

SUBCONJUNTOS EFICIENTES DE CÓDIGOS DE ALFABETO REDUNDANTE  
PARA A PROCURA NÃO - EXAUSTIVA DE CÓDIGOS ÓTIMOS

IVONETE MARKMAN

CPqD/TELEBRÁS

C.P. 1579

13075 - Campinas, SP

WALTER C. BORELLI

UNICAMP/FEE

C.P. 6101

13083 - Campinas, SP

RESUMO

Os códigos de alfabeto redundante (AR), propostos por Ungerboeck [1], bem como, generalizações dos mesmos [2,3,4], melhoram a imunidade ao ruído, introduzindo a redundância pelo aumento do alfabeto de modulação, em relação ao da transmissão não-codificada. Por serem códigos de treliça, os códigos AR ótimos costumam ser obtidos através de procura, sobre o universo destes códigos [1,5]. Em [5], foram propostos algoritmos de procura exaustiva e não-exaustiva de tais códigos, sob o critério da máxima distância mínima Euclidiana. A procura não-exaustiva se realiza sobre subconjuntos especiais, onde é alta a probabilidade de encontrar bons códigos. Este trabalho apresenta vários destes subconjuntos e a distância mínima de seus códigos ótimos. O conhecimento prévio destas distâncias simplifica a procura de códigos AR ótimos para qualquer sistema de modulação / codificação AR utilizado.

1. INTRODUÇÃO

Considerando a codificação de canal e a modulação como uma entidade, Ungerboeck [1] propôs uma classe de códigos aqui referidos como códigos de alfabeto redundante (AR). Esta classe de códigos caracteriza-se por adicionar a redundância através do aumento do alfabeto de modulação, em relação ao da transmissão não-codificada. Estes códigos têm sido aplicados a sistemas de modulação de uma ou mais dimensões [1,2], desta forma, melhorando a imunidade ao ruído, sem comprometimento da taxa de informação e da largura de faixa requerida. Uma generalização dos códigos AR [3,4] permitiu aplicar a técnica de codificação a sistemas multidimensionais, associados à modulação em frequência. Neste caso, foram obtidos melhores desempenhos, ao preço de maiores larguras de faixa requeridas.

1.1. A Codificação AR

Em sua estrutura, os códigos AR podem ser visualizados como codificador convolucional binário de taxa  $R = K/N$ , seguido de um mapeamento dos bytes codificados num conjunto com  $2^N$  símbolos de canal multiníveis (fig. 1).

Este mapeamento, conhecido como mapeamento por partição de conjunto [1], é obtido por sucessivas partições da constelação de sinais, em subconjuntos com crescentes distâncias mínimas Euclidianas ( $\Delta_0 < \Delta_1 < \dots < \Delta_{N-1}$ ), entre os sinais que os compõem (fig. 2).

1.2. A Distância Mínima Euclidiana

Através do processo de codificação convolucional, a cada seqüência de informação  $x$  (semi-infinita):

$$x = \{\underline{x}^1, \underline{x}^2, \underline{x}^3, \dots\} \quad (1)$$

composta de bytes de entrada  $\underline{x}^t = [X_1 X_2 \dots X_K]$ , o codificador faz corresponder uma seqüência transmitida, ou palavra-código  $y$  (semi-infinita):

$$y = \{\underline{y}^1, \underline{y}^2, \underline{y}^3, \dots\} \quad (2)$$

com bytes de saída  $\underline{y}^t = [Y_1 Y_2 \dots Y_N]$ , para  $t=1,2,3,\dots$ , e para um código de taxa  $R = K/N$ .

A distância mínima Euclidiana entre as seqüências de sinais modulados  $a(y)$  é definida [1] como:

$$d_{\text{MIN}} = \min d_E [a(y), a(y')] \quad (3)$$

onde  $d_E [a(y), a(y')] = \left\{ \sum_{t=1}^{L_y} d_E^2 [a(\underline{y}^t), a(\underline{y}'^t)] \right\}^{1/2}$  é a distância Euclidiana entre seqüências  $a(y)$  e  $a(y')$  de sinais modulados (símbolos de canal); e cada parcela dentro da somatória representa o quadrado da distância Euclidiana (em termos vetoriais) entre os símbolos de canal correspondentes aos bytes  $\underline{y}^t$  e  $\underline{y}'^t$ , respectivamente, para quaisquer duas seqüências codificadas  $y$  e  $y'$  não-coincidentes, de comprimento  $L_y = L_{y'}$ .

