# Decodificação iterativa com baixa complexidade sobre o canal binário com apagamento

Paulo Roberto de Freitas e Valdemar C. da Rocha Jr.

***Resumo*— A decodificação iterativa em duas dimensões sobre o canal binário com apagamento é investigada, quando códigos cíclicos lineares binários idênticos são usados em cada uma das duas dimensões. Uma característica importante da matriz de verificação de paridade dos códigos cíclicos permite, em vários casos de interesse prático, uma implementação com baixa complexidade da decodificação iterativa com permutação cíclica, ao invés da conhecida complexidade cúbica. Por meio de simulação computacional é observado um ganho considerável de desempenho quando é usada a decodificação iterativa com permutação cíclica ao invés da decodificação iterativa sem permutação cíclica. É também apresentada uma análise comparativa de diversos algoritmos de decodificação iterativa em duas dimensões, usando como exemplos os códigos produto $(49, 16, 9)$, $(225, 121, 9)$ e $(225, 49, 25)$, incluindo os resultados das simulações.**

***Palavras-Chave*— Códigos corretores de apagamentos, códigos cíclicos, decodificação iterativa em duas dimensões, canal binário com apagamento.**

***Abstract*— Iterative decoding in two dimensions over the binary erasure channel is investigated when identical linear binary cyclic codes are employed in each dimension. An important characteristic of the parity-check matrix of cyclic codes allows, in many cases of practical interest, an implementation with low complexity of iterative decoding with cyclic permutation, instead of the well known cubic complexity. By means of computer simulation a considerable gain in performance is observed when iterative decoding with cyclic permutations is employed instead of iterative decoding without cyclic permutations. It is also presented a comparative analysis of various iterative decoding algorithms in two dimensions, using as examples the product codes (49, 16, 9), (225, 121, 9) and (225, 49, 25), including simulation results.**

***Keywords*— Error-correcting codes, cyclic codes, two-dimensional iterative decoding, binary erasure channel.**

## I. INTRODUÇÃO

O canal binário com apagamento, denotado pela sigla BEC (*Binary Erasure Channel*), possui duas entradas representadas por $\{0, 1\}$ e três saídas representadas por $\{0, 1, \Delta\}$, em que o símbolo $\Delta$ denota um apagamento. Os bits transmitidos através de um BEC são recebidos corretamente com probabilidade $1 - \varepsilon$, ou são recebidos apagados com probabilidade $\varepsilon$, onde $0 \leq \varepsilon \leq 1$. Este modelo de canal foi introduzido por Elias [1] em 1955. Um código $C$, com distancia mínima $d$, é capaz de corrigir todos os padrões de apagamento contendo até $\rho$ apagamentos se $\rho \leq d - 1$ [2, págs. 13-14]. O BEC é um modelo de canal adequado para redes de dados cujos bits transmitidos podem ser perdidos devido a falhas de rede, tais como demora excessiva ou congestionamento nos nós intermediários, como por exemplo a Internet [3], [4]. Como é mencionado em [3], a decodificação iterativa em duas dimensões sobre o BEC tem recebido relativamente pouca atenção até o momento.

Paulo Roberto de Freitas e Valdemar C. da Rocha Jr., Grupo de Pesquisa em Comunicações - CODEC, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife, CEP 50711-970, Caixa Postal 7800, E-mails:paulo.freitas@ufpe.com.br, vcr@ufpe.br. Este trabalho foi parcialmente financiado pela CAPES e pelo CNPq, Proj. 304696/2010-2.

