

Algoritmos Matriciais para Cálculo do Espectro de Distâncias de Esquemas de Modulação Codificada em Treliça

Luiz Guedes Caldeira e Cecilio Pimentel

Resumo—Este trabalho apresenta dois algoritmos para calcular o espectro de distâncias de esquemas de modulação codificada em treliça (TCM - do inglês *trellis coded modulation*) baseados na manipulação algébrica dos elementos da matriz adjacência do diagrama de super estados (DSE) do codificador. O primeiro algoritmo calcula a função de transferência e o segundo calcula o espectro de n -ésima ordem. Para reduzir a complexidade do DSE é apresentado um algoritmo para a redução do número de estados através do colapso de estados equivalentes. Os algoritmos propostos têm a vantagem de serem facilmente implementados usando linguagens de programação simbólica.

Palavras-Chave—Espectro de distâncias, função de transferência, probabilidade de evento erro, modulação codificada.

Abstract—This paper presents two algorithms to calculate the distance spectrum of trellis coded modulation (TCM) schemes that manipulate iteratively the entries of the adjacency matrix of the error state diagram (DSE). The first algorithm computes the transfer function and the second one finds the truncated distance spectrum of n th order. A state reduction algorithm is also presented to reduce the complexity of the DSE. By using the matrix representation, the distance spectrum becomes easy to implement using a symbolic manipulation program.

Keywords—Error event probability, distance spectrum, transfer function, trellis coded modulation.

I. INTRODUÇÃO

Considere um sistema de comunicação digital que emprega um esquema de modulação codificada em treliça (TCM). Em cada intervalo de sinalização, o transmissor codifica k_c bits de informação em N_c símbolos codificados correspondendo a um ramo na treliça do codificador com 2^ν estados. Se $N_c > 1$, o esquema é denominado MTCM (do inglês *multiple trellis coded modulation*). A taxa do código é definida por $R_c = k_c/N_c$ bits/símbolo.

O cálculo de desempenho de esquemas TCM em canais com ruído aditivo Gaussiano branco baseia-se na enumeração de distâncias Euclidianas e de Hamming entre seqüências de símbolos na saída e na entrada do codificador, respectivamente. Esta enumeração constitui o espectro de distâncias do código. Considere que o decodificador escolhe um percurso na treliça que começa a divergir do percurso transmitido em um intervalo de sinalização fixo, e após L intervalos estes

percursos voltam a coincidir pela primeira vez, dizemos então que ocorreu um *primeiro evento erro* (ou *evento erro* simplesmente) de comprimento L . Para as seqüências de pontos da constelação transmitida $\mathbf{c} = \{c_1, \dots, c_L\}$ e decodificada $\mathbf{e} = \{e_1, \dots, e_L\}$, representado um evento erro de comprimento L , define-se o quadrado da distância Euclidiana entre \mathbf{c} e \mathbf{e} como $d^2 = \sum_{i=1}^L |c_i - e_i|^2$. A probabilidade de ocorrer um evento erro, P_e , é limitada superiormente por [1]:

$$P_e \leq \sum_{i=1}^{\infty} A_i Q \left(\sqrt{\frac{d_i^2 R_c E_b}{2N_0}} \right), \quad (1)$$

sendo E_b/N_0 a relação sinal ruído média por bit, A_i o número médio de eventos erro de qualquer comprimento com distância Euclidiana quadrada d_i^2 e $Q(\cdot)$ a integral da cauda da função de densidade de probabilidade Gaussiana. Um limitante para a probabilidade de erro de bit, P_b , é obtido considerando que eventos erro com distância d_i^2 têm em média B_i bits de informação errados e como o código processa k_c bits de informação por intervalo de sinalização, tem-se [1]:

$$P_b \leq \sum_{i=1}^{\infty} \frac{1}{k_c} B_i Q \left(\sqrt{\frac{d_i^2 R_c E_b}{2N_0}} \right). \quad (2)$$

A enumeração dos conjuntos de n pares $\mathcal{S}_A = \{(A_i, d_i^2)\}_{i=1}^n$ e $\mathcal{S}_B = \{(B_i, d_i^2)\}_{i=1}^n$, considerando as menores distâncias d_i^2 , constitui o *espectro de distâncias* de n -ésima ordem do código TCM, sendo a menor delas, d_1^2 , denominada de *distância livre*.