Para os códigos AR, Ungerboeck [1] também derivou um limitante inferior para a distância mínima Euclidiana, de forma que:

$$d_{\text{MIN}}^2 \geq \Delta_{\text{MIN}}^2 = \min_{y \neq 0} \sum_{t=0}^{L_y} \Delta_q^2(\underline{y}^t) \quad (4)$$

onde  $q(\underline{y}^t)$  é o número de zeros seguidos, a partir do bit de mais baixo índice, no byte codificado  $\underline{y}^t$  do caminho não-nulo  $y$  de comprimento  $L_y$ ; e  $\Delta_q(\underline{y}^t)$  é o peso AR do byte  $\underline{y}^t$  [4].

1.3. O Desempenho dos Códigos AR

Na decodificação por máxima verossimilhança dos códigos AR (Algoritmo de Viterbi [1]), para altos valores de relação sinal-ruído (SNR), a probabilidade de evento de erro aproxima-se assintoticamente do limitante inferior [1]:

$$P_e \geq N(d_{\text{MIN}}) \cdot Q(d_{\text{MIN}}/2\sigma) \quad (5)$$

onde  $N(d_{\text{MIN}})$  é o número (médio) de eventos de erro com distância mínima Euclidiana;  $\sigma^2$  é a variância do ruído; e  $Q(z)$  é a função de probabilidade de erro Gaus-

siana; 
$$Q(z) = (1/\sqrt{2\pi}) \cdot \int_z^{\infty} \exp(-t^2/2) dt \quad (6)$$

Das eqs. 5 e 6, conclui-se que devem-se procurar códigos AR com a máxima distância mínima Euclidiana entre as seqüências de sinais modulados, no sentido de minimizar a probabilidade de erro.

Assim, como uma medida do desempenho dos códigos AR, um ganho de codificação assintótico pode ser definido [1] como:

$$GCA = 20 \cdot \log_{10}(d_{MIN}/d_{REF}) \text{ dB} \quad (7)$$

onde  $d_{REF}$  é a distância mínima entre os sinais de um sistema não-codificado, utilizado como referência.

#### 1.4. Representações dos Códigos AR

Pela facilidade de tratamento dos números binários, costuma-se representar o código AR pelo código convolucional que o caracteriza, a menos do mapeamento.

Como se trata de uma máquina seqüencial de estados finitos, um codificador convolucional pode ser representado tal qual o exemplo da fig. 3. Na fig. 3a, está a representação esquemática (em termos de registradores de deslocamento, ou elementos de memória) de um codificador de taxa  $R = 2/3$  (duas entradas e três saídas), uma entrada codificada com dois elementos de memória e uma entrada não-codificada (sem elementos de memória). Na fig. 3b, está a representação em treliça para o mesmo codificador, onde a presença de transições paralelas entre os estados (nós) se deve à existência de uma entrada não-codificada [4].

Neste trabalho, serão adotados codificadores como os da fig. 3, cujas representações esquemáticas não possuem qualquer tipo de realimentação, e cujas entradas podem ser ou não codificadas. Além disso, para um codificador de taxa  $R = K/N$  e sendo  $m_i$  o número de elementos de memória na entrada  $i$  ( $i = 1, 2, \dots, K$ ), então:  $m_1 \geq m_2 \geq \dots \geq m_K \geq 0$ . Também, será chamado de comprimento total de memória de um codificador, o valor  $CTM = \sum_{i=1}^K m_i$ .

Assim, define-se uma classe simples de codificadores  $\{G_{KN}; m_1, m_2, \dots, m_K\}$  como o conjunto de todos os codificadores de taxa  $R = K/N$  e estrutura de memória:  $m_1, m_2, \dots, m_K$ . E define-se um conjunto total de classes  $\{G_{KN}; M\}$  como o conjunto de todas as classes simples, tais que  $CTM = M$  [5]. No exemplo da fig. 3, o codificador pertence à classe simples  $\{G_{23}; m_1 = 2, m_2 = 0\}$  e ao conjunto total de classes  $\{G_{23}; M = 2\}$ .

Na fig. 4, estão dispostas todas as possibilidades de estrutura de memória, para codificadores de  $K$  entradas ( $K \leq 5$ ) e  $CTM$  elementos de memória ( $CTM \leq 5$ ). Fixando-se um valor de  $N$  ( $N > K$ ), cada possibilidade representará uma classe simples de codificadores. Assim, para  $K = 2$  e  $CTM = 2$ , há duas classes simples de codificadores:  $\{G_{2N}; m_1 = 2, m_2 = 0\}$  e  $\{G_{2N}; m_1 = 1, m_2 = 1\}$ . Estas duas classes simples definem o conjunto total de classes  $\{G_{2N}; M = 2\}$ .

