

Técnica de decodificação bidimensional usando algoritmo Belief Propagation

R. F. da Silva, M. S. M. Lins, M. L. M. G. Alcoforado, V. C. da Rocha Jr.

Resumo— Este artigo introduz uma técnica de decodificação bidimensional, que juntamente com os códigos de bloco lineares e o uso da decodificação através do algoritmo Belief Propagation detectam e corrigem erros inseridos na mensagem por meio de um canal binário simétrico.

Palavras-Chave— Códigos Produto, Arranjos Bidimensionais, Belief Propagation.

Abstract— This paper introduces a two-dimensional decoding technique which, along with linear block codes and the use of the belief propagation decoding algorithm, detect and correct errors inserted in the message by a binary symmetric channel.

Keywords— Product Codes, Two-dimensional Arrays, Belief Propagation

I. INTRODUÇÃO

Vários autores têm investigado a decodificação iterativa em duas dimensões, podendo-se citar Tee e Taylor [1], que utilizaram canais com ruído aditivo gaussiano branco (*Additive White Gaussian Noise*, AWGN) e Freitas, que em sua dissertação de mestrado [2], utilizou canais com apagamento. Este artigo investiga um sistema de transmissão digital que utiliza um canal binário simétrico (*Binary Symmetric Channel*, BSC) [3, pág. 23]. O modelo considerado é de um sistema de comunicação ponto a ponto e o comportamento do remetente é similar ao de uma fonte de informação binária sem memória, na qual os símbolos emitidos são equiprováveis.

São apresentadas curvas de desempenho relacionando valores da probabilidade de erro por bit com os da probabilidade de transição do canal, para sistemas de transmissão utilizando códigos bidimensionais compostos por códigos de bloco lineares de Hamming [4, pág. 669], cíclicos [5] e de Reed-Muller [6], [7]. São utilizados dois decodificadores iterativos que empregam o algoritmo *belief propagation* (BP) [8], em que cada decodificador atua em uma dimensão do arranjo bidimensional [2]. Num primeiro momento a decodificação é realizada iterativamente em cada uma das linhas recebidas do arranjo bidimensional. Em seguida a decodificação passa a ser executada nas colunas, potencialmente beneficiando-se das modificações anteriormente ocorridas nas linhas, também de forma iterativa. Verificada a saturação na decodificação iterativa de cada linha e coluna, há o fechamento de uma etapa denominada ciclo. Esta sistemática de ciclos é repetida observando-se a iteração entre linhas e colunas, introduzindo a chamada decodificação duplamente iterativa.

R. F. da Silva, M. S. M. Lins e M. L. M. G. Alcoforado, Departamento de Engenharia de Sistemas, Labetel, Gcom, Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, 50.720-001, E-mail: mlmg@poli.br. V. C. da Rocha Jr., Grupo de Pesquisa em Comunicações, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife, 50.740-550, Email: vcr@ufpe.br.

As demais seções do artigo estão organizadas da seguinte forma. A Seção II traz uma breve explanação sobre o algoritmo BP. A Seção III faz uma síntese sobre a codificação produto serial. A Seção IV introduz a decodificação duplamente iterativa e a Seção V apresenta os casos simulados com seus respectivos resultados. Finalmente, o artigo se encerra na Seção VI, a qual apresenta as conclusões e comentários finais.

II. BELIEF PROPAGATION

O princípio de funcionamento deste algoritmo é uma troca de mensagens entre os nós, através dos grafos de Tanner e de grafos de fatores [3, pág. 38],[8, pág. 476],[9, pág. 281], a fim de propagar probabilidades (crenças) a partir de informações que relacionam os nós do grafo. Os códigos de bloco lineares considerados possuem parâmetros (n, k) , em que n denota o comprimento do bloco e k denota o número de dígitos de informação em cada palavra-código. Consideraremos que cada palavra-código binária tem a estrutura $\mathbf{v} = \{v_0, v_1, v_2, \dots, v_{n-1}\}$, em que n denota um número inteiro positivo. O processo de troca de informações é realizado através de perguntas (questionamentos) e respostas, em que cada nó de variável v_i questiona ao(s) nó(s) de função ao(s) qual(is) ele está conectado, informações que os outros nós de variável, interligados ao mesmo nó de função, possuem referentes ao nó v_i , em que i representa o índice do nó de variável. Cada questionamento $q_{ij}(v_i)$ tem por sua vez uma mensagem de retorno, caracterizada como resposta $r_{ji}(v_i)$, em que j representa o índice do nó de função. É neste momento que os outros nós de variável transmitem a informação a respeito desse questionamento realizado pelo nó inicial [9, págs. 282-284], [10, págs. 196-198]. Para o caso binário, caso em questão para o propósito deste artigo, é possível escrever as equações (1), (2) e (3) a seguir, relacionando as variáveis empregadas pelo decodificador.