A notação $(n, k, d)$ é usada para denotar um código de bloco linear $C$, no qual $n$ representa o comprimento da palavra-código, $k$ o número de bits de informação e $d$ a distância mínima entre palavras-código distintas. Considerando dois códigos de bloco $C_1$ e $C_2$, Elias [5] introduziu o código produto $C = C_1 \times C_2$, cujo dicionário é o conjunto de todas as matrizes $n_1 \times n_2$ cujas colunas são palavras-código de $C_1$ e cujas linhas são palavras-código de $C_2$. O código produto $C$ apresenta os seguintes parâmetros: $n = n_1 n_2$, $k = k_1 k_2$ e $d = d_1 d_2$. O código produto tem propriedades interessantes que faz com que ele tenha aplicação prática em diversos padrões de telecomunicações como, por exemplo, o IEEE 802.16 (WiMAX), CD padrão IEC-908 e DVD [6]. Entre as propriedades do código produto inclui-se um algoritmo simples de decodificação iterativa linha-coluna proporcionando um bom equilíbrio entre desempenho e complexidade em um canal com ruído Gaussiano branco aditivo (AWGN) [7]. Utilizando propriedades dos códigos produto cíclicos é investigado neste artigo um algoritmo de decodificação iterativa de baixa complexidade sobre um canal BEC.

Códigos simples $(7, 4, 3)$, $(15, 11, 3)$ e $(15, 7, 5)$, foram utilizados nas simulações, para evidenciar a eficácia dos mesmos em arranjos bidimensionais. As simulações também mostram que muitos padrões de apagamentos, com número de posições afetadas muito maior que $d^2 - 1$ são corrigidos. Uma complexidade computacional linear por iteração, com o comprimento do código, resultou para esta técnica, sendo o número de iterações aproximadamente $n/2$, tipicamente um número entre 4 e 9, para os códigos em tela.

Este artigo está organizado do seguinte modo. Na Seção II é apresentada uma breve revisão sobre códigos cíclicos. Na Seção III é introduzida uma simplificação na decodificação iterativa, que permite a sua implementação com baixa complexidade. Resultados de simulação em computador e comparação de desempenho são tratadas na Seção IV e, finalmente, na Seção V são apresentadas algumas conclusões.

## II. CÓDIGOS CÍCLICOS LINEARES

Um código $C$ é dito ser um código cíclico linear se, além das vantajosas propriedades da linearidade [2], agregar a propriedade que qualquer palavra-código deslocada ciclicamente também é uma palavra do código $C$. Neste trabalho serão considerados exclusivamente códigos cíclicos lineares

binários, de modo que daqui por diante a denominação *código cíclico* implicitamente refere-se a um código cíclico linear binário. Os códigos cíclicos obtiveram grande destaque em aplicações práticas como, por exemplo, no *compact disc* (CD) e na NASA (*Deep Space Standard*) para comunicações via satélite [10, págs. 428, 432].

Como é bem conhecido na literatura [8, Cap. 3], as palavras-código de um código cíclico podem ser representadas por meio de polinômios. Ou seja, uma palavra-código $\mathbf{c} = (c_0, c_1, c_2, \ldots, c_{n-1})$ pode ser representada pelo polinômio $c(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1}x^{n-1}$. Deslocamentos cíclicos realizados nas palavras-código também possuem uma representação polinomial simples. Por exemplo, a palavra-código obtida ao deslocar $\mathbf{c}$ de $s$ posições para a direita, onde $s$ denota um número inteiro não-negativo, é representada por $(c_{n-s}, c_{n-s+1}, \ldots, c_0, c_1, c_2, \ldots, c_{n-s+1})$ e a respectiva representação polinomial é denotada por

$$x^s c(x) = (c_0 + c_1 x + \ldots + c_{n-1}x^{n-1})x^s \bmod (x^n - 1). \quad (1)$$

Denotando por $g(x) = g_0 + g_1 x + \ldots + g_{n-k}x^{n-k}$ o polinômio gerador [8] do código cíclico $C$, a matriz $G$ geradora de $C$ pode ser escrita da seguinte forma.