Ainda uma terceira representação está exemplificada na fig. 3c: a dos vetores coeficientes. Um vetor coeficiente

$VC(j)$  ( $j = 1, 2, \dots, N$ ) consiste na representação da saída  $j$  do codificador, em função das conexões desta saída às entradas e aos elementos de memória do mesmo. Para tanto, a partir da representação esquemática, devem-se numerar os elementos de memória e as entradas, no sentido de cima para baixo, e da direita para a esquerda, em ordem crescente, como na fig. 3a. Em seguida, para cada vetor coeficiente  $VC(j)$ , preenche-se um '1' (ou '0'), numa dada posição do vetor, se a saída  $j$  tiver (ou não) conexão com esta posição.

Pode-se verificar que, para codificadores de uma mesma classe simples  $\{G_{KN}; m_1, m_2, \dots, m_K\}$ , a sua representação por vetores coeficientes possuirá o mesmo número de vetores ( $N$ ) e as mesmas posições por vetor (em número de  $K + M$ ). Os codificadores diferirão entre si pela designação de '0' ou '1', para estas posições [4].

## 2. A PROCURA DE CÓDIGOS AR ÓTIMOS

Como possui um código convolucional embutido em sua estrutura, o código AR é um código de treliça, não apresentando, portanto, um tratamento algébrico, como ocorre para códigos de bloco. Desta forma, a determinação de códigos AR ótimos costuma ser realizada através de procura, sobre o universo destes códigos [1,5].

Um critério de otimalidade apropriado, como já visto (eqs. 5 e 6), é a distância mínima Euclidiana do código. No entanto, os códigos AR não apresentam as propriedades de linearidade dos códigos convolucionais [6]. Assim é que, para efeito de procura, utiliza-se o critério de máximo  $\Delta_{MIN}$  (eq.4), que é um limitante inferior muito próximo, quando não, igual a  $d_{MIN}$  (eq.3) [1,5]. A adoção deste limitante inferior e, portanto, das distâncias entre os sinais como função dos  $\Delta_i$  da partição de conjunto, permite trabalhar com o código como se ele tivesse essas propriedades de linearidade. Isto facilita enormemente o trabalho computacional envolvido.

A definição de código AR ótimo está associada a uma dada taxa  $R = K/N$  e dado valor de  $CTM$ , já que estes fatores definem a complexidade de decodificação [4]. Desta forma, será definido como ótimo, o código que possuir o máximo  $\Delta_{MIN}$ , entre todos os códigos cujos codificadores pertençam ao conjunto total de classes  $\{G_{KN}; M\}$ , onde  $CTM = M$ . Isto implica a realização da procura para todas as classes simples de um conjunto total de classes (fig.4) e comparação final dos melhores códigos de cada uma destas classes. Também será definido como codificador AR ótimo, aquele que gera o código AR ótimo.

### 2.1. A Procura Exaustiva

Na procura exaustiva de códigos ótimos, são analisados todos os codificadores de uma classe simples, para todas as classes simples de um conjunto total de classes. Para tornar a procura eficiente, são eliminadas realizações equivalentes de um mesmo codificador, ou seja, aquelas que geram códigos com mesmas propriedades de distância, [5].

Infelizmente, valores crescentes de  $K$ ,  $N$  e  $CTM$  po-

dem tornar inviável este tipo de procura, a depender das disponibilidades computacionais. Daí, surge a necessidade de se definirem tipos não-exaustivos de procura que sejam eficientes.

## 2.2. A Procura Não-Exaustiva

A procura não-exaustiva é realizada sobre subconjuntos especiais de codificadores de cada classe simples de um conjunto total de classes. A eficiência deste tipo de procura está em serem encontrados subconjuntos adequados, com um pequeno número de codificadores, e com uma alta probabilidade de possuírem bons códigos.

Em [5], foram sistematizados procedimentos que permitam a criação de subconjuntos eficientes, sem a necessidade de um conhecimento profundo da estrutura dos códigos AR.

A economia em esforço computacional permitiu a realização de procura, para classes simples de complexidade de tal a tornar impraticável a procura exaustiva.

A criação destes subconjuntos foi feita a partir da representação por vetores coeficientes dos codificadores de uma dada classe simples. Para tanto, são consideradas as entradas não-codificadas e os vetores coeficientes resultantes são dispostos como linhas de uma matriz, chamada MVC. A partir daí, através da aplicação de certas regras [5], vão-se preenchendo '0', '1', ou 'X' (don't care), nas posições da matriz. O resultado será um subconjunto de codificadores, dentro da classe simples. As posições em 'X' definem possibilidades de codificadores, dentro do subconjunto.