$$q_{ij} = K_{ij} p_i \prod_{j' \neq j} r_{j'i}, \quad (1)$$

$$r_{ji} = \frac{1}{2} \left\{ 1 - \prod_{i' \neq i} [1 - 2q_{i'j}] \right\}, \quad (2)$$

$$K_{ij} = \frac{1}{p_i \prod_{j' \neq j} r_{j'i} + (1 - p_i) \prod_{j' \neq j} (1 - r_{j'i})}, \quad (3)$$

em que K_{ij} denota um fator de normalização e p_i denota valor de probabilidade a priori.

É necessário determinar um limiar de decisão, para que a estimativa em cada símbolo seja enfim concretizada. No caso de variáveis binárias, o símbolo v_i pode ser o bit 0 ou o bit 1. Para tal, é necessário calcular a razão de Log Verossimilhança

(Log Likelihood Ratio - LLR) *a Posteriori*, que é dada por (4) [8, pág. 484],[11], i.e.,

$$L(v_i|\mathbf{y}) = \ln \left(\frac{P(v_i = 1|\mathbf{y})}{P(v_i = 0|\mathbf{y})} \right), \quad (4)$$

em que

$$P(v_i = 1|\mathbf{y}) = \frac{p_i \prod_{j' \neq j} r_{j'i}}{p_i \prod_{j' \neq j} r_{j'i} + (1 - p_i) \prod_{j' \neq j} (1 - r_{j'i})}. \quad (5)$$

O valor encontrado com o cálculo da expressão do LLR é usado na aplicação do critério de decisão final, para a determinação da palavra-código estimada $\hat{\mathbf{v}} = \{\hat{v}_0, \hat{v}_1, \dots, \hat{v}_{n-1}\}$, que obedece (6) [8, pág. 484], [11], i.e.,

$$\hat{v}_i = \begin{cases} 1, & \text{se } L(v_i|\mathbf{y}) > 0, \\ 0, & \text{se } L(v_i|\mathbf{y}) < 0. \end{cases} \quad (6)$$

III. CODIFICADOR

O sistema de transmissão digital usado neste artigo está ilustrado na Figura 1. A codificação produto [12] na estrutura serial é realizada através de C^- e C^l , codificadores de bloco lineares e sistemáticos associados aos códigos com parâmetros (n_1, k_1) e (n_2, k_2) , respectivamente. No que segue consideraremos códigos associados a C^- e C^l com parâmetros idênticos (n, k) . Cada bloco de k mensagens geradas na fonte de informação é organizado em uma matriz denominada \mathbf{U} , assim cada linha de \mathbf{U} , lida da esquerda para a direita, corresponde a um dado bloco de mensagem. Estes k blocos entram inicialmente no codificador horizontal C^- , e cada uma das k palavras-código geradas vai corresponder a uma linha de uma matriz denominada \mathbf{A}_h . Inicialmente portanto, a matriz \mathbf{A}_h tem k linhas e n colunas. Em seguida é feito o entrelaçamento [12] e cada um dos n blocos de entrada do codificador vertical C^l , corresponde a uma coluna de \mathbf{A}_h , lida de cima para baixo, a fim de adicionar bits de paridade nas linhas inferiores, originando n palavras-código no sentido vertical. Ao final da codificação realizada pelo codificador C^l , é originada uma matriz denominada \mathbf{A}_p , que por sua vez possui n linhas e n colunas. Em seguida, todos os bits da matriz \mathbf{A}_p são transmitidos pelo canal BSC, e reorganizados em uma outra matriz denominada \mathbf{A}_r , que representa uma versão ruidosa de \mathbf{A}_p .