Fórmulas fechadas para limitantes superiores das probabilidades P_e e P_b também podem ser obtidas a partir da função de transferência do código $T(x, y)$, definida por [1]:

$$T(x, y) = \sum_{d_H} \sum_d C_{d_H, d} x^{d_H} y^{d^2}, \quad (3)$$

sendo $C_{d_H, d}$ o número médio de eventos erro com distância Euclidiana d e distância de Hamming d_H entre as seqüências de informação transmitida e decodificada.

Como os códigos TCM, em geral, não são geometricamente uniformes [2], a enumeração da função de transferência e do espectro de distâncias é realizada através de uma busca em um diagrama de super estados (DSE) do codificador [3]. Os estados do DSE, $\sigma_i, i = 1, \dots, 2^{2\nu}$, são formados por pares de estados da treliça do codificador, $(p, q), p, q = 1, \dots, 2^\nu$, onde p e q são os estados da seqüência transmitida e da seqüência decodificada, respectivamente, em um dado intervalo de sinalização. A transição entre estados do DSE, $\sigma_i \rightarrow$

σ_j , corresponde a transição entre pares de estados $(p, q) \rightarrow (r, s)$, isto é, a seqüência codificada transiciona de p para r e a seqüência decodificada transiciona de q para s . Denotamos por $d_H(\sigma_i \rightarrow \sigma_j)$ e $d^2(\sigma_i \rightarrow \sigma_j)$ a distância de Hamming e o quadrado da distância Euclidiana correspondendo a esta transição. Existem 2^{k_c} transições equiprováveis que divergem de cada estado da treliça do codificador. O rótulo do ramo que conecta a transição de estados $\sigma_i \rightarrow \sigma_j$ do DSE é dado por:

$$\frac{1}{2^{k_c}} \sum_{c,e} x^{d_H(\sigma_i \rightarrow \sigma_j)} y^{d^2(\sigma_i \rightarrow \sigma_j)}. \quad (4)$$

O somatório em (4) indica que os símbolos transmitido e decodificado devem ser escolhidos em todas as transições paralelas que conectam os estados σ_i e σ_j . O somatório não deve ser considerado se não existirem transições paralelas. Em sistemas MTCM as distâncias devem ser obtidas considerando os N_c símbolos em cada ramo da treliça. O estado σ_i é chamado de estado *bom* (G) se $p = q$ ou de estado *ruim* (B) se $p \neq q$. O DSE tem 2^ν estados *bons* e $2^{2\nu} - 2^\nu$ estados *ruins*.

O (i, j) -ésimo elemento da matriz adjacência, denotada por \mathbf{A} , do DSE é o rótulo do ramo correspondente a transição $\sigma_i \rightarrow \sigma_j$. Se não existir esta transição faz-se este elemento igual a 0. O (i, j) -ésimo elemento da L -ésima potência da matriz adjacência \mathbf{A} é um polinômio de 2 variáveis x e y cujos expoentes são as distâncias de Hamming e Euclidiana quadrada, respectivamente, de todos os caminhos no DSE de comprimento L originando em σ_i e terminando em σ_j e os seus coeficientes são as multiplicidades médias associadas a estas distâncias.

Como o número de estados do DSE é $2^{2\nu}$, o processo de busca do espectro de distâncias torna-se computacionalmente ineficiente para valores de ν moderados (por exemplo, $\nu > 4$), tornando necessária uma redução da matriz adjacência do DSE via colapso de estados equivalentes, como proposto em [1], [4], [5].

Neste trabalho são propostos 2 algoritmos para o cálculo do espectro de distâncias de códigos TCM. O primeiro algoritmo calcula a função de transferência a partir da matriz adjacência \mathbf{A} , ou a partir da sua forma reduzida. O algoritmo apresentado é uma adaptação para códigos TCM do algoritmo iterativo proposto inicialmente para códigos convolucionais em [6]. Em cada iteração do algoritmo calcula-se uma matriz adjacência do DSE com 1 estado *ruim* a menos, mas esta preserva toda informação para o cálculo da função de transferência do DSE inicial. Este processo é repetido até que resulte uma matriz adjacência de um DSE com apenas estados *bons*. Neste trabalho, também é proposto um novo algoritmo para calcular o espectro de distâncias de n -ésima ordem a partir da matriz adjacência \mathbf{A} . Propõe-se criar uma seqüência de vetores \mathbf{F}^ℓ , $\ell = 1, \dots, U$, em que os elementos de \mathbf{F}^ℓ são polinômios em x, y que enumeram eventos erro de comprimento ℓ , bem como caminhos de comprimento ℓ que começam em estados *bons* mas que ainda não convergiram para algum estado *bom* (caminhos não terminados). Os polinômios que enumeram os eventos erro atualizam o espectro de n -ésima ordem, enquanto alguns caminhos não terminados podem ser eliminados porque a distância Euclidiana não decresce com ℓ . O algoritmo

termina quando todos os caminhos não terminados forem eliminados.