Na fig.5a, está exemplificado um subconjunto de codificadores, pertencente à classe simples  $\{G_{23}; m_1 = 4, m_2 = 0\}$ . As posições em 'X' representam 16 possibilidades de codificadores, a serem analisadas, na procura do codificador ótimo. Na fig.5b, está a sua representação por vetores coeficientes, obtida da matriz MVC, acrescentando-se a linha e a coluna, referentes à entrada não-codificada.

## 3. ALGUNS SUBCONJUNTOS ESPECIAIS E A DISTÂNCIA MÍNIMA DE SEUS CÓDIGOS ÓTIMOS

Neste trabalho, são apresentados subconjuntos especiais de codificadores, para todas as classes simples constantes na fig.4, ao se fixar o número de saídas  $N = K+1$ . Isto corresponde a códigos de taxa  $R = K/(K+1)$ , exatamente aqueles mais promissores, em termos dos parâmetros ganho versus complexidade, para os esquemas de modulação analisados na literatura [1,2,3,4].

Na fig.6, estão estes subconjuntos, representados por suas matrizes MVC. Como a matriz MVC é obtida, a partir dos vetores coeficientes, eliminando-se as entradas não-codificadas, então, várias classes simples têm a mesma matriz MVC. Para obter a representação VC(j) completa, basta que se acrescentem as últimas linhas e colunas, em número igual ao de entradas não-codificadas da classe simples.

A partir destes subconjuntos especiais e do conheci-

mento da estrutura dos códigos AR, foram obtidos os  $\Delta_{MIN}$  (eq.4) dos seus códigos ótimos. A fig. 7 apresenta os valores de  $\Delta_{MIN}^2$  definidos em função dos valores  $\Delta_i$  ( $i = 0, 1, \dots, N-1$ ) da partição de conjunto da constelação de sinais.

A utilidade da tabela da fig. 7 está na simplificação da procura de códigos AR ótimos que esta acarreta. Isto porque, para uma dada constelação de sinais, de um sistema de modulação, e um dado conjunto total de classes de codificadores (definido pelos valores de K e CTM, na fig. 7), pode-se obter a classe simples de máximo  $\Delta_{MIN}$ . A partir daí, ao invés de realizar a procura para todas as possíveis classes simples, analisar apenas a classe simples escolhida. Isto reduz bastante o trabalho de procura de códigos, podendo ser utilizado para qualquer sistema de modulação, em que se aplique a codificação AR.

Um outro aspecto é a possibilidade de união destas informações com as regras heurísticas de construção de códigos AR ótimos, propostas por Ungerboeck [1]. Estas regras, embora não haja prova formal de resultarem nos códigos de desempenhos ótimos, são comuns aos melhores códigos encontrados. No entanto, a aplicação destas regras é subsequente à escolha de uma representação em treliça, para o codificador. Isto equivale à escolha de uma classe simples para o mesmo. A partir da tabela da fig. 7, pode-se determinar qual a classe simples mais apropriada (a de máximo valor para  $\Delta_{MIN}$ ) e, em seguida, aplicar as regras.

Vale, ainda, ressaltar algumas observações, resultantes da análise das figs. 4, 6 e 7, as quais, podem ser estendidas para valores maiores de K e CTM:

- A aplicação das regras de criação de subconjuntos à matriz MVC resulta, para algumas classes simples, na definição do(s) código(s) ótimo(s), eliminando, assim, a necessidade de procura. Um exemplo é a classe simples  $\{G_{23}; m_1 = 1, m_2 = 1\}$ .
- A expressão do  $\Delta_{MIN}$  de classes simples com entradas não-codificadas (transições paralelas na treliça) é o mínimo, entre a expressão para a classe simples que a originou (aquela sem as entradas não-codificadas) e o valor  $\Delta_{u+1}$ , onde u é o número de entradas da classe simples original.
- Os códigos de memória unitária ( $m_1 = m_2 = \dots = m_K = 1$ ) ótimos apresentam  $\Delta_{MIN}^2 = \Delta_1^2 + \Delta_0^2$ , para qualquer valor de K. Desta forma, são códigos pouco eficientes, havendo, pelo menos, uma classe simples mais eficiente, em qualquer conjunto total de classes, com exceção do caso  $K = CTM = 1$ , onde há uma única classe simples.
- Para classes simples em que  $K > 1$ , com  $m_1$  elementos de memória na primeira entrada, e estrutura de memória decrescente ( $m_1 > m_2 > \dots > m_K$ ), então, o código ótimo possui  $\Delta_{MIN}^2 \leq \Delta_{MIN}^{*2}$ , onde  $\Delta_{MIN}^{*2}$  é relativo ao código ótimo da classe simples  $\{G_{12}; m_1\}$ .
- Para um dado conjunto total de classes, há uma tendência dos melhores códigos se encontrarem nas clas-

ses simples com maior número de entradas não-codificadas. Isto está de acordo com as observações de Ungerboeck [1], quanto à otimalidade dos códigos AR.