IV. DECODIFICAÇÃO DUPLAMENTE ITERATIVA

O processo de decodificação, também ilustrado na Figura 1, tem D^- e D^l representando o decodificador horizontal e o decodificador vertical, respectivamente. Em ambas as dimensões é utilizado o algoritmo de decodificação BP.

Inicialmente o processo de decodificação, através de D^- , é aplicado em cada linha da matriz \mathbf{A}_r , sendo realizadas um número tal de iterações por linha, que a partir deste não haja melhorias. A matriz \mathbf{A}_r é atualizada dando-se início ao processo de decodificação vertical, através de D^l , aplicado em cada coluna da matriz \mathbf{A}_r . O número de iterações realizadas nas colunas segue o mesmo número aplicado nas linhas e a matriz \mathbf{A}_r é novamente atualizada. Em seguida, inicia-se um novo ciclo, permitindo a iteração entre linhas e colunas,

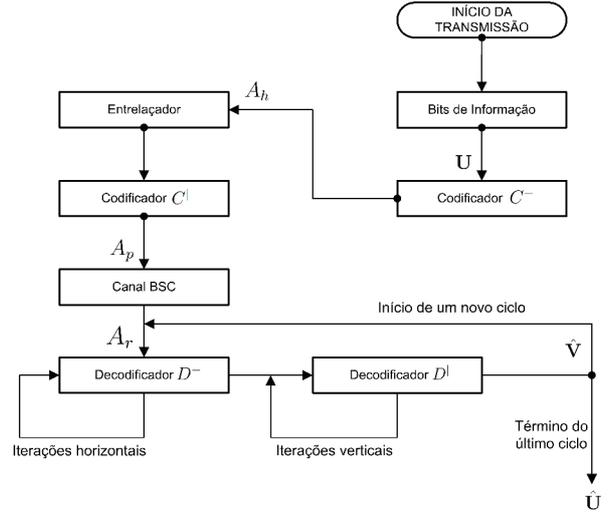


Fig. 1

ESTRUTURA DO SISTEMA DE TRANSMISSÃO DIGITAL.

com o decodificador D^- novamente tratando as linhas, e em seguida o decodificador D^l atuando nas colunas. Este processo de repetição dos ciclos encerra quando não trouxer mais melhorias de desempenho, ou seja, quando houver saturação. Este processo de decodificação é chamado de duplamente iterativo, uma vez que a decodificação, além de ser feita com iterações entre linhas e colunas, também é feita iterativamente em cada bloco de palavras-código, seja ele horizontal ou vertical.

Exemplo 1: Admita um bloco de $k = 4$ mensagens sendo elas: 1100, 1000, 0001, 1010, compondo a matriz \mathbf{U} , em que

$$\mathbf{U} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

São utilizados códigos de Hamming $C(7,4)$ com a matriz de verificação de paridade \mathbf{H} , mostrada a seguir, tanto no codificador horizontal C^- como no vertical C^l , i.e.,

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Inicia-se o processo de construção da matriz \mathbf{A}_p , utilizando C^- para codificar horizontalmente, obtendo portanto a matriz \mathbf{A}_h a seguir

$$\mathbf{A}_h = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Após a criação das paridades horizontais, o entrelaçamento segue e o codificador C^l insere os bits de paridade em cada uma das 7 colunas de \mathbf{A}_h , obtendo desta forma \mathbf{A}_p , i.e.,

$$\mathbf{A}_p = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Com a montagem da matriz \mathbf{A}_p , o processo de codificação é concluído. A próxima etapa é enviar \mathbf{A}_p pelo canal BSC de transmissão, nesse exemplo com probabilidade de transição $p = 0,1$. A matriz \mathbf{A}_p portanto, pode ter alguns de seus bits modificados pela inserção do ruído, sendo reorganizados na matriz \mathbf{A}_r , i.e.,

$$\mathbf{A}_r = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

O decodificador D^- irá atuar em cada linha. Observando a linha 1 da matriz \mathbf{A}_r , é possível verificar que esta corresponde a uma palavra do código, portanto, a linha 1 não é alterada e o processo de decodificação irá avançar para a próxima linha. De maneira análoga, a linha 2 também corresponde a uma palavra do código. Sendo assim, o decodificador segue para a linha 3.