Este trabalho está organizado da seguinte forma. A Seção II trata do algoritmo de redução de estados equivalentes do DSE. A Seção III apresenta o algoritmo que calcula da função de transferência de códigos TCM. A Seção IV introduz o algoritmo para o cálculo do espectro de distâncias de n -ésima ordem e a Seção V resume as principais conclusões deste trabalho.

II. ALGORITMO PARA REDUÇÃO DA MATRIZ ADJACÊNCIA DO DSE

Um evento erro de comprimento maior que 1 corresponde a um caminho no DSE que começa em um estado *bom*, a primeira transição é para um estado *ruim* e este retorna pela primeira vez a um estado *bom* em algum instante futuro. Um evento erro de comprimento 1 corresponde a transições paralelas entre estados *bons*. Por conveniência, particionaremos a matriz adjacência \mathbf{A} da seguinte forma:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{GG} & \mathbf{A}_{GB} \\ \mathbf{A}_{BG} & \mathbf{A}_{BB} \end{bmatrix}, \quad (5)$$

onde \mathbf{A}_{GG} representa transições entre estados *bons*, \mathbf{A}_{GB} representa transições de estados *bons* para estados *ruins* e similarmente para as outras submatrizes. \mathbf{A}_{GG} , \mathbf{A}_{GB} , \mathbf{A}_{BG} e \mathbf{A}_{BB} têm dimensão $2^\nu \times 2^\nu$, $2^\nu \times 2^{2\nu} - 2^\nu$, $2^{2\nu} - 2^\nu \times 2^\nu$, $2^{2\nu} - 2^\nu \times 2^{2\nu} - 2^\nu$, respectivamente. A função de transferência pode ser obtida diretamente da matriz \mathbf{A} [1]:

$$T(x, y) = \mathbf{\Pi} \mathbf{A}_{GG} \mathbf{1} + \mathbf{\Pi} \mathbf{A}_{GB} (\mathbf{I} - \mathbf{A}_{BB})^{-1} \mathbf{A}_{BG} \mathbf{1}, \quad (6)$$

onde $\mathbf{\Pi}$ é um $1 \times 2^\nu$ vetor linha de probabilidades, cujos elementos são iguais a $1/2^\nu$ (probabilidade de um evento erro iniciar em algum estado *bom*) e $\mathbf{1}$ é um vetor coluna de 1's. O primeiro termo do lado direito da igualdade em (6) corresponde a contribuição de eventos erro de comprimento 1 (percursos paralelos) para a função de transferência, enquanto o segundo termo corresponde a todos os eventos erro de comprimento maiores que 1. Uma forma alternativa para calcular $T(x, y)$ consiste na resolução de um sistema de equações lineares [7].

A complexidade computacional do cálculo de $T(x, y)$ varia de acordo com o grau de simetria do código. Por exemplo, se o código é geometricamente uniforme pode-se fixar uma seqüência particular (em geral a seqüência toda zero) como a seqüência transmitida. Neste caso, o número de estados do DSE pode ser reduzido para 2^ν . Códigos denominados de *quase regulares* [8] não satisfazem esta propriedade de equipartição uniforme, mas a função de transferência pode ser calculada usando um DSE reduzido com 2^ν estados [8]. Uma discussão sobre os vários graus de simetria de códigos TCM e os algoritmos usados para enumerá-los é detalhada em [9]. Entretanto, em casos gerais em que estas simetrias não são observadas, pode-se reduzir a complexidade do DSE eliminando-se estados equivalentes (caso estes existam), obtendo-se uma nova matriz adjacência de dimensão reduzida, sem alterar a função de transferência do código. Esta redução da matriz adjacência será discutida a seguir.

