
INTEGER*4 C(81,4), COEF(3),DI,DP
COMMON L,DP,DI,NC,SUM,NE
COEF(1)=0
COEF(2)=1
COEF(3)=-1
DO 5 L=1,2187
DO 5 K=1,3 !GERACAO DAS PALAVRAS 8T
L0=K+3*(L-1)
5 C(L0,8)=COEF(K)
NF=0
DO 15 MA1=1,729
DO 15 K=1,3
DO 10 L=1,3
L0=L-NF
10 C(L0,7)=COEF(K)
15 NF=NF-3
NF=0
DO 25 MA1=1,243
DO 25 K=1,3
DO 20 L=1,9
L0=L-NF
20 C(L0,6)=COEF(K)
25 NF=NF+9
DO 35 MA1=1,81
DO 35 K=1,3
DO 30 L=1,27
L0=L-NF
30 C(L0,5)=COEF(K)
35 NF=NF-27
NF=0
DO 45 MA1=1,27
DO 45 K=1,3
DO 40 L=1,81
L0=L-NF
40 C(L0,4)=COEF(K)
45 NF=NF-81
NF=0
DO 55 MA1=1,9
DO 55 K=1,3
DO 50 L=1,243

```

```

L0=L+NF
50 C(L0.3)=COEF(K)
55 NF=NF+243
NF=0
DO 65 MA1=1,3
DO 65 K=1,3
DO 60 L=1,729
L0=L+NF
60 C(L0.2)=COEF(K)
65 NF=NF+729
DO 70 K=1,3
DO 70 L=1,2187
L0=L+2187*(K-1)
70 C(L0.1)=COEF(K)
K=0
DO 75 I=1,3
DO 75 J=1,3
K=K+1
75 NE(I,J)=K
LA=0
LB=0
LC=0
LD=0
DO 120 L=1,6561
LI=C(L,1)+C(L,3)
LP=C(L,2)+C(L,4)
LI1=LI+C(L,5)
LP1=LP-C(L,6)
DI=LI1+C(L,7)
DP=LP1-C(L,8)
IF(DI.GT.2.OR.DI.LT.0)GO TO 85
IF(DP.GT.2.OR.DP.LT.0)GO TO 85
IF(LI.LT.-1.OR.LP.LT.-1)GO TO 85
IF(DI.EQ.0.AND.DP.EQ.2)GO TO 85
IF(DI.EQ.2.AND.DP.EQ.0)GO TO 85
IF(LI.EQ.0.AND.LP.GE.2)GO TO 85
IF(LI.EQ.-1.AND.LP.GT.0)GO TO 85
IF(LI1.EQ.0.AND.LP1.GE.2)GO TO 85
IF(LI1.EQ.-1.AND.LP1.GT.0)GO TO 85
IF(LP.EQ.-1.AND.LI.GT.0)GO TO 85
IF(LP.EQ.0.AND.LI.EQ.2)GO TO 85
IF(LP1.EQ.-1.AND.LI1.GT.0)GO TO 85
IF(LP1.EQ.0.AND.LI1.EQ.2)GO TO 85
IF(LP1.EQ.1.AND.LI1.GT.2)GO TO 85
IF(LI1.EQ.1.AND.LP1.GT.2)GO TO 85
IF(C(L,1).EQ.0.AND.C(L,2).EQ.-1.AND.C(L,3).EQ.1)GO TO 85
IF(C(L,1).EQ.1.AND.C(L,2).EQ.0.AND.C(L,3).EQ.1)GO TO 85
IF(C(L,1).EQ.0.AND.C(L,2).EQ.1.AND.C(L,3).EQ.-1)GO TO 85
IF(LI.EQ.0.AND.LP.EQ.1.AND.C(L,5).EQ.-1)GO TO 85
IF(LI1.EQ.0.AND.LP1.EQ.1.AND.C(L,7).EQ.-1)GO TO 85
IF(LI.EQ.0.AND.LP.EQ.-1.AND.C(L,5).EQ.1)GO TO 85
IF(LI1.EQ.0.AND.LP1.EQ.-1.AND.C(L,7).EQ.1)GO TO 85

```

```

IF(LI.EQ.2.AND.LP.EQ.1.AND.C(L,5).EQ.1)GO TO 85
IF(LI.EQ.1.AND.LP.EQ.2.AND.C(L,5).EQ.-1)GO TO 85
IF(LI1.EQ.1.AND.LP1.EQ.2.AND.C(L,7).EQ.-1)GO TO 85
IF(LP.EQ.0.AND.LI.EQ.1.AND.C(L,5).EQ.1)GO TO 85
IF(LP1.EQ.0.AND.LI1.EQ.1.AND.C(L,7).EQ.1)GO TO 85
IF(LP.EQ.0.AND.LI.EQ.-1.AND.C(L,5).EQ.-1)GO TO 85
IF(C(L,1).EQ.1.AND.C(L,2).EQ.-1)GO TO 85
IF(C(L,1).EQ.-1.AND.C(L,2).EQ.1)GO TO 85
LD=LD+1
DO 80 J=1,8
IF(C(L,J).EQ.1) NDL(LD,J)=-1
IF(C(L,J).EQ.-1) NDL(LD,J)=1
IF(C(L,J).EQ.0) NDL(LD,J)=0
80 ND(LD,J)=C(L,J)
85 IF(DI.LT.0.OR.DP.GT.2)GO TO 95
IF(DP.LT.-1.OR.DP.GT.1)GO TO 95
IF(C(L,1).EQ.-1)GO TO 95
IF(LI.LT.-1.OR.LI1.LT.-1)GO TO 95
IF(DI.EQ.0.AND.DP.EQ.1)GO TO 95
IF(DI.EQ.2.AND.DP.EQ.-1)GO TO 95
IF(LI.EQ.0.AND.LP.GT.0)GO TO 95
IF(LI1.EQ.0.AND.LP1.GT.0)GO TO 95
IF(LI.EQ.2.AND.LP.LT.0)GO TO 95
IF(LI1.EQ.2.AND.LP1.LT.0)GO TO 95
IF(LI.EQ.-1.AND.LP.GE.0)GO TO 95
IF(LI1.EQ.-1.AND.LP1.GE.0)GO TO 95
IF(LI.EQ.1.AND.LP.LT.-1)GO TO 95
IF(LI1.EQ.1.AND.LP1.LT.-1)GO TO 95
IF(LI.EQ.1.AND.LP.GT.1)GO TO 95
IF(LI1.EQ.1.AND.LP1.GT.1)GO TO 95
IF(C(L,1).EQ.1.AND.C(L,2).EQ.-1.AND.C(L,3).EQ.1)GO TO 95
IF(C(L,1).EQ.1.AND.C(L,2).EQ.1.AND.C(L,3).EQ.-1)GO TO 95
IF(C(L,1).EQ.0.AND.C(L,2).EQ.0.AND.C(L,3).EQ.-1)GO TO 95
IF(C(L,1).EQ.0.AND.C(L,2).EQ.1)GO TO 95
IF(LI.EQ.2.AND.LP.EQ.2.AND.C(L,5).EQ.-1)GO TO 95
IF(LI1.EQ.2.AND.LP1.EQ.2.AND.C(L,7).EQ.-1)GO TO 95
IF(LI.EQ.0.AND.LP.EQ.0.AND.C(L,5).EQ.-1)GO TO 95
IF(LI1.EQ.0.AND.LP1.EQ.0.AND.C(L,7).EQ.-1)GO TO 95
IF(LI.EQ.0.AND.LP.EQ.-2.AND.C(L,5).EQ.1)GO TO 95
IF(LI1.EQ.0.AND.LP1.EQ.-2.AND.C(L,7).EQ.1)GO TO 95
IF(LI.EQ.2.AND.LP.EQ.0.AND.C(L,5).EQ.1)GO TO 95
IF(LI1.EQ.2.AND.LP1.EQ.0.AND.C(L,7).EQ.1)GO TO 95
IF(LI.EQ.1.AND.LP.EQ.-1.AND.C(L,5).EQ.1)GO TO 95
IF(LI1.EQ.1.AND.LP1.EQ.-1.AND.C(L,7).EQ.1)GO TO 95
IF(LI.EQ.1.AND.LP.EQ.1.AND.C(L,5).EQ.-1)GO TO 95
IF(LI1.EQ.1.AND.LP1.EQ.1.AND.C(L,7).EQ.-1)GO TO 95
LC=LC+1
DO 90 J=1,8
IF(C(L,J).EQ.1) NCL(LC,J)=-1
IF(C(L,J).EQ.-1) NCL(LC,J)=1

```