O processo de troca de mensagens do BP [8] é iniciado com os nós-folha, indexados por p_i , enviando para cada nó v_i , respectivamente, seus valores de probabilidade a priori, conforme (7), i.e.,

$$p_i = p(y_i|v_i) = \begin{cases} 1-p, & \text{se } y_i = v_i, \\ p, & \text{se } y_i \neq v_i. \end{cases} \quad (7)$$

Após a montagem do grafo de fatores, é possível obter os valores de q_{ij} que, para a primeira iteração, são decorrentes dos valores de p_i , como exemplo:

$$\begin{aligned} q_{00} &= q_{01} = 0,1, \\ q_{30} &= q_{32} = 0,9. \end{aligned}$$

Por consequência, é necessário que os nós de paridade respondam com a informação solicitada no questionamento. Utilizando (2), verifica-se que:

$$\begin{aligned} r_{00} &= r_{02} = r_{15} = r_{21} = r_{22} = 0,244, \\ r_{03} &= r_{04} = r_{10} = r_{11} = r_{12} = r_{23} = r_{26} = 0,756. \end{aligned}$$

De acordo com (4) são calculados os valores de $L(v_i|\mathbf{y})$. Exemplificando o cálculo de $L(v_1|\mathbf{y})$, tem-se que:

$$p(v_0 = 1|\mathbf{y}) = \ln \left(\frac{0,1 \times 0,244 \times 0,756}{0,1 \times 0,756 \times 0,244 + 0,9 \times 0,244 \times 0,756} \right) = 0,1,$$

$$L(v_0|\mathbf{y}) = \ln \left(\frac{p(v_0=1|\mathbf{y})}{p(v_0=0|\mathbf{y})} \right) = \ln \left(\frac{0,1}{1-0,1} \right) = \ln \left(\frac{0,1}{0,9} \right) = -2,19.$$

De maneira análoga é realizado o cálculo dos seguintes valores de $L(v_i|\mathbf{y})$, conforme a Tabela I.

Aplicando (6), é possível determinar os bits estimados de $\hat{\mathbf{v}}$, que por se tratar de decodificação horizontal serão

TABELA I
VALORES DO LLR DA DECODIFICAÇÃO HORIZONTAL

$L(v_0 \mathbf{y})$	$L(v_1 \mathbf{y})$	$L(v_2 \mathbf{y})$	$L(v_3 \mathbf{y})$	$L(v_4 \mathbf{y})$	$L(v_5 \mathbf{y})$	$L(v_6 \mathbf{y})$
-2,19	-2,19	-3,32	4,45	3,32	1,06	3,32

TABELA II
VALORES \hat{v}_i^-

$\hat{v}_{3,0}^-$	$\hat{v}_{3,1}^-$	$\hat{v}_{3,2}^-$	$\hat{v}_{3,3}^-$	$\hat{v}_{3,4}^-$	$\hat{v}_{3,5}^-$	$\hat{v}_{3,6}^-$
0	0	0	1	1	1	1

denominados $\hat{v}_{l,i}^-$, em que o índice l representa a linha que está sendo decodificada, que neste caso é 3, conforme a Tabela II.

Para finalizar a primeira iteração, é necessário calcular os valores de q_{ij} utilizando (1), sobrescrevendo os valores anteriores, para que possam ser utilizados na próxima iteração.

Os valores de $\hat{v}_l^- = \{\hat{v}_{l,0}^-, \hat{v}_{l,1}^-, \dots, \hat{v}_{l,n-1}^-\}$ obtidos em cada iteração, serão inseridos na linha l da matriz \mathbf{A}_r , substituindo os valores antigos, para que possa auxiliar a decodificação na próxima iteração. Admitindo que a quantidade de iterações escolhidas pelo usuário seja 10, o valor que ficará no final de todo o processo na linha l , será o valor obtido na décima iteração, para que seja utilizado pelo processo de decodificação vertical.

Após 10 iterações em cada linha, é analisada a próxima linha até a linha n , que neste caso, é a linha 7. Quando completada a análise das n linhas, será iniciado o processo de decodificação vertical, utilizando a nova matriz \mathbf{A}_r com os novos valores obtidos da decodificação horizontal.

Ao final das 10 iterações em todas as linhas, tem-se a nova matriz \mathbf{A}_r , indicada a seguir

$$\mathbf{A}_r = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Note que o processo de decodificação horizontal já foi capaz de corrigir alguns erros inseridos na matriz \mathbf{A}_r pelo canal BSC.