#### 4. CONCLUSÃO

Neste trabalho, foram apresentados diversos subconjuntos de códigos AR, para diferentes níveis de complexidade de seus codificadores. Esta complexidade foi definida pela taxa de codificação  $R = K/(K+1)$  e pelo comprimento total de memória do codificador CTM, sendo analisados valores de  $K \leq 5$  e  $CTM \leq 5$ . Além disso, foi apresentada uma tabela, com expressões do  $\Delta_{MIN}^2$  dos códigos ótimos destes subconjuntos, em função das distâncias mínimas de subconjunto  $\Delta_i$ , resultantes da partição de conjunto da constelação de sinais. Os subconjuntos de códigos e os respectivos  $\Delta_{MIN}$  dos seus códigos ótimos permitem simplificar consideravelmente o trabalho de procura de códigos AR ótimos, para qualquer sistema de modulação, em que se aplique a codificação AR.

#### REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Ungerboeck, G.: Channel Coding with Multilevel/Phase Signal - IEEE, vol. IT-28, January 1982, p. 55
- [2] Wei, L.F.: Trellis-Coded Modulation with Multidimensional Constellations - IEEE, vol. IT-33, July 1987, p. 483.
- [3] Padovani, R. & Wolf, J.K.: Coded Phase/Frequency Modulation - IEEE, vol. COM-34, May 1986, p. 446.
- [4] Markman, I.: Esquemas de Modulação Codificada em Três e Quatro Dimensões - Tese de Mestrado, UNICAMP SP, junho 1987.
- [5] Markman, I. & Forelli, W.C.: Um Método de Procura de Códigos de Alfabeto Redundante - Anais do 5º SBT, setembro 1987, p. 135.
- [6] Marsan, M.A. et alii: Theoretical Aspects and Performance Evaluation of Channel Encoding with Redundant Modulator Alphabets - Proc. GLOBECOM'84, Atlanta, U.S.A., November 1984, p. 688.

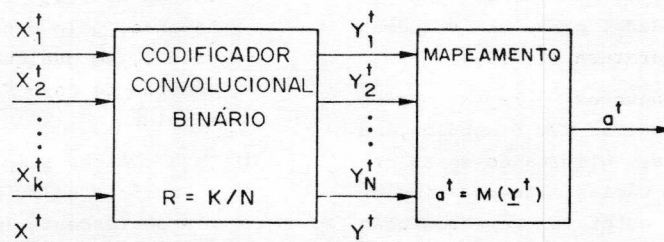


Figura 1 - Estrutura do codificador AR.