Define-se o rótulo de um caminho no DSE como o produto dos rótulos correspondendo a cada transição de estados que formam o caminho. Dois estados do DSE são ditos equivalentes se o conjunto formado pelos rótulos de todos os caminhos de qualquer comprimento que começam nestes estados são idênticos. Um conjunto de estados equivalentes do DSE, digamos $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$, podem ser colapsados mantendo-se apenas 1 estado representativo da classe (consideramos que este é o de menor índice I_1). Estados com linhas idênticas na matriz adjacência são equivalentes e o colapso destes estados forma uma nova matriz adjacência efetuando-se as seguintes operações: o primeiro passo consiste em adicionar as colunas com índices em \mathcal{I} (adição definida como componente a componente). O resultado desta operação é a coluna I_1 da matriz adjacência reduzida. Em seguida, deve-se zerar as linhas e colunas I_2, \dots, I_n . Estes dois passos consistem em redirecionar para I_1 todos os ramos terminando em I_2, \dots, I_n e eliminar estes estados. Os outros elementos da matriz reduzida são idênticos aos da matriz original. Este procedimento é aplicado separadamente nos estados *bons* e *ruins*. Quando os estados *bons* forem colapsados, o vetor de probabilidades $\mathbf{\Pi} = [\pi_1, \pi_2, \dots, \pi_{2\nu}]$ deve ser atualizado de tal forma que $\pi_{I_1} = \pi_{I_2} + \dots + \pi_{I_n}$ e para os estados colapsados, $\pi_{I_2} = \dots = \pi_{I_n} = 0$. Se após o colapso de conjuntos de estados equivalentes ainda restarem linhas (não nulas) idênticas, aplica-se novamente o procedimento descrito anteriormente até que todas as linhas (não nulas) da matriz sejam distintas. Após todas as reduções, as linhas não nulas da matriz resultante formam uma matriz adjacência reduzida \mathbf{A}' , de dimensão $L_a \times L_a$, com n'_g estados *bons* e n'_b estados *ruins*, onde $L_a = n'_g + n'_b$. O novo $1 \times n'_g$ vetor de probabilidades é denotado por $\mathbf{\Pi}'$.

Exemplo 1: Seja o código de Ungerboeck, $2^\nu = 4$, $R_c = 1$ bit/símbolo, QPSK [2]. Este código possui uma matriz adjacência \mathbf{A} de dimensão 16×16 , que após a aplicação do algoritmo de redução de estados produz uma matriz com $n'_g = 1$, $n'_b = 2$, $L_a = 3$, dada por:

$$\mathbf{A}' = \begin{bmatrix} 1 & xy^4 & 0 \\ 0 & xy^2 & y^2 \\ y^4 & x & 0 \end{bmatrix}. \quad (7)$$

O vetor de probabilidades é $\mathbf{\Pi}' = [1]$.

É importante destacar que o algoritmo apresentado, apesar de poder apresentar uma redução de estados do DSE, não é ótimo quanto ao número de estados reduzidos [5]. Porém, este conta com a vantagem da facilidade de implementação por qualquer software de programação simbólica. As próximas duas seções descreverão os algoritmos para calcular a função de transferência e o espectro de n -ésima ordem de códigos TCM.

III. ALGORITMO PARA CÁLCULO DA FUNÇÃO DE TRANSFERÊNCIA DE CÓDIGOS TCM

O algoritmo apresentado nesta seção para o cálculo da função de transferência de códigos TCM é uma adaptação do algoritmo proposto em [6] para calcular a função de transferência de códigos convolucionais e emprega um algoritmo de redução de estados (SRA - do inglês *state reduction*

algorithm) a partir da matriz reduzida \mathbf{A}' . O objetivo do algoritmo iterativo SRA é calcular a contribuição para a função de transferência de eventos erro de comprimento maiores que 1, reduzindo a cada passo 1 estado *ruim* do DSE e calculando uma nova matriz adjacência que produz a função de transferência do código.

Para calcular a função de transferência, deve-se abrir o DSE eliminando as transições entre os estados *bons*, gerando uma nova matriz adjacência reduzida, \mathbf{A}^* , da forma:

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{0} & \mathbf{A}'_{GB} \\ \mathbf{A}'_{BG} & \mathbf{A}'_{BB} \end{bmatrix}, \quad (8)$$

onde $\mathbf{0}$ é uma submatriz $n'_g \times n'_g$ com todos os elementos iguais a 0. Se $n'_g > 1$, o número de estados *bons* do DSE aberto pode ser reduzido a 1, produzindo uma nova matriz adjacência, denotada por \mathbf{B} , de dimensão $L_b \times L_b$, onde $L_b = 1 + n'_b$, dada por [5]:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{GG} & \mathbf{B}_{GB} \\ \mathbf{B}_{BG} & \mathbf{B}_{BB} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{\Pi}' \mathbf{A}'_{GB} \\ \mathbf{A}'_{BG} \mathbf{1} & \mathbf{A}'_{BB} \end{bmatrix}. \quad (9)$$