```

IF(C(L,J).EQ.0) NCL(LC,J)=0
90 NC(LC,J)=C(L,J)
95 IF(DP.GT.0.OR.DP.LT.-2)GO TO 105
IF(DI.LT.-1.OR.DI.GT.1)GO TO 105
IF(LI.LT.-1.OR.C(L,1).EQ.-1)GO TO 105
IF(DI.EQ.-1.AND.DP.EQ.0)GO TO 105
IF(DI.EQ.1.AND.DP.EQ.-2)GO TO 105
IF(LI.EQ.-1.AND.LP.GT.-1)GO TO 105
IF(LI1.EQ.-1.AND.LP1.GT.-1)GO TO 105
IF(LI.EQ.0.AND.LP.GT.0)GO TO 105
IF(LI1.EQ.0.AND.LP1.GT.0)GO TO 105
IF(LI1.EQ.0.AND.LP1.LT.-2)GO TO 105
IF(LI.EQ.1.AND.LP.LT.-1)GO TO 105
IF(LI1.EQ.1.AND.LP1.LT.-1)GO TO 105
IF(LP.EQ.-1.AND.LI.GT.1)GO TO 105
IF(LP1.EQ.-1.AND.LI1.GT.1)GO TO 105
IF(LP1.EQ.-1.AND.LI.LT.-1)GO TO 105
IF(C(L,1).EQ.0.AND.C(L,2).EQ.1)GO TO 105
IF(C(L,1).EQ.1.AND.C(L,2).EQ.-1.AND.C(L,3).EQ.1)GO TO 105
IF(C(L,1).EQ.1.AND.C(L,2).EQ.1.AND.C(L,3).EQ.-1)GO TO 105
IF(C(L,1).EQ.0.AND.C(L,2).EQ.0.AND.C(L,3).EQ.-1)GO TO 105
IF(LI.EQ.0.AND.LP.EQ.0.AND.C(L,5).EQ.-1)GO TO 105
IF(LI1.EQ.0.AND.LP1.EQ.0.AND.C(L,7).EQ.-1)GO TO 105
IF(LI.EQ.0.AND.LP.EQ.-2.AND.C(L,5).EQ.1)GO TO 105
IF(LI1.EQ.0.AND.LP1.EQ.-2.AND.C(L,7).EQ.1)GO TO 105
IF(LI.EQ.-1.AND.LP.EQ.-1.AND.C(L,5).EQ.-1)GO TO 105
IF(LI1.EQ.-1.AND.LP1.EQ.-3.AND.C(L,7).EQ.1)GO TO 105
IF(LP.EQ.-1.AND.LI.EQ.1.AND.C(L,5).EQ.1)GO TO 105
IF(LI.EQ.1.AND.LP.EQ.-1.AND.C(L,5).EQ.1)GO TO 105
IF(LI1.EQ.1.AND.LP1.EQ.1.AND.C(L,5).EQ.-1)GO TO 105
IF(LI1.EQ.1.AND.LP1.EQ.1.AND.C(L,7).EQ.-1)GO TO 105
LB=LB+1
DO 100 J=1,8
IF(C(L,J).EQ.1) NBL(LB,J)=-1
IF(C(L,J).EQ.-1) NBL(LB,J)=1
IF(C(L,J).EQ.0) NBL(LB,J)=0
100 NB(LB,J)=C(L,J)
105 IF(DI.LT.-1.OR.DI.GT.1)GO TO 115
IF(DP.LT.-1.OR.DP.GT.1)GO TO 115
IF(DI.EQ.1.AND.DP.EQ.-1)GO TO 115
IF(DI.EQ.-1.AND.DP.EQ.1)GO TO 115
IF(LP.EQ.0.AND.LI.GT.1)GO TO 115
IF(LP1.EQ.0.AND.LI1.GT.1)GO TO 115
IF(LP.EQ.0.AND.LI.LT.-1)GO TO 115
IF(LP1.EQ.0.AND.LI1.LT.-1)GO TO 115
IF(LP.EQ.1.AND.LI.LT.0)GO TO 115
IF(LP1.EQ.1.AND.LI1.LT.0)GO TO 115
IF(LP.EQ.-1.AND.LI.GT.0)GO TO 115
IF(LP1.EQ.-1.AND.LI1.GT.0)GO TO 115
IF(LI.EQ.0.AND.LP.GT.1)GO TO 115
IF(LI1.EQ.0.AND.LP1.GT.1)GO TO 115
IF(LI.EQ.0.AND.LP.LT.-1)GO TO 115

```

```

IF(LI1.EQ.0.AND.LP1.LT. -1)GO TO 115
IF(LI.EQ.1.AND.LP.LT.0)GO TO 115
IF(LI1.EQ.1.AND.LP1.LT.0)GO TO 115
IF(LI.EQ. -1.AND.LP.GT.0)GO TO 115
IF(LI1.EQ. -1.AND.LP1.GT.0)GO TO 115
IF(C(L,1).EQ.1.AND.C(L,2).EQ. -1)GO TO 115
IF(C(L,1).EQ. -1.AND.C(L,2).EQ.1)GO TO 115
IF(C(L,1).EQ.1.AND.C(L,2).EQ.0.AND.C(L,3).EQ.1)GO TO 115
IF(C(L,1).EQ.0.AND.C(L,2).EQ.1.AND.C(L,3).EQ. -1)GO TO 115
IF(C(L,1).EQ.0.AND.C(L,2).EQ. -1.AND.C(L,3).EQ.1)GO TO 115
IF(LP.EQ.0.AND.LI.EQ.1.AND.C(L,5).EQ.1)GO TO 115
IF(LP1.EQ.0.AND.LI1.EQ.1.AND.C(L,7).EQ.1)GO TO 115
IF(LP.EQ.0.AND.LI.EQ. -1.AND.C(L,5).EQ. -1)GO TO 115
IF(LP1.EQ.0.AND.LI1.EQ. -1.AND.C(L,7).EQ. -1)GO TO 115
IF(LP.EQ. -1.AND.LI.EQ.0.AND.C(L,5).EQ.1)GO TO 115
IF(LP1.EQ. -1.AND.LI1.EQ.0.AND.C(L,7).EQ.1)GO TO 115
IF(LI.EQ.0.AND.LP.EQ.1.AND.C(L,5).EQ. -1)GO TO 115
IF(LI1.EQ.0.AND.LP1.EQ.1.AND.C(L,7).EQ. -1)GO TO 115
IF(LI.EQ.1.AND.LP.EQ.2.AND.C(L,5).EQ. -1)GO TO 115
IF(LI1.EQ.1.AND.LP1.EQ.2.AND.C(L,7).EQ. -1)GO TO 115
IF(LI.EQ. -1.AND.LP.EQ. -2.AND.C(L,5).EQ.1)GO TO 115
IF(LI1.EQ. -1.AND.LP1.EQ. -2.AND.C(L,7).EQ.1)GO TO 115
LA=LA+1
DO 110 J=1,8
110   NA(LA,J)=C(L,J)
115   CONTINUE
      NO=8           !NO E O NUMERO DE OLHOS
      DO 350 NFA=1,NO   !NFA E A FASE
      TI=NFA-1-0.20
      TF=NFA-1+0.20
      DO 345 T1=TI,TF,0.005
      DO 335 NC=0,1     !NC E O COEFICIENTE DA PALAVRA CENTRAL
      DO 120 I=1,9
      CMAT(I)=0
120   CMIT(I)=0
      DO 330 NI=-32,32,8   !NI E O NUMERO DE INTERVALOS VEZES 8
      DO 125 I=1,9
      DO 125 J=1,9
      CMAX(I,J)=-50
125   CMIN(I,J)=50
      DO 130 I=-2,2
      DO 130 J=-2,2
      VAUXM(I,J)=-35
130   VAUXMI(I,J)=35
      DO 150 L=2,LD       !CALCULO DOS CUSTOS ASSOCIADOS A CADA
      SUM=0               !PALAVRA EM CADA TRANSICAO
      DO 135 J=1,8
      T0=-NI+1-J+T1
135   SUM=SUM+ND(L,J)*P(T0)
      IF(NI)145.140.145
140   IF(ND(L,NFA).EQ.NCO)THEN
145   DI=ND(L,1)+ND(L,3)+ND(L,5)+ND(L,7)

```