O processo de decodificação vertical é realizado da mesma maneira que o processo de decodificação horizontal, a diferença é que cada vetor a ser analisado será extraído das colunas. Sendo assim, o conjunto de vetores recebidos pelo decodificador vertical é $\mathbf{Y} = (1100000, 1000110, 0001101, 0010101, 1111010, 0111011, 1011110)$, que simplesmente são as colunas da matriz \mathbf{A}_r , reorganizadas como vetores linha.

A reorganização de colunas faz com que o decodificador vertical tenha exatamente o mesmo processo de funcionamento do decodificador horizontal, com apenas uma ressalva, os valores de $\hat{\mathbf{v}}$ denominados de $\hat{v}_c^l = \{\hat{v}_{c,0}^l, \hat{v}_{c,1}^l, \dots, \hat{v}_{c,n-1}^l\}$ obtidos em cada iteração, serão inseridos na coluna c da matriz \mathbf{A}_r , substituindo os valores antigos, para que possa auxiliar a decodificação na próxima iteração.

Após 10 iterações em cada coluna, será analisada a próxima coluna até a coluna n , que neste caso é a coluna 7. Quando completada a análise das n colunas é concluído um ciclo e será retomado o processo de decodificação horizontal, utilizando a nova matriz A_r com os novos valores obtidos da decodificação vertical.

Este processo de decodificação horizontal e vertical ocorrerá por 10 ciclos. No final deste processo, tem-se a matriz A_r final.

$$A_r = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Uma vez concluídas as operações de decodificação para obter a matriz A_r final, basta remover as colunas que representam os bits de paridade (colunas 5, 6 e 7) e as linhas que também representam os bits de paridade (linhas 5, 6 e 7). Dessa forma tem-se as 4 mensagens estimadas organizadas em uma matriz \hat{U} indicada a seguir

$$\hat{U} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Cada linha de \hat{U} representa uma mensagem estimada. Como as mensagens enviadas no início do processo foram 1100, 1000, 0001, 1010, verifica-se que o decodificador corrigiu todos os erros inseridos pelo canal BSC.

V. SISTEMAS DE TRANSMISSÃO SIMULADOS

Para efeitos de comparação, as Figuras 2, 3, 4, 5 e 6 apresentam curvas geradas a partir de simulação computacional de sistema utilizando a codificação unidimensional e decodificação iterativa (algoritmo BP), bem como curvas geradas a partir do sistema ilustrado na Figura 1 utilizando codificação bidimensional e a decodificação duplamente iterativa (algoritmo BP em cada dimensão). A quantidade de iterações utilizada representa um número tal que a partir dele não há melhorias significativas de desempenho. Os parâmetros utilizados para cada caso simulado são indicados a seguir.

- 1) A curva apresentada na Figura 2 para o caso unidimensional é gerada a partir do uso do codificador de Hamming $C(7, 4)$ e decodificação BP com 3 iterações. As curvas apresentadas para o caso bidimensional foram geradas a partir do uso do código produto $C_p(49, 16)$, tendo como componentes dois codificadores de Hamming, $C^- = C^l = C(7, 4)$. Em cada ciclo, são executadas 3 iterações por linha e por coluna.
- 2) A curva apresentada na Figura 3 para o caso unidimensional é gerada a partir do uso do codificador cíclico $C(15, 5)$ com polinômio gerador $g(x) = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$ e decodificação BP com apenas 1 iteração. As curvas apresentadas para o caso bidimensional foram geradas a partir do uso do código produto $C_p(225, 25)$, tendo como componentes dois codificadores cíclicos na

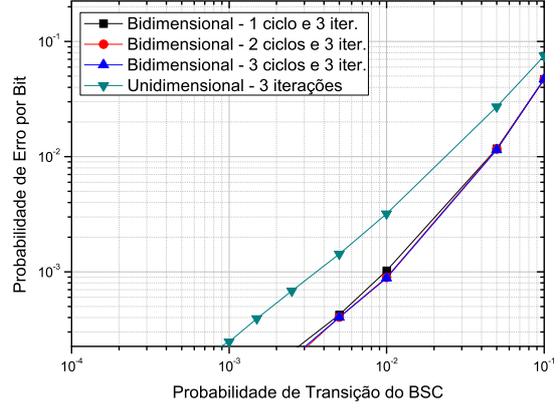


Fig. 2

RESULTADOS OBTIDOS PARA O CASO 1

forma sistemática, $C^- = C^l = C(15, 5)$. Em cada ciclo são executadas 1 iteração por linha e por coluna.