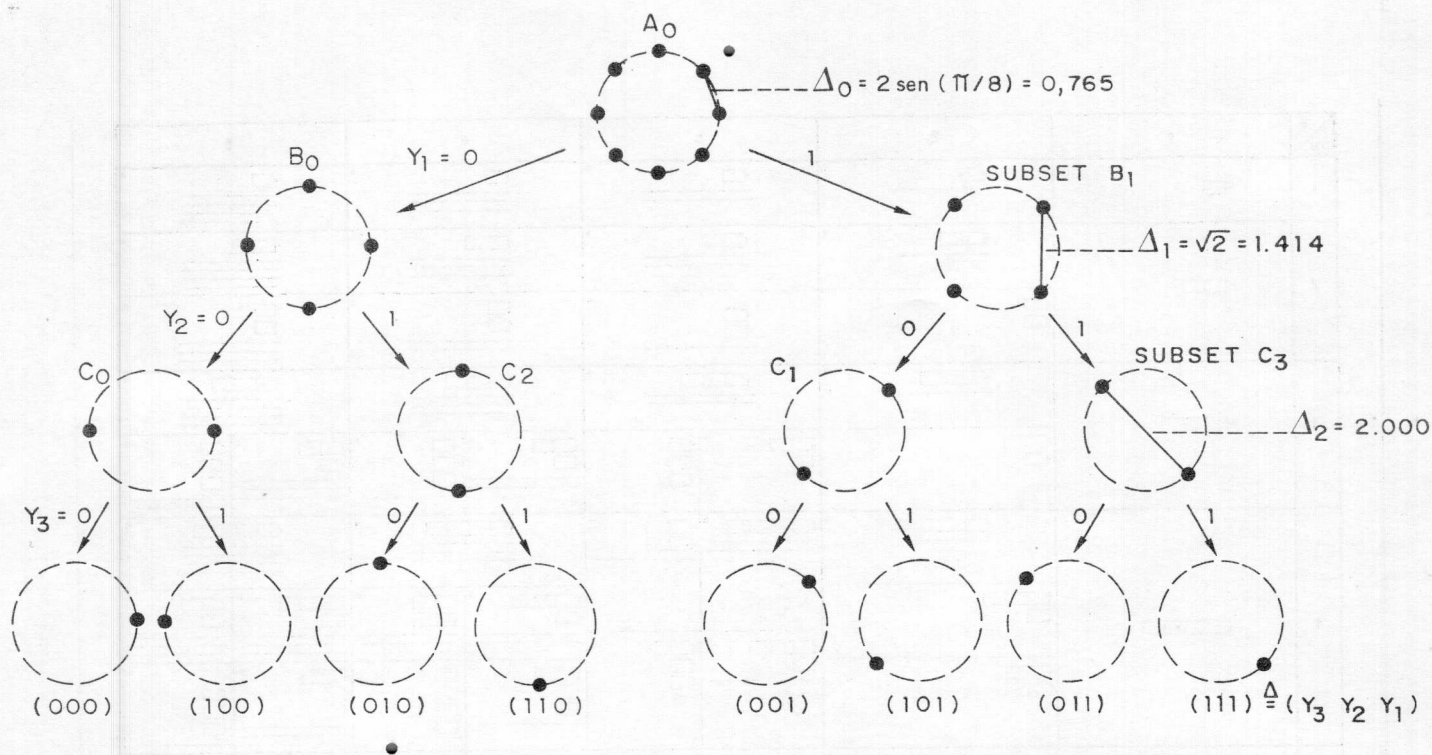
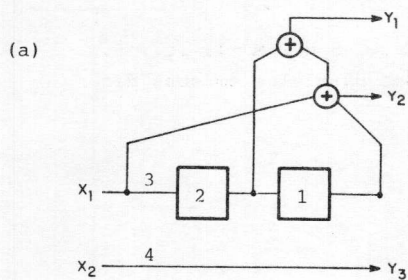


Figura 2 - Método da partição de conjunto aplicado à constelação 8-PSK.



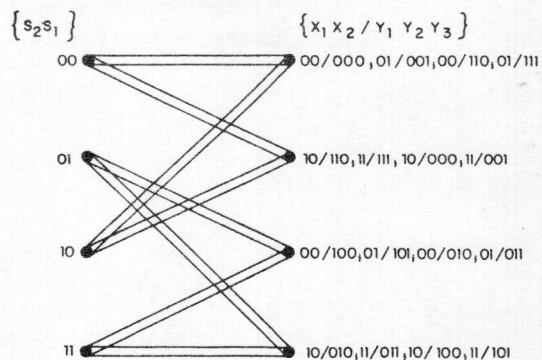
(c)

$$\underline{VC}(j) = | 1 \ 2 \ 3 \ 4 |$$

$$\underline{VC}(1) = | 1 \ 1 \ 1 \ 0 |$$

$$\underline{VC}(2) = | 1 \ 0 \ 1 \ 0 |$$

$$\underline{VC}(3) = | 0 \ 0 \ 0 \ 1 |$$



(b)

Figura 3 - Representações de um codificador convolucional de taxa  $R = 2/3$ :  
a) Esquemática; b) Treliça; c) Vetores coeficientes.