```

        DP=ND(L,2)+ND(L,4)+ND(L,6)+ND(L,8)
        CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
        CALL ATRB(1,1,1,1, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
END IF
150  CONTINUE
      DO 155 I=-2,2
      DO 155 J=-2,2
      VAUXM(I,J)=-35
155  VAUXMI(I,J)=35
      DO 175 L=2,LD      !CALCULO DOS CUSTOS ASSOCIADOS A CADA
      SUM=0             !PALAVRA EM CADA TRANSICAO
      DO 160 J=1,8
      T0=-NI+1-J+T1
160  SUM=SUM+NDL(L,J)*P(T0)
      IF(NI)170,165,170
165  IF(NDL(L,NFA).EQ.NCO)THEN
170    DI=NDL(L,1)+NDL(L,3)+NDL(L,5)+NDL(L,7)
      DP=NDL(L,2)+NDL(L,4)+NDL(L,6)+NDL(L,8)
      CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
      CALL ATRB(3,3,3,3, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
END IF
175  CONTINUE
      DO 180 I=-2,2
      DO 180 J=-2,2
      VAUXM(I,J)=-35
180  VAUXMI(I,J)=35
      DO 200 L=2,LC      !CALCULO DOS CUSTOS ASSOCIADOS A CADA
      SUM=0             !PALAVRA EM CADA TRANSICAO
      DO 185 J=1,8
      T0=-NI+1-J+T1
185  SUM=SUM+NC(L,J)*P(T0)
      IF(NI)195,190,195
190  IF(NC(L,NFA).EQ.NCO)THEN
195    DI=NC(L,1)+NC(L,3)+NC(L,5)+NC(L,7)
      DP=NC(L,2)+NC(L,4)+NC(L,6)+NC(L,8)
      CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
      CALL ATRB(1,1,2,2, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
END IF
200  CONTINUE
      DO 205 I=-2,2
      DO 205 J=-2,2
      VAUXM(I,J)=-35
205  VAUXMI(I,J)=35
      DO 225 L=2,LC      !CALCULO DOS CUSTOS ASSOCIADOS A CADA
      SUM=0             !PALAVRA EM CADA TRANSICAO
      DO 210 J=1,8
      T0=-NI+1-J+T1
210  SUM=SUM+NCL(L,J)*P(T0)
      IF(NI)220,215,220
215  IF(NCL(L,NFA).EQ.NCO)THEN
220    DI=NCL(L,1)+NCL(L,3)+NCL(L,5)+NCL(L,7)
      DP=NCL(L,2)+NCL(L,4)+NCL(L,6)+NCL(L,8)

```

```

        CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
        CALL ATRB(3,3,2,2, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
END IF
225 CONTINUE
DO 230 I=-2,2
DO 230 J=-2,2
VAUXM(I,J)=-35
230 VAUXMI(I,J)=35
DO 250 L=2,LB      !CALCULO DOS CUSTOS ASSOCIADOS A CADA
SUM=0             !PALAVRA EM CADA TRANSICAO
DO 235 J=1,8
T0=-NI+1-J+T1
235 SUM=SUM+NB(L,J)*P(T0)
IF(NI)245,240,245
240 IF(NB(L,NFA).EQ.NCO)THEN
245   DI=NB(L,1)+NB(L,3)+NB(L,5)+NB(L,7)
   DP=NB(L,2)+NB(L,4)+NB(L,6)+NB(L,8)
   CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
   CALL ATRB(2,2,3,3, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
END IF
250 CONTINUE
DO 255 I=-2,2
DO 255 J=-2,2
VAUXM(I,J)=-35
255 VAUXMI(I,J)=35
DO 275 L=2,LB      !CALCULO DOS CUSTOS ASSOCIADOS A CADA
SUM=0             !PALAVRA EM CADA TRANSICAO
DO 260 J=1,8
T0=-NI+1-J+T1
260 SUM=SUM+NBL(L,J)*P(T0)
IF(NI)270,265,270
265 IF(NBL(L,NFA).EQ.NCO)THEN
270   DI=NBL(L,1)+NBL(L,3)+NBL(L,5)+NBL(L,7)
   DP=NBL(L,2)+NBL(L,4)+NBL(L,6)+NBL(L,8)
   CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)
   CALL ATRB(2,2,1,1, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
END IF
275 CONTINUE
DO 280 I=-2,2
DO 280 J=-2,2
VAUXM(I,J)=-35
280 VAUXMI(I,J)=35
DO 300 L=2,LA      !CALCULO DOS CUSTOS ASSOCIADOS A CADA
SUM=0             !PALAVRA EM CADA TRANSICAO
DO 285 J=1,8
T0=-NI+1-J+T1
285 SUM=SUM+NA(L,J)*P(T0)
IF(NI)295,290,295
290 IF(NA(L,NFA).EQ.NCO)THEN
295   DI=NA(L,1)+NA(L,3)+NA(L,5)+NA(L,7)
   DP=NA(L,2)+NA(L,4)+NA(L,6)+NA(L,8)
   CALL CALCTRE(CUSMAX, CUSMIN,MA,MI,VAUXM,VAUXMI)

```