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & & & \mathbf{0} \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & g_0 & g1 & \cdots & g_{n-k} & \\ \mathbf{0} & & & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}.$$

Como o polinômio gerador $g(x)$ de um código cíclico $C$ é fator de $x^n - 1$, para cada $g(x)$ existe um polinômio mônico de grau $k$, $h(x) = 1 + h_1 x + h_2 x^2 + \ldots + h_{k-1}x^{k-1} + x^k$, tal que $g(x)h(x) = x^n - 1$. Seja $h'(x)$ o polinômio recíproco de $h(x)$, i.e., $h'(x) = x^k h(1/x)$. Para cada código cíclico $C$ com polinômio gerador $g(x)$ existe um código dual $C'$ correspondente, com polinômio gerador $h'(x)$. Sabe-se que as $n-k$ linhas da matriz de verificação de paridade $H$ de um código cíclico $C$ podem ser obtidas por deslocamentos cíclicos de uma única palavra-código do código dual, por exemplo $h'(x)$. Portanto, a matriz $H$ pode ser escrita da seguinte forma.

$$H = \begin{bmatrix} 1 & h_{k-1} & \ldots & h_1 & 1 & 0 & 0 & \ldots \\ 0 & 1 & h_{k-1} & \ldots & h_1 & 1 & 0 & \ldots \\ & \ddots & \ddots & \ddots & \ddots & & & \\ 0 & 0 & \ldots & 1 & \ldots & & h_1 & 1 \end{bmatrix} \quad (2)$$

## III. Decodificação iterativa simplificada em duas dimensões sobre o BEC

Idealmente, devem ser usados decodificadores que determinam a máxima probabilidade a posteriori (MAP) para decodificar as palavras ou os bits recebidos [11, pág. 74]. A decodificação MAP de códigos lineares sobre o BEC pode ser realizada através da resolução de um sistema de equações lineares cujas incógnitas são os valores dos bits onde houve apagamento [11, pág. 73]. No entanto, quando o comprimento e o número de palavras-código aumentam torna-se proibitiva a utilização da decodificação MAP, visto que sua complexidade é cúbica com o comprimento das palavras-código [11, pág. 74].

Neste ponto é feita uma observação relevante para o desenvolvimento deste artigo, a qual permite reduzir significativamente a complexidade da decodificação iterativa em duas dimensões com permutação cíclica. Considerando o uso de códigos cíclicos idênticos nas duas dimensões de um código produto, observa-se de (2) que em várias situações de interesse prático, essencialmente apenas uma única equação de verificação de paridade precisa ser utilizada em todo o processo de decodificação, como será detalhado a seguir.

Para um arranjo binário bidimensional, $n \times n$, no qual as $n$ linhas e as $n$ colunas são palavras-código de um mesmo código cíclico $(n, k, d)$, a decodificação iterativa com permutação cíclica é realizada do seguinte modo. Começando a decodificação pelas linhas do arranjo, cada linha, a partir da primeira, é verificada quanto à ocorrência de apagamentos com o auxílio da equação de verificação de paridade definida pelo polinômio $h'(x)$, a qual é deslocada ciclicamente sobre a linha escolhida. A cada deslocamento, caso haja um único apagamento entre as posições verificadas por $h'(x)$ efetua-se a correção do mesmo. Caso ocorram dois ou mais apagamentos entre as posições verificadas por $h'(x)$, nada é feito e efetua-se mais um deslocamento cíclico de $h'(x)$ sobre a linha escolhida, continuando dessa forma até completar $n$ deslocamentos cíclicos. Procede-se da mesma forma sobre a segunda linha do arranjo bidimensional e assim sucessivamente até alcançar a linha $n$. A partir desse ponto o decodificador opera de modo análogo sobre as colunas, usando $h'(x)$ e seus deslocamentos cíclicos para verificar a ocorrência de apagamentos nas colunas e corrigí-los sempre que haja apenas um único apagamento entre as posições verificadas por $h'(x)$ naquele instante. Terminada a verificação das colunas, o decodificador volta a examinar as linhas do arranjo e assim sucessivamente. Numa nova passagem de $h'(x)$ e seus deslocamentos cíclicos pelas linhas do arranjo, por exemplo, é possível que vários padrões de apagamentos, não corrigidos anteriormente, possam vir a ser corrigidos agora pelo fato de que algumas de suas posições com apagamento tenham sido corrigidas na iteração anterior, aplicada às colunas do arranjo. Dessa forma, o decodificador resultante passa a ter uma baixa complexidade quantificada pelo número de iterações vezes o comprimento do código. Nos casos investigados, o número de iterações necessárias foi de aproximadamente $n/2$ até que um conjunto de parada fosse alcançado. Na seção seguinte são apresentados resultados experimentais, obtidos por simulação computacional usando vários algoritmos de decodificação iterativa para alguns códigos produto.