Na k -ésima iteração do algoritmo SRA, $k = 2, \dots, L_b$, o estado k é eliminado produzindo a matriz $\mathbf{B}(k)$ a partir da mudança de alguns elementos de $\mathbf{B}(k-1)$, como descrito a seguir. Seja $b_{i,k}^{(k-1)}$ o (i, k) -ésimo elemento de $\mathbf{B}(k-1)$. Para cada $i \neq k$, $i = 1, \dots, L_b$, tal que $b_{i,k}^{(k-1)} \neq 0$ e para cada $j \neq k$, $j = 1, \dots, L_b$, tal que $b_{k,j}^{(k-1)} \neq 0$, o elemento $b_{i,j}^{(k)}$ é igual a [6]:

$$b_{i,j}^{(k-1)} + b_{i,k}^{(k-1)} \left(1 - b_{k,k}^{(k-1)}\right)^{-1} b_{k,j}^{(k-1)}. \quad (10)$$

A eliminação do k -ésimo estado é realizada zerando-se a k -ésima linha e a k -ésima coluna de $\mathbf{B}(k)$. Observe que os rótulos de todos os ramos que iniciam ou terminam neste estado foram redirecionados para outros ramos do DSE. A matriz adjacência inicial é $\mathbf{B}(1) = \mathbf{B}$ e a matriz no final do algoritmo, $\mathbf{B}(L_b)$, tem apenas o elemento $(1, 1)$ não nulo que pode ser interpretado como $\mathbf{B}(L_b)_{GG}$. A função de transferência pode ser calculada da seguinte forma:

$$T(x, y) = \mathbf{\Pi}' \mathbf{A}'_{GG} \mathbf{1} + \mathbf{B}(L_b)_{GG}. \quad (11)$$

O primeiro termo do lado direito da igualdade de (11) deve ser empregado apenas se a treliça do codificador tiver transições paralelas. O algoritmo não utiliza inversão de matrizes levando a uma redução no esforço computacional.

Exemplo 2: A matriz reduzida (7) do Exemplo 1 produz a seguinte matriz inicial do algoritmo RSA:

$$\mathbf{B}(1) = \begin{bmatrix} 0 & xy^4 & 0 \\ 0 & xy^2 & y^2 \\ y^4 & x & 0 \end{bmatrix}. \quad (12)$$

O algoritmo SRA produz a seguinte seqüência de matrizes:

- Eliminação do estado 2:

$$\mathbf{B}(2) = \begin{bmatrix} 0 & 0 & \frac{xy^6}{1-xy^2} \\ 0 & 0 & 0 \\ y^4 & 0 & \frac{xy^2}{1-xy^2} \end{bmatrix}.$$

- Eliminação do estado 3:

$$\mathbf{B}(3) = \begin{bmatrix} \frac{xy^{10}}{1-2xy^2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

A função de transferência é então calculada a partir de (11):

$$T(x, y) = \mathbf{B}(L_b)_{GG} = \frac{xy^{10}}{1-2xy^2}. \quad (13)$$

A partir da matriz adjacência reduzida \mathbf{A}' , o algoritmo SRA resume-se nos seguintes passos:

- A matriz inicial é $\mathbf{B}(1) = \mathbf{B}$ dada em (9).
- Para $k = 2, \dots, L_b$ forme a seqüência de matrizes $\mathbf{B}(k)$.
- A função de transferência é dada em (11).

IV. ALGORITMO ITERATIVO PARA O CÁLCULO DO ESPECTRO DE DISTÂNCIAS DE n -ÉSIMA ORDEM

Uma forma de enumerar o espectro de distâncias de códigos TCM sem calcular a sua função de transferência consiste em avaliar o espectro até a n -ésima ordem de interesse.