```

        CALL ATRB(2,2,2,2, CMAX, CMIN,MAXS,MINS, CUSMAX, CUSMIN, MA,MI)
    END IF
300  CONTINUE
    OPEN(UNIT=23,FILE='D8.OUT',STATUS=' UNKNOWN')
    IF(NC.EQ.1)THEN
        DO 305 IEF=1,9
        DO 305 IEI=1,9
        IF(IEI.EQ.3.OR.IEI.EQ.7)GO TO 305
        IF(IEF.EQ.3.OR.IEF.EQ.7)GO TO 305
        CST(IEI)=CMIT(IEI)+CMIN(IEI, IEF)
        IF(IEI.EQ.1)THEN
            CMI(IEF)=CST(IEI)
            MISQ(IEF)=MINS(IEI, IEF)
            ITMI(IEF)=1
        ELSE
            CMI(IEF)=AMIN1(CST(IEI), CAMIN)
            IF(CMI(IEF).EQ.CST(IEI))THEN
                MISQ(IEF)=MINS(IEI, IEF)
                ITMI(IEF)=IEI
            END IF
        END IF
        CAMIN=CMI(IEF)
305  CONTINUE
    ELSE
        DO 310 IEF=1,9
        DO 310 IEI=1,9
        IF(IEI.EQ.3.OR.IEI.EQ.7)GO TO 310
        IF(IEF.EQ.3.OR.IEF.EQ.7)GO TO 310
        CLT(IEI)=CMAT(IEI)+CMAX(IEI, IEF)
        IF(IEI.EQ.1)THEN
            CMA(IEF)=CLT(IEI)
            MASQ(IEF)=MAXS(IEI, IEF)
            ITMA(IEF)=1
        ELSE
            CMA(IEF)=AMAX1(CLT(IEI), CAMAX)
            IF(CMA(IEF).EQ.CLT(IEI))THEN
                MASQ(IEF)=MAXS(IEI, IEF)
                ITMA(IEF)=IEI
            END IF
        END IF
        CAMAX=CMA(IEF)
310  CONTINUE
    END IF
    DO 325 IEF=1,9
    CMAT(IEF)=CMA(IEF)
    CMIT(IEF)=CMI(IEF)
    IF(NC.EQ.0)THEN
        WRITE(23,315)T1,NI, IEF, ITMA(IEF), CMA(IEF),MASQ(IEF)
315  FORMAT(X,'T='F6.4,2X,'NI=' I3,2X,' EF=' I2,2X,'EI=' I2,2X,'MAX='F10.5,2X,
1      'SOBMA=' I2)
    ELSE

```