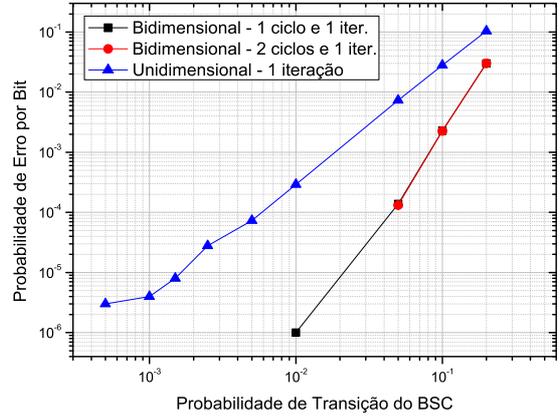


Fig. 3

RESULTADOS OBTIDOS PARA O CASO 2

- 3) A curva apresentada na Figura 4 para o caso unidimensional é gerada a partir do uso do codificador cíclico $C(21, 7)$ com polinômio gerador $g(x) = 1 + x^3 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{13} + x^{14}$ e decodificação BP com apenas 1 iteração. As curvas apresentadas para o caso bidimensional foram geradas a partir do uso do código produto $C_p(441, 49)$, tendo como componentes dois codificadores cíclicos na forma sistemática, $C^- = C^l = C(21, 7)$. Em cada ciclo, são executadas 1 iteração por linha e por coluna.
- 4) A curva apresentada na Figura 5 para o caso unidimensional é gerada a partir do uso de codificador sistemático, equivalente ao de Reed-Muller $C(8, 4)$, e decodificação BP com 3 iterações. As curvas apresentadas para o caso bidimensional foram geradas a partir do uso do código produto $C_p(64, 16)$, tendo como componentes dois codificadores iguais aos usados para o caso unidimensional, $C^- = C^l = C(8, 4)$. Em cada ciclo, são executadas 3 iterações por linha e por coluna.
- 5) A curva apresentada na Figura 6 para o caso unidimensional é gerada a partir do uso de codificador sistemático,