## IV. Resultados experimentais

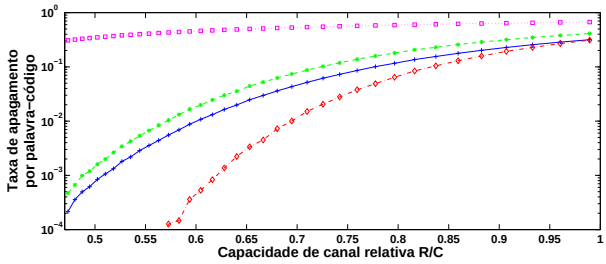
O desempenho da decodificação iterativa com deslocamento cíclico, como descrita na Seção III, foi comparado com três casos descritos a seguir. O primeiro caso considera um algoritmo clássico de decodificação baseado no limitante $\rho \leq d-1$, que corrige no máximo $d-1$ apagamentos por palavra e que é conhecido como decodificador *bounded-distance* [12]. O segundo caso considera o algoritmo *belief propagation* (BP).

O algoritmo BP foi introduzido por Gallager [13] para a decodificação de códigos LDPC. Por último, no terceiro caso, é feita uma comparação de desempenho do algoritmo proposto com aquele de um código artificial de distância máxima separável (MDS), que estabelece um limite teórico inferior para a taxa de erro por palavra-código (CER) alcançável com códigos de comprimento finito sobre o BEC [9].

Sobre o canal BEC, a decodificação em duas dimensões utilizando BP consiste em encontrar as equações de verificação de paridade, tanto nas linhas quanto nas colunas, que envolvam um único apagamento e então resolvê-las para recuperar o respectivo bit apagado [14]. O processo é repetido até que todos os bits sejam corrigidos ou até que todas as equações restantes envolvam pelo menos dois bits apagados. O algoritmo BP apresenta baixa complexidade para códigos com matriz de verificação de paridade esparsa. Por outro lado, ele apresenta uma complexidade maior para códigos com matriz de verificação de paridade densa, como exemplo os códigos BCH e Reed-Solomon. A fim de superar em parte as limitações do algoritmo BP, uma abordagem eficiente foi introduzida em [15] para códigos cíclicos. Essa abordagem consiste na aplicação de uma permutação cíclica, na linha ou na coluna, quando o algoritmo BP trava e não consegur ir adiante. A permutação cíclica altera os bits que participam das equações de verificação de paridade, aumentando, assim, a probabilidade de que o correspondente sistema de equações lineares tenha solução e assim permita a correção dos bits apagados. Em nenhum momento a referência [15] detalha a implementação da decodificação com permutação cíclica, deixando subentendido que foi utilizada a técnica de resolver um sistema de equações lineares, cujas incógnitas são as posições dos apagamentos. A seguir, são analisados os códigos testados e é feita uma comparação entres os vários algoritmos.

### A. *Código produto* $(49, 16, 9)$

A Fig. 1 mostra o comportamento da taxa de apagamento por palavra-código em função da variação da capacidade de canal relativa, isto é, da razão entre a taxa do código e a capacidade do canal. Com o código $(49, 16, 9)$ fixo, varia-se a probabilidade de apagamento do canal e, consequentemente, sua capacidade.




$\cdot\cdot\square\cdot\cdot$ *bounded-distance*, $-*-$ *belief propagation*, $-+-$ *belief propagation* com permutação cíclica, 4 iterações, $\cdot-\diamond-$ código virtual MDS $(49, 16, 34)$.