```
WRITE(23,320)T1,NI, IEF, ITMI(IEF), CMI(IEF),MISQ(IEF)
320   FORMAT(X,'T='F6.4,2X,'NI=' I3,2X,' EF=' I2,2X,'EI=' I2,2X,'MIN='F10.5,2X,
1     'SOBMI=' I2)
END IF
325   CONTINUE
VMX1=AMAX1(CMAT(1), CMAT(2), CMAT(3), CMAT(4), CMAT(5))
VMX2=AMAX1(CMAT(6), CMAT(7), CMAT(8), CMAT(9))
VMN1=AMIN1(CMIT(1), CMIT(2), CMIT(3), CMIT(4), CMIT(5))
VMN2=AMIN1(CMIT(6), CMIT(7), CMIT(8), CMIT(9))
VMX=AMAX1(VMX1,VMX2)
VMN=AMIN1(VMN1,VMN2)
330   CONTINUE
335   CONTINUE
OPEN(UNIT=25,FILE='ÓLH8SRES.DAT',STATUS=' UNKNOWN')
WRITE(25,340)VMX,VMN
340   FORMAT(X,F10.5,3X,F10.5)
345   CONTINUE
350   CONTINUE
CLOSE(UNIT=23)
CLOSE(UNIT=25)
END
```

5.4 Resultados

A Fig.5.4 mostra os olhos obtidos para a classe de códigos 5B4T. Para o cálculo foram considerados os seguintes dados:

- 34 palavras para cada um dos quatro alfabetos. A palavra 0000 foi excluída de todos os alfabetos e as palavras --++, -+ +-, ++-- e +- -+ foram excluídas dos alfabetos correspondentes ao primeiro, segundo, terceiro e quarto quadrantes, respectivamente;
- 32 interferentes de cada lado ($U = 32$);
- intervalo entre as amostras $\Delta t = 0,005$;

A largura do olho é aproximadamente $0,2T$, onde T é o período de sinalização. Foi possível, também, determinar a seqüência de pior caso.

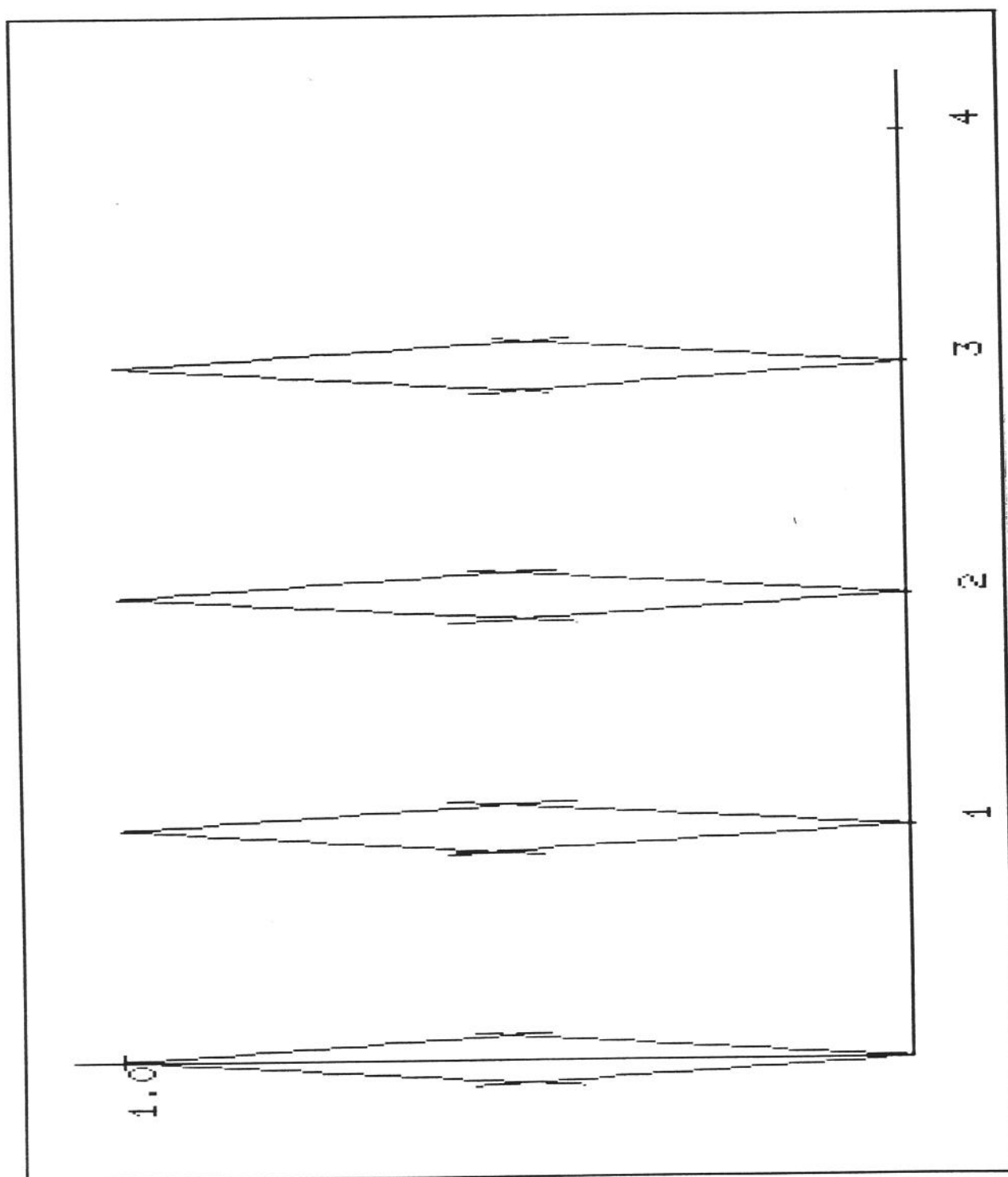


Figura 5.4. 5B4T. $\Delta t = 0.005$

- além da palavra 0000 foram eliminadas de todos os alfabeto as palavras $-+++$ e $+---$;
- 32 interferentes de cada lado;
- intervalo entre as amostras $\Delta t = 0,002$;

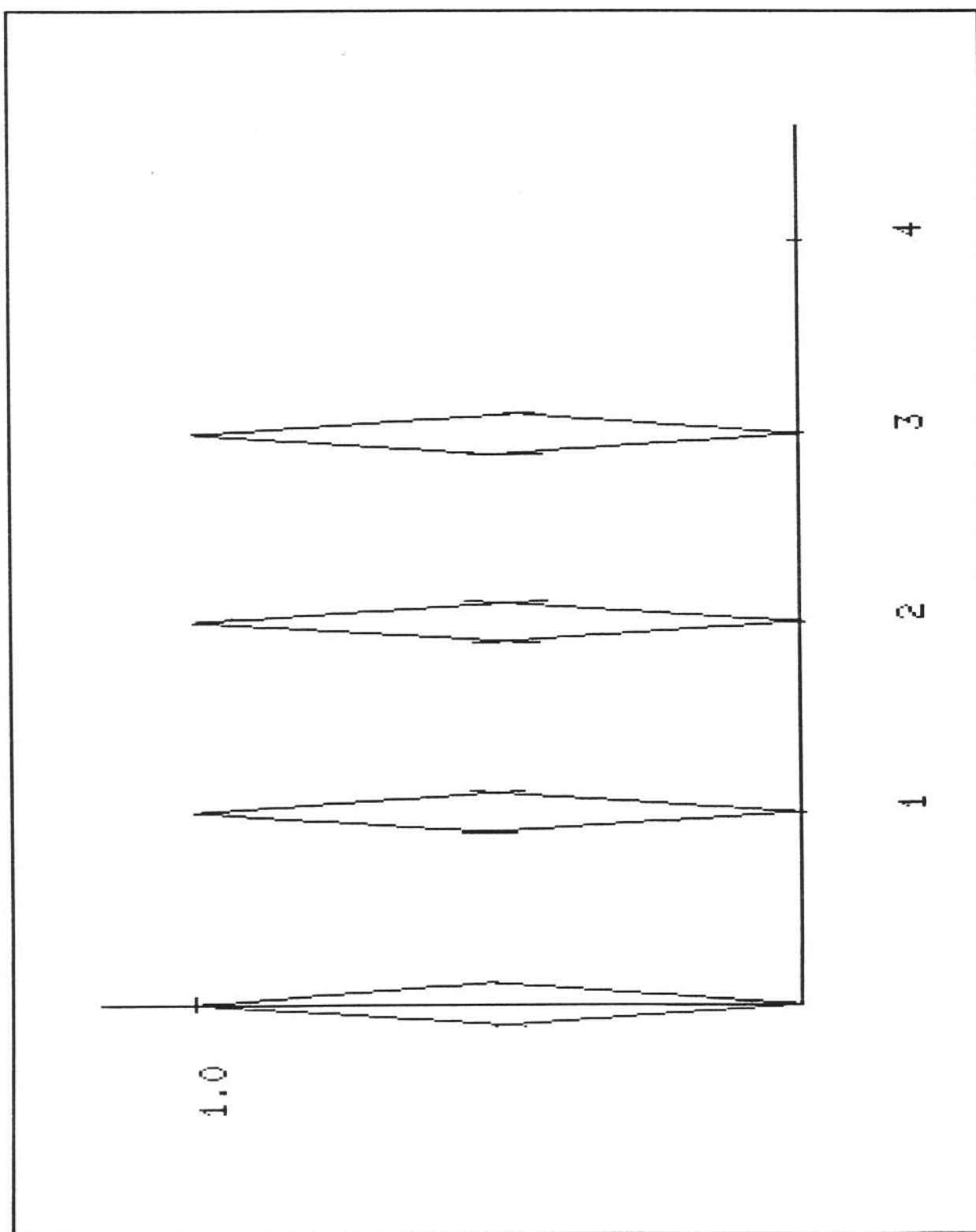


Figura 5.5: 5B4T, excluindo-se $-+++$ e $+---$ e $\Delta t = 0,005$

A Fig.5.6 mostra o diagrama de olho correspondente à classe $9B8T$, onde considerou-se os seguintes dados:

- foram consideradas todas as palavras de cada um dos alfabetos, eliminando-se apenas a palavra 00000000;
- 16 interferentes de cada lado ($U = 16$);
- intervalo entre amostras $\Delta t = 0,005$;

A largura do olho para o $9B8T$ é aproximadamente $0,36T$. Este valor é levemente superior ao do DBM , confirmando assim, a discussão do Capítulo 4. Deve-se notar que este valor é o de pior caso, pois existem pelo menos 76 palavras que podem ser excluídas de cada alfabeto. Se estas palavras forem escolhidas convenientemente pode-se obter uma abertura maior que a obtida neste caso.

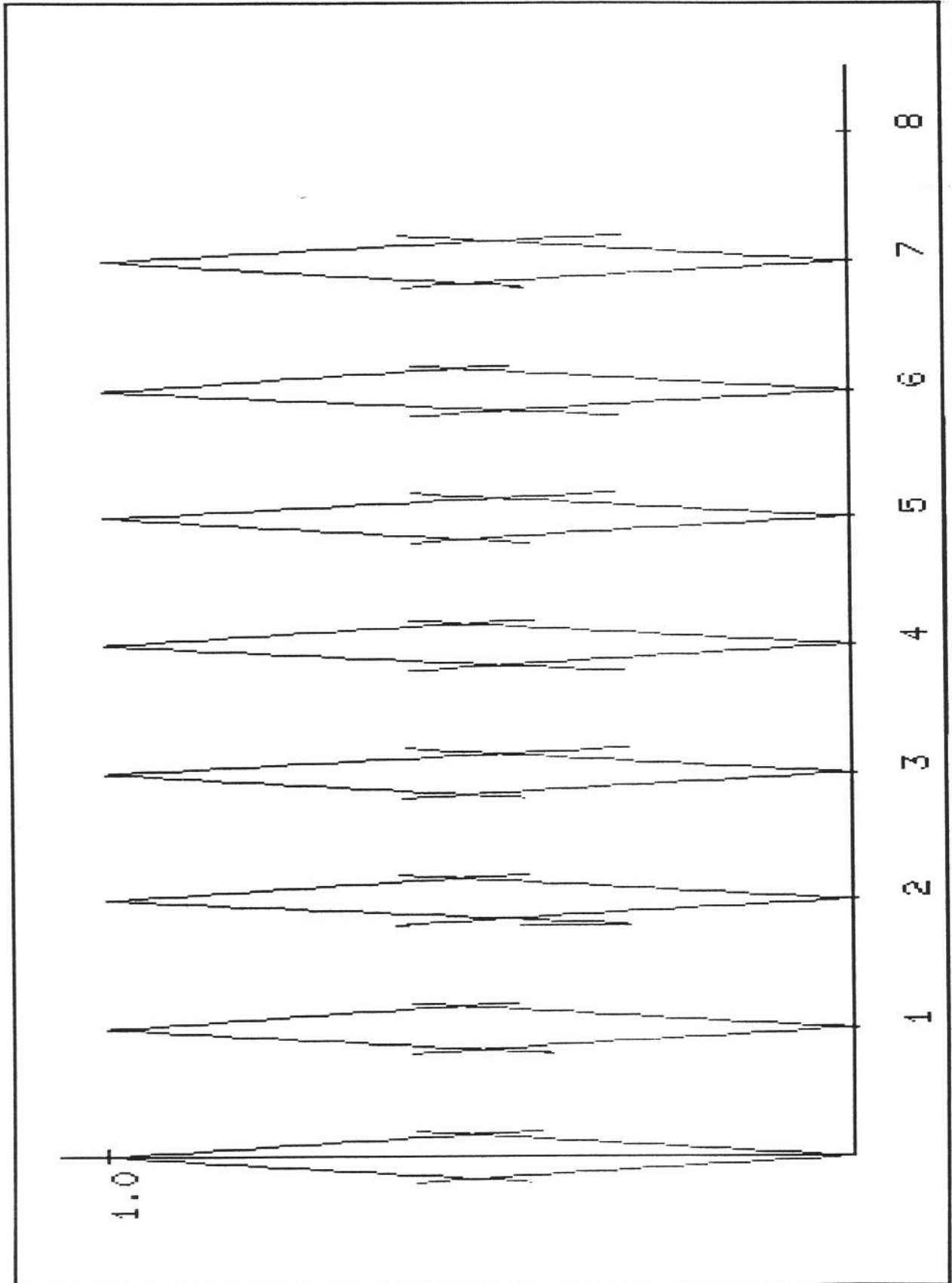


Figura 5.6: 9B8T, $\Delta t = 0.005$

Para comparar melhor os olhos obtidos, estes foram traçados no mesmo gráfico que é mostrado na Fig.5.7

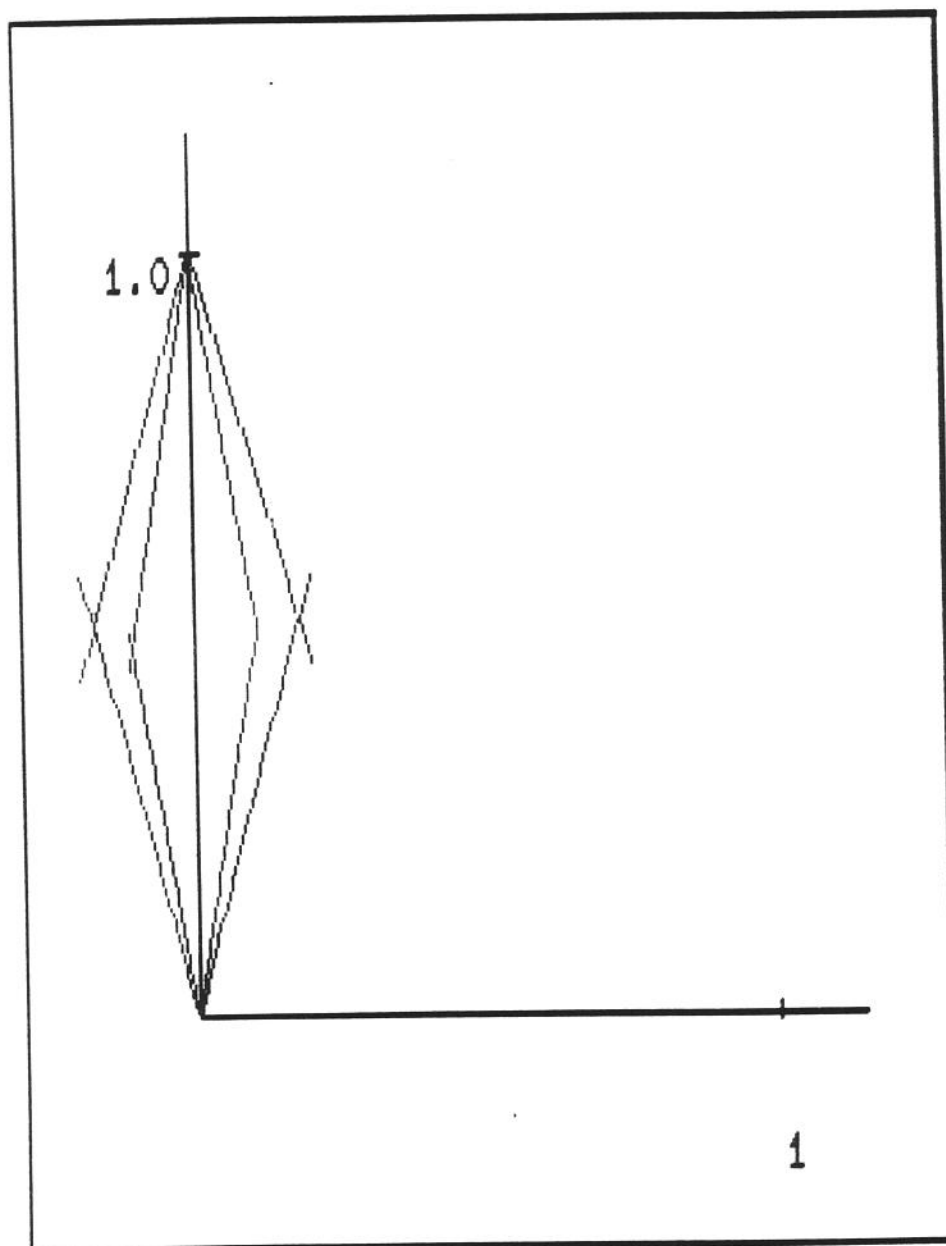


Figura 5.7: Comparação dos olhos do 5B4T e do 9B8T

APÊNDICE 1

Dadas duas matrizes, o que este programa faz, basicamente, é a multiplicação delas. A ordem está limitada em 20. A sua aplicação neste trabalho é gerar as matrizes de transição para um determinado número de bits, a partir das matrizes de transição dos bits pares e ímpares. Verifica, também, o número de palavras disponíveis para se implementar um código que reduza a taxa de símbolos, a partir de uma dada configuração.