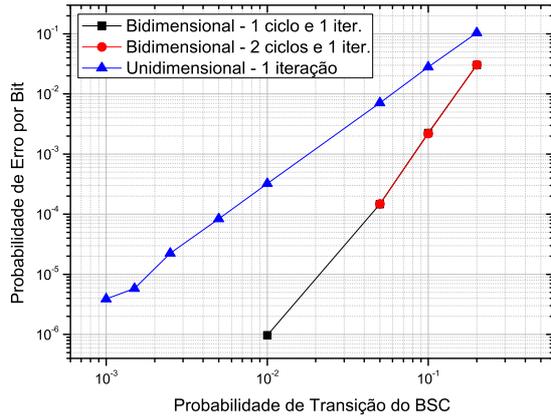


Fig. 4

RESULTADOS OBTIDOS PARA O CASO 3

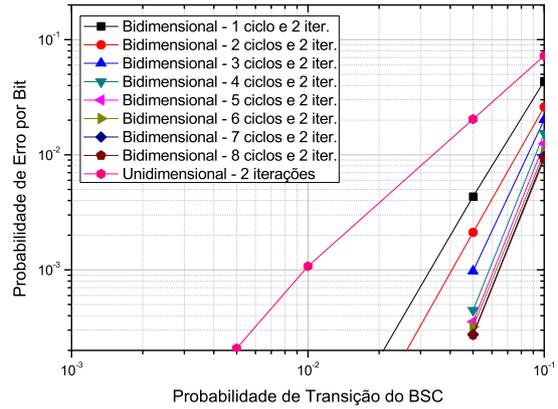


Fig. 6

RESULTADOS OBTIDOS PARA O CASO 5

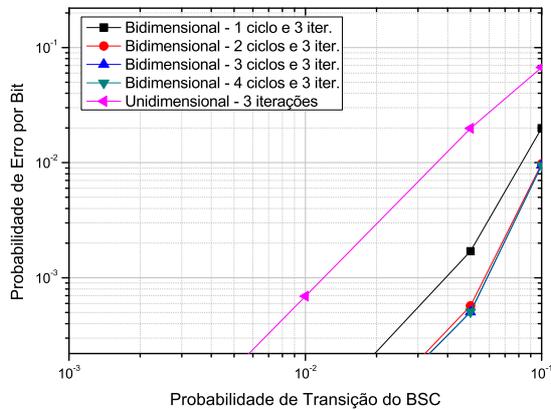


Fig. 5

RESULTADOS OBTIDOS PARA O CASO 4

equivalente ao de Reed-Muller $C(16, 5)$, e decodificação BP com 2 iterações. As curvas apresentadas para o caso bidimensional foram geradas a partir do uso do código produto $C_p(256, 25)$, tendo como componentes dois codificadores iguais aos usados para o caso unidimensional, $C^- = C^l = C(16, 5)$. Em cada ciclo, são executadas 2 iterações por linha e por coluna.

VI. CONCLUSÕES

É um fato bastante conhecido, na área de códigos corretores de erros, que a codificação bidimensional traz melhorias de desempenho do sistema de transmissão ao custo do uso de uma maior largura de banda. A intenção neste artigo é fazer uma análise quantitativa desta melhoria, considerando a decodificação denominada de duplamente iterativa em que são realizadas várias iterações em apenas uma dimensão até que se obtenha a estagnação de melhoria do desempenho, só então a decodificação é realizada em outra dimensão, concluindo o ciclo. A partir de análise nas curvas mostradas nas Figuras 2, 3, 4, 5, 6 verifica-se qual a melhor região a se trabalhar, a custo de qual esforço computacional desenvolvido.

Analisando a Figura 2, verifica-se que a decodificação iterativa unidimensional satura com três iterações, por este motivo, foram utilizadas 3 iterações por dimensão em cada

ciclo do caso bidimensional com dupla iteratividade. A partir do segundo ciclo já não há melhorias significativas. Para os casos 2 e 3 cujas curvas estão ilustradas respectivamente nas Figuras 3, 4, a decodificação iterativa unidimensional satura com apenas 1 iteração, sendo utilizada apenas 1 iteração por dimensão em cada ciclo do caso bidimensional com dupla iteratividade. Para ambos os casos não há melhorias a partir do segundo ciclo. Considerando o caso 4, cuja curva está ilustrado na Figura 5, a decodificação unidimensional satura com 3 iterações, sendo portanto utilizadas 3 iterações por dimensão em cada ciclo do caso bidimensional com dupla iteratividade. Verifica-se que não há melhorias de desempenho a partir do terceiro ciclo. Finalmente, considerando o caso 5, cuja curva está ilustrada na Figura 6, a decodificação unidimensional satura com 2 iterações, sendo portanto utilizadas 2 iterações por dimensão em cada ciclo do caso bidimensional com dupla iteratividade. Verifica-se que não há melhoria significativas de desempenho a partir do quarto ciclo.

REFERÊNCIAS

- [1] J. S. K. Tee, D. Taylor, "Iterative decoding of systematic binary algebraic block codes," *Global Telecommunications Conference*, vol. 2, pp. 842-846, 2000.
- [2] P. R. de Freitas, "Decodificação iterativa com baixa complexidade para o canal binário com apagamento," Dissertação de Mestrado, UFPE, 2011.
- [3] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley, 2005.
- [4] S. S. Haykin, *Communication systems*. Wiley, 2001.
- [5] P. R. L. Martins, "Correção de manchas de erros em arranjos bidimensionais," Dissertação de Mestrado, UFPE, 2012.
- [6] J. A. Davis e J. Jedwab, "Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes," in *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2397-2417, 1999.
- [7] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447-449, 2008.
- [8] S. A. Abrantes, *Códigos Correctores de Erros em Comunicações Digitais*. FEUP Edições, 2010.
- [9] J. C. Moreira e P. G. Farrell, *Essentials of Error-Control Coding*. Wiley, 2006.
- [10] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. Wiley, 2006.
- [11] J. Hagenauer, E. Offer, e L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445, 1996.
- [12] I. M. M. de Souza, "Decodificação iterativa bidimensional para canais de acesso múltiplo," Dissertação de Mestrado, UPE, 2015.