Considere o conjunto ordenado $\mathcal{S} = \{(p_i(x), d_i^2)\}_{i=1}^n$ de tal forma que $d_i^2 < d_{i+1}^2$. O polinômio $p_i(x)$ é o coeficiente de $y^{d_i^2}$ em $T(x, y)$. O espectro de distâncias de n -ésima ordem é calculado a partir de \mathcal{S} , visto que:

$$\mathcal{S}_A = \{(p_i(1), d_i^2)\}_{i=1}^n; \quad (14)$$

$$\mathcal{S}_B = \left\{ \left(\frac{dp_i(x)}{dx} \Big|_{x=1}, d_i^2 \right) \right\}_{i=1}^n. \quad (15)$$

O cálculo deste espectro é realizado iterativamente usando uma seqüência de $1 \times L_b$ vetores \mathbf{F}^ℓ , $\ell = 1, \dots, U$, onde cada elemento de \mathbf{F}^ℓ é um polinômio em x, y . O primeiro elemento de \mathbf{F}^ℓ , $\mathbf{F}^\ell(1)$, é a contribuição parcial para a função de transferência de eventos erro de comprimento ℓ . Este polinômio atualiza o conjunto \mathcal{S} em cada iteração. Cada polinômio $\mathbf{F}^\ell(j)$, $j = 2, \dots, L_b$, enumera seqüências que partem do estado *bom* do DSE (estado 1) e terminam no estado *ruim* j em ℓ transições. O critério de parada do algoritmo, aplicado nos polinômios $\mathbf{F}^\ell(j)$, $j = 2, \dots, L_b$, é baseado no fato que o quadrado da distância Euclidiana de um caminho no DSE não decresce com o comprimento do caminho. O cálculo do vetor \mathbf{F}^ℓ é baseado em potências da matriz \mathbf{B} . O algoritmo proposto para calcular o espectro de n -ésima ordem é resumido nos seguintes passos:

- 1 Faça $\ell = 1$, $\mathcal{S} = \emptyset$ e $\mathbf{F}^1 = [0 \mathbf{B}_{GB}]$.
- 2 Se existirem percursos paralelos:
Então calcule $\mathbf{\Pi}' \mathbf{A}'_{GG} \mathbf{1}$. Atualize o conjunto \mathcal{S} com as menores distâncias Euclidianas quadrada e os polinômios $p_i(x)$.
- 3 Faça $\ell = \ell + 1$ e calcule $\mathbf{F}^\ell = \mathbf{F}^{\ell-1} \mathbf{B}$.
- 4 Se $\mathbf{F}^\ell(1) = 0$:
Então vá para o Passo 3.
Caso contrário atualize o conjunto \mathcal{S} a partir de $\mathbf{F}^\ell(1)$, mantendo em \mathcal{S} as n menores distâncias quadradas e seus respectivos polinômios $p_i(x)$. Faça $\mathbf{F}^\ell(1) = 0$.

- 5 Se existirem n distâncias distintas em \mathcal{S} e d_{max}^2 sendo a maior delas:

Então em cada elemento de $\mathbf{F}^\ell(j)$, $j = 2, \dots, L_b$, faça igual a 0 a multiplicidade de cada termo que possuir o expoente de y maior que d_{max}^2 .

Caso contrário vá para o Passo 3.

- 6 Se todos os elementos de $\mathbf{F}^\ell(j)$, $j = 2, \dots, L_b$, são iguais a zero:

Então pare. O algoritmo termina com U iterações.

Caso contrário vá para o Passo 3.

O Passo 5 elimina os caminhos não terminados que não mais atualizarão o conjunto \mathcal{S} no Passo 3 quando estes terminarem em um estado *bom*.

Exemplo 3: Aplicando o algoritmo para calcular o espectro de quarta ordem ($n = 4$) do código do Exemplo 1 verificamos que foram necessárias $U = 9$ iterações. A Tabela I ilustra a atualização do conjunto $\mathcal{S} = \{(p_i(x), d_i^2)\}$ em função do comprimento do evento erro ℓ . Observa-se na tabela que para $\ell = 1, 2$, ainda não é possível listar as distâncias, visto que o primeiro evento erro tem comprimento igual a 3. Observa-se ainda que o código tem distância livre igual a 10. O espectro de distâncias de ordem quarta ordem para este código é dada por:

$$\mathcal{S}_A = \{(1, 10), (2, 12), (4, 14), (8, 16)\}; \quad (16)$$

$$\mathcal{S}_B = \{(1, 10), (4, 12), (12, 14), (32, 16)\}. \quad (17)$$

TABELA I

ESPECTRO DE DISTÂNCIAS DE QUARTA ORDEM DO CÓDIGO TCM DO EXEMPLO 1.