```

DIMENSION A(20,20),B(20,20),C(20,20),D(20,20),MI(20,20)
DIMENSION MP(20,20),C1(20,20),D1(20,20)
TYPE *,'ORDEM'
INTEGER*4 NEST
ACCEPT *.N
TYPE *,'MAT. TRANS. IMP.'
DO 10 I=1,N
10 ACCEPT 15,(MI(I,J),J=1,N)
15 FORMAT(20I2)
TYPE *,'MAT. TRANS. PAR.'
DO 20 I=1,N
20 ACCEPT 15,(MP(I,J),J=1,N)
DO 110 MD=1,2
DO 25 I=1,N
DO 25 J=1,N
25 A(I,J)=MI(I,J)
B(I,J)=MP(I,J)
L=2
30 MINI=1000000
LM=L+MD
NEST=2**LM
DO 35 I=1,N
DO 35 J=1,N
35 C(I,J)=0
DO 50 I=1,N
D(I)=0
DO 45 J=1,N
DO 40 K=1,N
40 C(I,J)=C(I,J)+A(I,K)*B(I,J)
45 D(I)=D(I)+C(I,J)
IF(D(I).LE.MINI) MINI=D(I)
50 CONTINUE
IF(MINI.LE.NEST)THEN
IF(L.EQ.2)THEN
DO 55 I=1,N
DO 55 J=1,N
55 A(I,J)=C(I,J)
B(I,J)=C(I,J)
ELSE
DO 60 I=1,N

```



```
DO 60 J=1,N
60  A(I,J)=C(I,J)
    TYPE *,L
    DO 75 I=1,N
75  TYPE 80,(C(I,J),J=1,N)
80  FORMAT(X,20F5.0)
    END IF
    L=L+2
    GO TO 30
    ELSE
    MINIM=10000000
    L1=L-1
    DO 85 I=1,N
    DO 85 J=1,N
85  C1(I,J)=0
    DO 100 I=1,N
    D1(I)=0
    DO 95 J=1,N
    DO 90 K=1,N
90  C1(I,J)=C1(I,J)+A(I,K)*MI(K,J)
95  D1(I)=D1(I)+C1(I,J)
    IF(MINIM.GE.D1(I)) MINIM=D1(I)
100 CONTINUE
    L2=L1+MD
    IF(MINIM.GE.2**L2)THEN
105  TYPE 105, L2, L1
    FORMAT(X, I2, 'B', I2, 'T')
    ELSE
    TYPE 105,LM,L
    END IF
    END IF
110 CONTINUE
    END
```

APÊNDICE 2

Dada uma matriz de transição, este programa calcula o auto-valor dentro de um intervalo limitado por VI e VF.