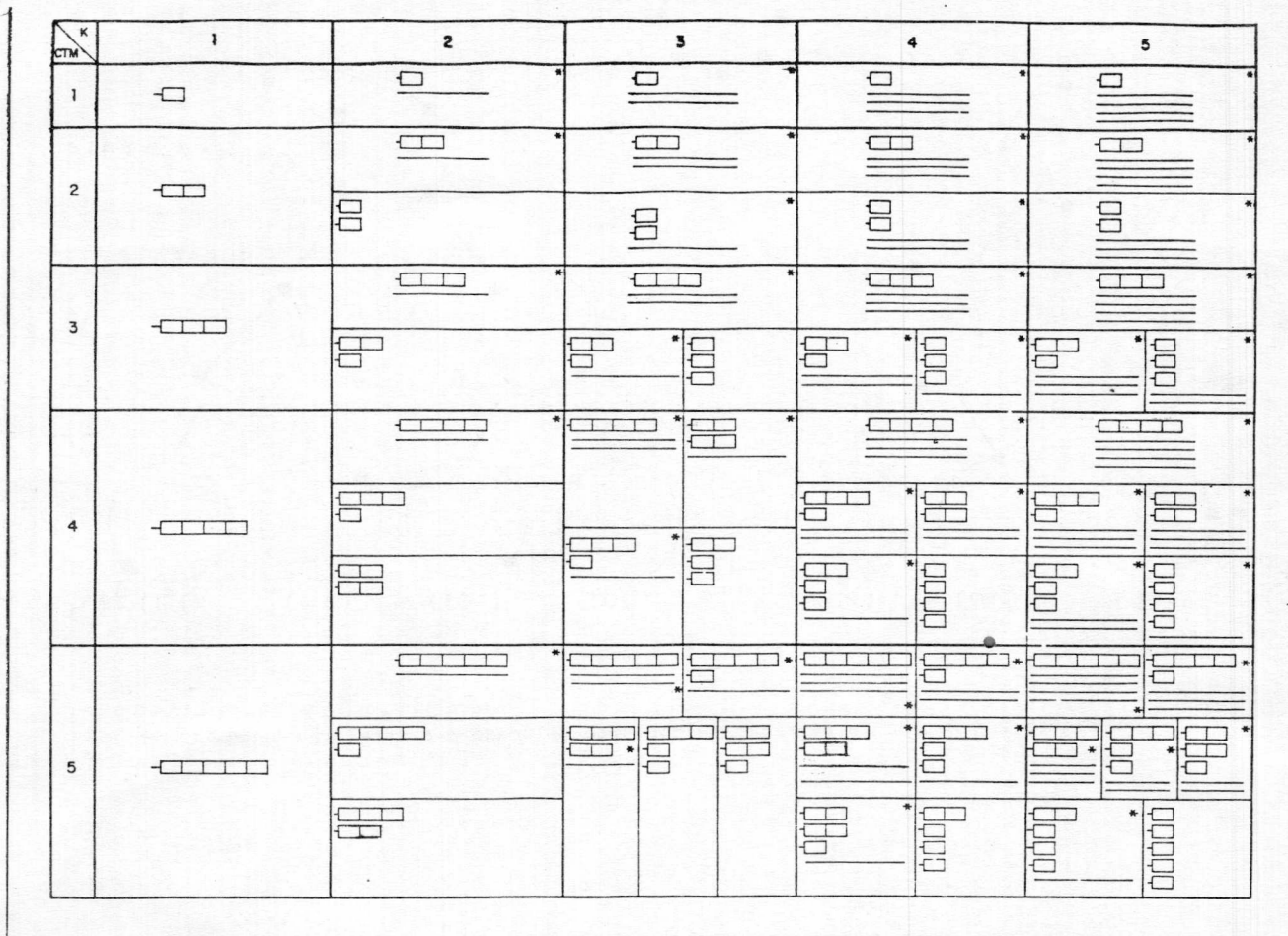


Figura 4 - Estrutura de memória de codificadores com  $K=1, \dots, 5$  e  $CTM=1, \dots, 5$ . O asterisco '\*' significa a presença de transições paralelas em seus diagramas de treliça.

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | X | X | 0 |
| 1 | 0 | X | X | 1 |

(a)

$$\begin{aligned} \underline{VC}(1) &= | 0 \ 1 \ X \ X \ 0 \ 0 | \\ \underline{VC}(2) &= | 1 \ 0 \ X \ X \ 1 \ 0 | \\ \underline{VC}(3) &= | 0 \ 0 \ 0 \ 0 \ 0 \ 1 | \end{aligned}$$

(b)

Figura 5 - Subconjunto de codificadores, pertencente à classe simples  $\{G_{23}; m_1=4, m_2=0\}$ : a) Matriz MVC; b) Vetores Coeficientes.

$$\{G_{K,K+1} ; m_1=1, m_i=0 \ i \geq 2\}$$

|   |   |
|---|---|
| 0 | 1 |
| 1 | 0 |

$$\{G_{K,K+1} ; m_1=3, m_i=0 \ i \geq 2\}$$

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | X | 0 |
| 1 | 0 | X | 1 |

$$\{G_{K,K+1} ; m_1=5, m_i=0 \ i \geq 2\}$$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | X | 1 | 0 | X | 0 |
| 1 | X | X | 1 | X | 1 |

$$\{G_{K,K+1} ; m_1=2, m_2=1, m_i=0 \ i \geq 3\}$$

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | X | 1 |

$$\{G_{K,K+1} ; m_1=2, m_2=2, m_i=0 \ i \geq 3\}$$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 | X | 0 | 0 |
| 0 | 1 | 0 | X | 1 | 0 |
| 1 | 0 | 0 | X | X | 1 |