ℓ	$\mathcal{S} = \{(p_i(x), d_i^2)\}, i = 1, 2, 3, 4.$
1,2	\emptyset
3	$\{(x, 10), -, -, -\}$
4	$\{(x, 10), (x^2, 12), -, -\}$
5	$\{(x, 10), (2x^2, 12), (x^3, 14), -\}$
6	$\{(x, 10), (2x^2, 12), (x^3, 14), (x^4, 16)\}$
7	$\{(x, 10), (2x^2, 12), (4x^3, 14), (4x^4, 16)\}$
8	$\{(x, 10), (2x^2, 12), (4x^3, 14), (7x^4, 16)\}$
9	$\{(x, 10), (2x^2, 12), (4x^3, 14), (8x^4, 16)\}$

O próximo exemplo ilustra o cálculo do espectro de terceira ordem de um código MTCM com percursos paralelos e com duas transmissões ($N_c = 2$) entre estados consecutivos da treliça.

Exemplo 4: Considere um código MTCM com 4 estados, QPSK, $N_c = 2$ e $R_c = 1,5$ bits/símbolo [7, Cap. 7, p. 278], conforme ilustrado na Figura 1. Os rótulos deste codificador são representados pelas matrizes de sinais $\mathbf{A}, \mathbf{B}, \dots, \mathbf{H}$ abaixo, cujo número de linhas indicam os percursos paralelos e o número de colunas as N_c transmissões entre estados

consecutivos do codificador.

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} 0 & 0 \\ 4 & 4 \end{pmatrix}, & \mathbf{B} &= \begin{pmatrix} 0 & 4 \\ 4 & 0 \end{pmatrix}, \\ \mathbf{C} &= \begin{pmatrix} 2 & 2 \\ 6 & 6 \end{pmatrix}, & \mathbf{D} &= \begin{pmatrix} 2 & 6 \\ 6 & 2 \end{pmatrix}, \\ \mathbf{E} &= \begin{pmatrix} 0 & 2 \\ 4 & 6 \end{pmatrix}, & \mathbf{F} &= \begin{pmatrix} 0 & 6 \\ 4 & 2 \end{pmatrix}, \\ \mathbf{G} &= \begin{pmatrix} 2 & 0 \\ 6 & 4 \end{pmatrix}, & \mathbf{H} &= \begin{pmatrix} 2 & 4 \\ 6 & 0 \end{pmatrix}. \end{aligned}$$

Aplicando o algoritmo verificamos que foram necessárias $U = 7$ iterações. A Tabela II ilustra a atualização do conjunto $\mathcal{S} = \{(p_i(x), d_i^2)\}$ em função do comprimento do evento erro ℓ . Por uma questão de simplicidade, representaremos o polinômio $p_i(x) = a_0 + a_1x + a_2x^2 + \dots$, como um vetor contendo os seus coeficientes, $[a_0, a_1, a_2, \dots]$. Observa-se na tabela eventos erro de comprimento $\ell = 1$ devido aos percursos paralelos. Observa-se ainda que o código tem distância livre igual a 6. A iteração $\ell = 7$ fornece os conjuntos \mathcal{S}_A e \mathcal{S}_B conforme (14) e (15):

$$\begin{aligned} \mathcal{S}_A &= \left\{ \left(\frac{127}{4}, 6 \right), \left(\frac{1225}{4}, 8 \right), \left(\frac{120943}{64}, 10 \right) \right\}; \\ \mathcal{S}_B &= \left\{ \left(\frac{291}{4}, 6 \right), \left(\frac{17365}{16}, 8 \right), \left(\frac{648645}{64}, 10 \right) \right\}. \end{aligned}$$