```

DIMENSION A(20,20)
INTEGER*4 Z(20,20)
TYPE *, ÓRDEM'
ACCEPT *,N
TYPE *, 'V.INICIAL,PASSO,V.FINAL'
ACCEPT *,VI,PA,VF
TYPE *, 'MATRIZ DE TRANS.'
DO 5 I=1,N
TYPE *, 'LINHA'
ACCEPT 10,(Z(I,J),J=1,N)
5   FORMAT(20I1)
10  DO 20 I=1,N
15  DO 20 J=1,N
20  A(I,J)=Z(I,J)
    DT=1
25  DO 25 I=1,N
    A(I, I)=Z(I, I)-VI
    K=1
30  IF(A(K,K).NE.0)THEN
    M=K
    C=A(K,K)
    DT=DT*C
35  DO 35 J=K,N
    A(K,J)=A(K,J)/C
    DO 45 I=1,N
    IF(I.NE.K)THEN
    D=A(I,K)
40  DO 40 J=K,N
    A(I,J)=A(I,J)-A(K,J)*D
    END IF
45  CONTINUE
    K=K+1
    IF(K.LE.N)THEN
    GO TO 30
    ELSE
    GO TO 55
    END IF
    END IF
    M=M+1
    DO 50 J=1,N
    B=A(K,J)

```