$$\{G_{K,K+1} ; m_1=3, m_2=2, m_i=0 \ i \geq 3\}$$

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | X | X | 0 | 0 |
| 0 | 0 | 1 | X | X | 1 | 0 |
| 1 | 0 | 0 | X | X | X | 1 |

$$\{G_{K,K+1} ; m_1=2, m_i=0 \ i \geq 2\}$$

|   |   |   |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 1 |

$$\{G_{K,K+1} ; m_1=4, m_i=0 \ i \geq 2\}$$

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | X | 1 | 0 |
| 1 | 0 | X | X | 1 |

$$\{G_{K,K+1} ; m_1=1, m_2=1, m_i=0 \ i \geq 3\}$$

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |

$$\{G_{K,K+1} ; m_1=3, m_2=1, m_i=0 \ i \geq 3\}$$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 1 | X | 0 | 0 | 0 |
| 0 | 0 | X | 1 | 1 | 0 |
| 1 | 0 | X | X | X | 1 |

$$\{G_{K,K+1} ; m_1=4, m_2=1, m_i=0 \ i \geq 3\}$$

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 0 | 1 | X | X | 0 | 0 | 0 |
| 0 | 0 | X | X | 1 | 1 | 0 |
| 1 | 0 | X | X | X | X | 1 |

$$\{G_{K,K+1} ; m_1=1, m_2=1, m_3=1, m_i=0 \ i \geq 4\}$$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | X | 1 |

$$\{G_{K,K+1} ; m_1=2, m_2=1, m_3=1, m_i=0 \ i \geq 4\}$$

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | X | 1 | 0 |
| 1 | 0 | 0 | 0 | X | X | 1 |

$$\{G_{K,K+1} ; m_1=2, m_2=2, m_3=1, m_i=0 \ i \geq 4\}$$

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | X | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | X | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | X | X | X | 1 | 0 |
| 1 | 0 | 0 | X | X | X | X | 1 |

$$\{G_{K,K+1} ; m_1=2, m_2=1, m_3=1, m_4=1, m_i=0 \ i \geq 5\}$$

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | X | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | X | X | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | X | X | X | 1 |

$$\{G_{56} ; m_1=1, m_2=1, m_3=1, m_4=1, m_5=1\}$$

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | X | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | X | X | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | X | X | X | 1 |

$$\{G_{K,K+1} ; m_1=3, m_2=1, m_3=1, m_i=0 \ i \geq 4\}$$

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | X | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | X | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | X | 1 | X | X | 1 | 0 |
| 1 | 0 | X | X | X | X | X | 1 |

$$\{G_{K,K+1} ; m_1=1, m_2=1, m_3=1, m_4=1, m_i=0 \ i \geq 5\}$$

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | X | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | X | X | 1 |

Figura 6 - Subconjuntos de codificadores AR, representados por suas matrizes MVC, para classes simples com  $K \leq 5$  e  $CTM \leq 5$ .

| $\frac{k}{N}$ | 1                           | 2   | 3   | 4   | 5   |
|---------------|-----------------------------|---|---|---|---|
| 1             | $\Delta_1^2 + \Delta_0^2$   | $\text{MIN}(\Delta_2^2; \Delta_1^2 + \Delta_0^2)$   | $\text{MIN}(\Delta_2^2; \Delta_1^2 + \Delta_0^2)$   | $\text{MIN}(\Delta_2^2; \Delta_1^2 + \Delta_0^2)$   | $\text{MIN}(\Delta_2^2; \Delta_1^2 + \Delta_0^2)$   |
| 2             | $2\Delta_1^2 + \Delta_0^2$  | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + \Delta_0^2)$  | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + \Delta_0^2)$  | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + \Delta_0^2)$  | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + \Delta_0^2)$  |
| 3             | $2\Delta_1^2 + 2\Delta_0^2$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 2\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 2\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 2\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 2\Delta_0^2)$ |
| 4             | $2\Delta_1^2 + 3\Delta_0^2$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 3\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 3\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 3\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 3\Delta_0^2)$ |
| 5             | $2\Delta_1^2 + 4\Delta_0^2$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 4\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 4\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 4\Delta_0^2)$ | $\text{MIN}(\Delta_2^2; 2\Delta_1^2 + 4\Delta_0^2)$ |

Figura 7 - Expressões de  $\Delta_{\text{MIN}}^2$  para as classes simples da fig. 4, definidas para taxas  $R = k/N$ , onde  $N = k + 1$ .