TABELA II

ESPECTRO DE DISTÂNCIAS DE TERCEIRA ORDEM DO CÓDIGO MTCM DO EXEMPLO 4

ℓ	$\mathcal{S} = \{(p_i(x), d_i^2)\}, i = 1, 2, 3.$
1	$\{([0, 1], 8), -, -\}$
2	$\{([0, 4, \frac{9}{2}, 2, \frac{1}{2}], 6), ([0, 7, 4, 1], 8), ([0, 0, \frac{9}{2}, 4, \frac{1}{2}], 10)\}$
3	$\{([0, 8, 12, \frac{15}{2}, \frac{9}{2}, \frac{1}{4}], 6), ([0, 17, \frac{31}{2}, 15, \frac{11}{2}, \frac{7}{4}, \frac{15}{4}], 8), ([0, 0, 18, \frac{3}{2}, \frac{69}{4}, 5, \frac{3}{4}], 10)\}$
4	$\{([0, 8, 12, \frac{15}{2}, \frac{9}{2}, \frac{1}{4}], 6), ([0, 29, \frac{61}{2}, \frac{153}{4}, \frac{227}{8}, \frac{59}{4}, \frac{75}{4}], \frac{3}{2}, \frac{1}{2}, 8), ([0, 0, 36, 83, \frac{565}{8}, 49, \frac{99}{4}, 7, \frac{9}{8}], 10)\}$
5	$\{([0, 8, 12, \frac{15}{2}, \frac{9}{2}, \frac{1}{4}], 6), ([0, 41, \frac{91}{2}, \frac{249}{4}, \frac{469}{8}, \frac{163}{4}, \frac{297}{8}], \frac{55}{8}, \frac{35}{16}, \frac{3}{16}, 8), ([0, 0, 57, \frac{549}{4}, \frac{1139}{8}, \frac{203}{2}, \frac{207}{4}, 18, \frac{145}{16}], \frac{3}{8}, 10)\}$
6	$\{([0, 8, 12, \frac{15}{2}, \frac{9}{2}, \frac{3}{2}, \frac{1}{4}], 6), ([0, 41, \frac{91}{2}, \frac{249}{4}, \frac{469}{8}, \frac{163}{4}], \frac{297}{8}, \frac{55}{8}, \frac{35}{16}, \frac{3}{16}, 8), ([0, 0, 81, \frac{797}{4}, \frac{1795}{8}, \frac{1971}{8}, 219, \frac{571}{4}], \frac{1107}{16}, \frac{69}{2}, \frac{35}{8}, \frac{5}{2}, \frac{7}{32}, 10)\}$
7	$\{([0, 8, 12, \frac{15}{2}, \frac{9}{2}, \frac{3}{2}, \frac{1}{4}], 6), ([0, 41, \frac{91}{2}, \frac{249}{4}, \frac{469}{8}, \frac{163}{4}], \frac{297}{8}, \frac{55}{8}, \frac{35}{16}, \frac{3}{16}, 8), ([0, 0, 105, \frac{1045}{4}, \frac{2451}{8}, \frac{2847}{8}, \frac{689}{2}], \frac{3995}{16}, \frac{1119}{8}, \frac{4411}{64}, \frac{503}{32}, \frac{101}{16}, \frac{27}{32}, \frac{1}{8}, 10)\}$

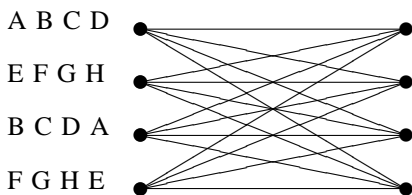


Fig. 1. Treliça para o codificador MTCM do Exemplo 4.

V. CONCLUSÕES

Dois algoritmos foram apresentados para enumerar o espectro de distâncias de códigos TCM. O primeiro calcula a função de transferência do código e o segundo calcula o espectro de n -ésima ordem. Também foi mostrado um algoritmo que realiza um *pré-processamento* da matriz adjacência do DSE, que também é totalmente matricial. A principal vantagem dos algoritmos apresentados é a facilidade de implementação usando qualquer software de computação simbólica. É interessante notar que estes algoritmos podem ser aplicados a outras classes de códigos em treliça, tais como, espacial-temporal.

REFERÊNCIAS

- [1] C. Schelegel, *Trellis Coding*. New York: IEEE Press, 1997.
- [2] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. 28, pp. 55-67, Jan. 1982.
- [3] E. Biglieri, "High level modulation and coding for nonlinear satellite channels," *IEEE Trans. Commun.*, vol. 32, pp. 616-626, May 1984.
- [4] D. K. Aktas and M. P. Fitz, "The distance spectrum of space-time trellis coded modulations in quasi-static Rayleigh fading channels," *IEEE Trans. Inform. Theory*, submitted.
- [5] J. Shi and R. D. Wesel, "Efficient Computation of Trellis Code Generating Functions," *IEEE Trans. Commun.*, vol. 52, pp. 219-227, Feb. 2004.
- [6] C. Pimentel, "On the computation of weight enumerators for convolutional codes," *IEEE Trans. Commun.*, vol. 51, pp. 313-317, Mar. 2003.
- [7] E. Biglieri, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications*. New York: Macmillan, 1991.
- [8] M. Rouanne and D. J. Costello, "An algorithm for computing the distance spectrum of trellis codes," *IEEE J. Sel. Areas Commun.*, vol. 7, pp. 929-940, Aug. 1989.
- [9] S. Benedetto and M. Mondin and G. Montorsi, "Performance evaluation of trellis-coded modulation schemes," *Proceedings of the IEEE*, vol. 82, pp. 833-855, Jun. 1994.