```
A(K,J)=A(M,J)
50 A(M,J)=B
GO TO 30
55 TYPE 60,VI
60 FORMAT(X,'AUTOVALOR',F6.4)
TYPE 65.DT
65 FORMAT(X,'DETERM.',F20.6)
VI=VI+PA
DO 70 I=1.20000
70 CONTINUE
IF(VLLT.VF)GO TO 15
END
```

APÊNDICE 3

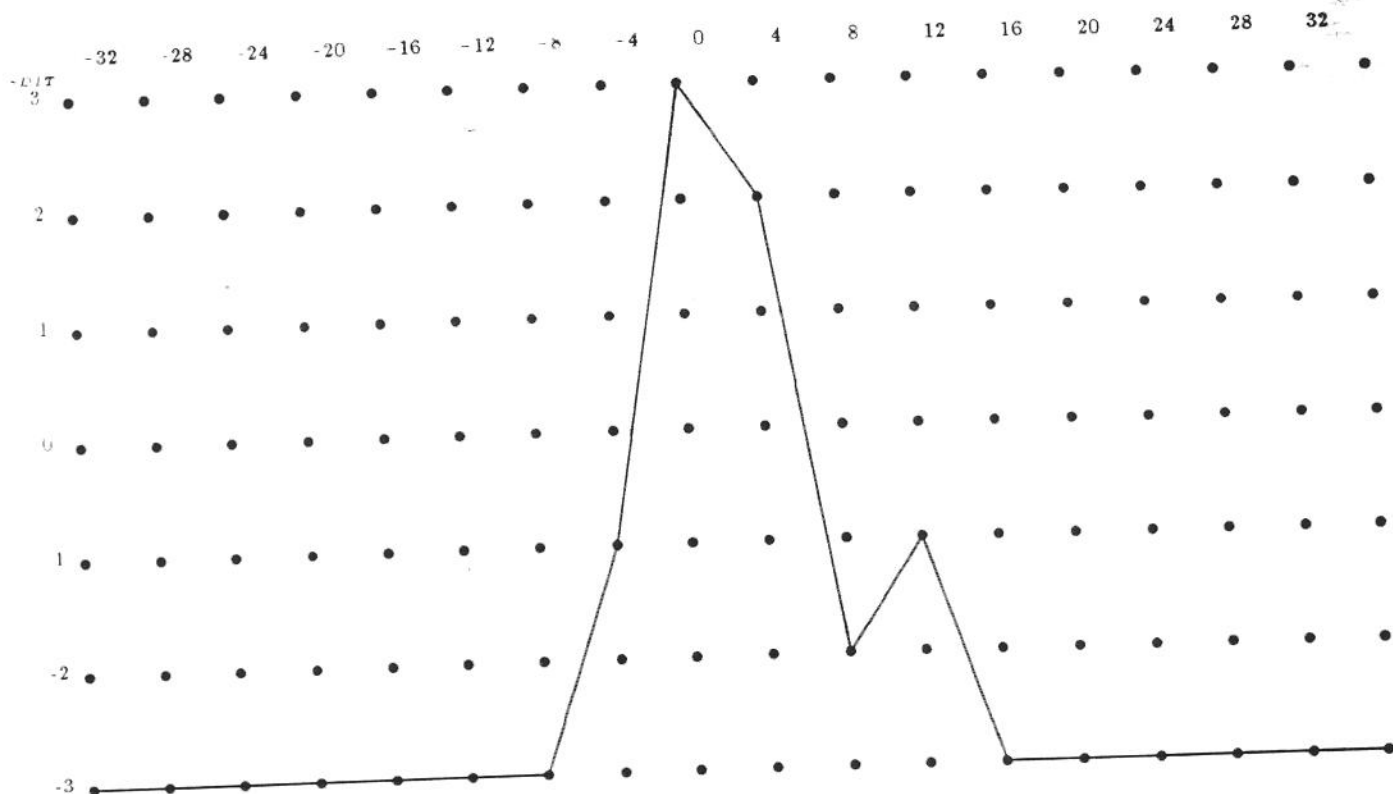


Figura .1: 5B4T com bit central igual a zero

Baseado na Fig.1 a seqüência de pior caso , quando o pulso analisado é 0 é dada por:

$$0-0+ / 0-0+ / 0-0+ / 0-0+ / 0-0+ / 0-0+ / 00-+ / -+-+ / -0+- / +$$

$$-+ - / 0- -+ / + -++ / +0-0 / +0-0 / +0-0 / +0-0 / +0-0$$

E quando o pulso central é 1 a seqüência de pior caso é a seguinte:

$$0+0- / 0+0- / 0+0- / 0+0- / 0+0- / 0+0- / 00+- / +-+- / +-+$$

$$+ / -+ - - / -0+- / -+++ / 0++- / -+-- / -0+0 / -0+0 / -0+0$$

conforme a Fig.2.

A seqüência que gera o diagrama de olho de pior caso para o 9B8T, considerando-se 16 interferentes de cada lado é mostrada nas Figs.3 e 4.

As seqüências de pior caso para o 9B8T são:

para bit=0

+ + 0 + 0 - 0 - / - 00000 + - / 0 + - 00000 / 0000000 - / - 0000000

para bit=1

- - 0 - 0 + 0 + / + 00000 - + / + - 000000 / 0000000 + / + 00000 - +

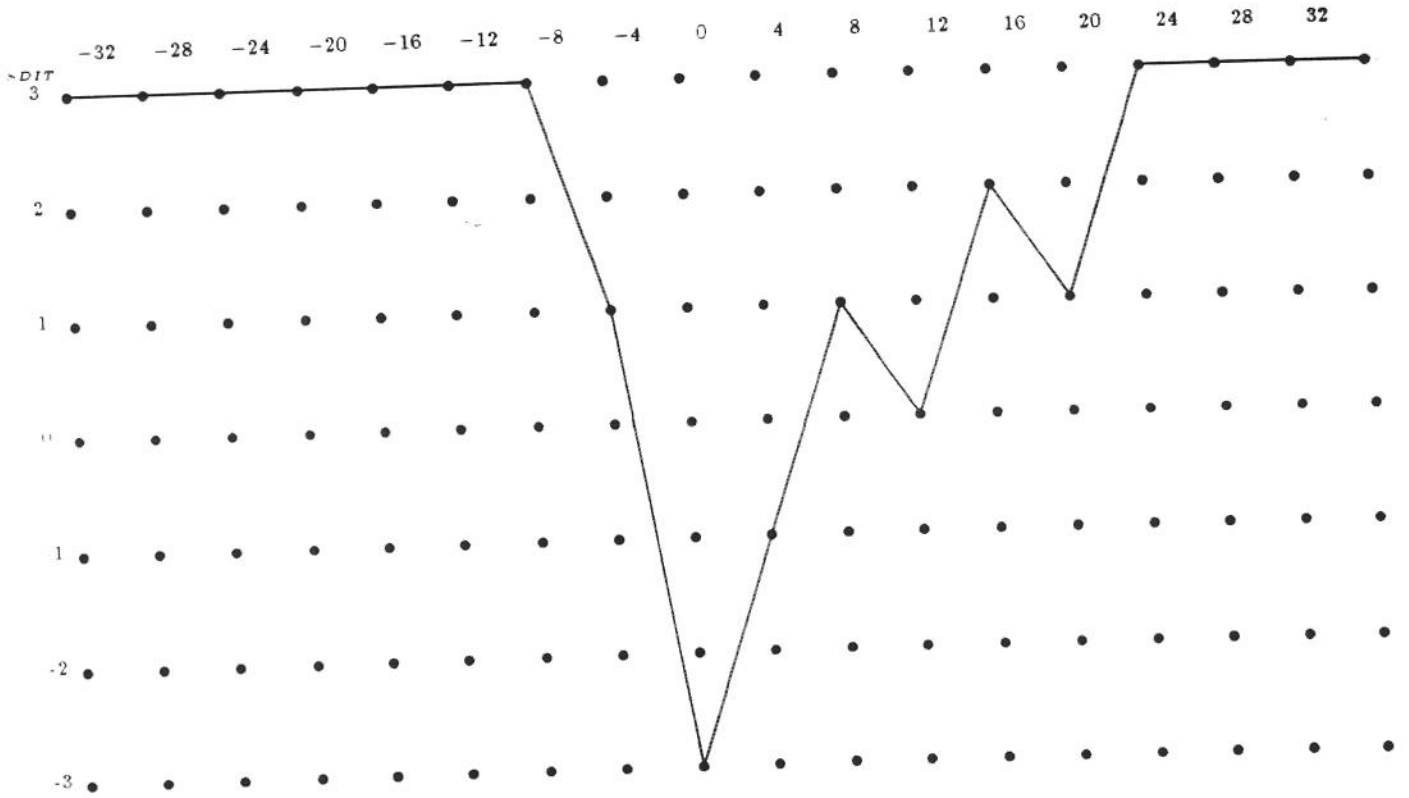


Figura .2: 5B4T com bit central igual a um

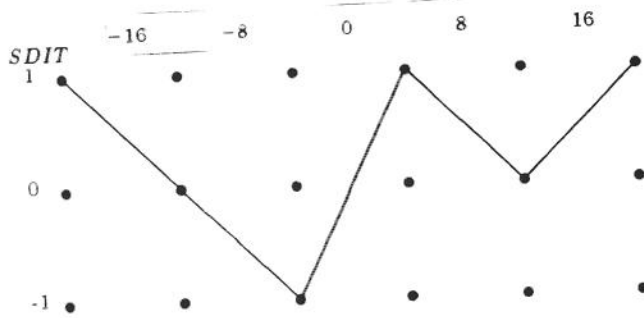


Figura .3: 9B8T com bit central igual a zero

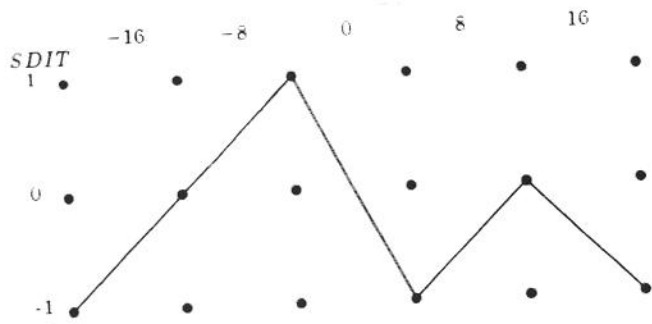


Figura .4: 9B8T com bit central igual a um

BIBLIOGRAFIA

- [1] Nyquist, H., *Certain Topics in Telegraph Transmission Theory*, Trans. AIEE, vol.47, Apr.1928, pp. 617-644.
- [2] Kabal, P.; Pasupathy, S., *Partial-Response Signaling*, IEEE Trans. on Communications, vol. COM-23, Sep.1975, pp. 921-934.
- [3] Mota de Alcântara, H.J.; Waldman, H., *Códigos de Bloco Binários de Faixa Mínima para Comunicações via Fibras Ópticas*, FEC/UNICAMP, Jul. 1988.
- [4] Dingjam A.J.M., *Philips optical fibre transmission systems:I. System evaluation*, Philips Telecommunication Review, vol.40, Jul.1982.
- [5] Romano, J.M., *Códigos Balanceados para Transmissão Digital por Fibras Ópticas*, FEC/UNICAMP, Jul.1984.
- [6] Kim, D.Y., *A Study on the Line Codes of Minimum Bandwidth*, Ph.D. Dissertation, D.E.E. Korea Advanced Institute of Science and Technology, 1983.
- [7] Waldman, H.; Mota de Alcântara, H.J.; Iramina, M.M., *Códigos de Bloco de Faixa Mínima*, 6o. Simpósio Brasileiro de Telecomunicações, Set.1988, pp. 149-155.
- [8] Forney Jr, G.D., *The Viterbi Algorithm*, Proc. IEEE, vol.61, Mar. 1973, pp. 268-278.
- [9] Bennett, W.R.; Davey, J.R., *Data Transmission*, McGraw-Hill, New York, 1965.
- [10] Lucky, R.W.; Salz, J.; Weldon, E.J.J., *Principles of Data Communications*, McGraw-Hill, 1965.
- [11] Lender, A., *The Duobinary Technique for High Speed Data Transmission*, IEEE Trans. on Commun. Electron., vol.82, May 1963, pp. 214-218.
- [12] Monti, C.M.; Pupolin, S.G., *Fast Computer Calculation of the Eye Diagram*, Alta Frequenza, vol.XLVIII, Nov.1979, pp.458-466.
- [13] Omura, J.K., *On the Viterbi Decoding Algorithm*, IEEE Trans. on Inform. Theory, vol.IT-15, Ja.1969, pp.177-179.
- [14] Gallager, R.G., *Information Theory and Reliable Communication*, John Wiley and Sons, 1968, pp.63-66.