Fig. 1. CER para o código produto $(49, 16, 9)$ versus a capacidade de canal relativa $(R/C)$.

Pode-se observar na Fig. 1 que utilizando o algoritmo BP combinado com permutação cíclica obtém-se o melhor resultado. A decodificação iterativa utilizando esse algoritmo foi a que mais se aproximou do limite teórico conservador dado pelo código MDS.

Para investigar o desempenho do decodificador iterativo em duas dimensões, no que se refere à capacidade de correção de apagamentos, foi gerada aleatoriamente uma sequência de palavras-código e, a cada dez mil palavras-código geradas, a probabilidade de apagamento do canal era incrementada em $1\%$, fazendo esta variar de $1\%$ a $100\%$. A Fig. 2 mostra a distribuição do percentual de bits recuperados corretamente sem permutação cíclica (Fig. 2a) e com permutação cíclica (Fig. 2b).

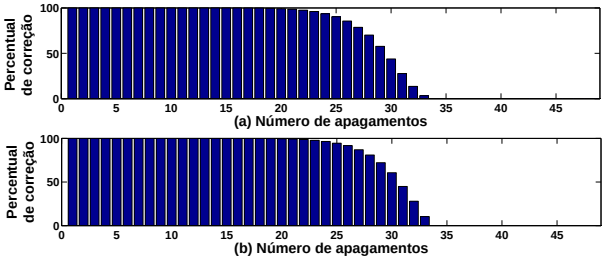
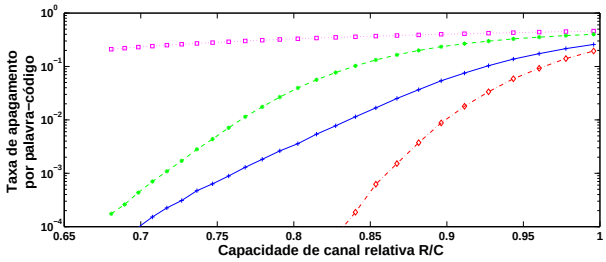



Fig. 2. Percentual de bits corretamente recuperados versus o o número de apagamentos em uma palavra-código recebida para o código produto $(49, 16, 9)$. a) sem permutação cíclica, b) com permutação cíclica.

Pode-se observar na Fig. 2 que o algoritmo BP sem permutação cíclica recuperou corretamente praticamente todos os bits, para todos os padrões de até $21$ apagamentos gerados pelo canal. Quando o algoritmo BP é utilizado com permutação cíclica, é possível corrigir praticamente todos os bits com padrões de até 22 apagamentos. Pode-se observar que esse valor é bem maior que o valor máximo garantido que é $d - 1 = 8$. Essa melhora em desempenho deve-se à decodificação iterativa em duas dimensões. Outra observação é que alguns padrões de até 33 apagamentos também foram corrigidos, sempre numa proporção maior quando se utiliza a decodificação BP combinada com permutação cíclica.

### B. *Código produto* $(225, 121, 9)$

A Fig. 3 mostra o comportamento da taxa de apagamento por palavra-código em função da variação da capacidade de canal relativa.




$\cdot\cdot\square\cdot\cdot$ *bounded-distance*, $-*-$ *belief propagation*, $-+-$ *belief propagation* com permutação cíclica, 7 iterações, $\cdot-\diamond-$ código virtual MDS $(225, 121, 105)$.

Fig. 3. CER para o código produto $(225, 121, 9)$ versus a capacidade de canal relativa $(R/C)$.

Para o código produto $((225, 121, 9)$ observa-se na Fig. 3 que, utilizando o algoritmo BP combinado com permutação cíclica, obtém-se o melhor resultado. A decodificação iterativa utilizando permutação cíclica foi a que mais se aproximou do limite teórico conservador dado pelo código MDS. A Fig. 4 mostra a distribuição do percentual de bits recuperados corretamente sem permutação cíclica (Fig. 4a) e com permutação cíclica (Fig. 4b).

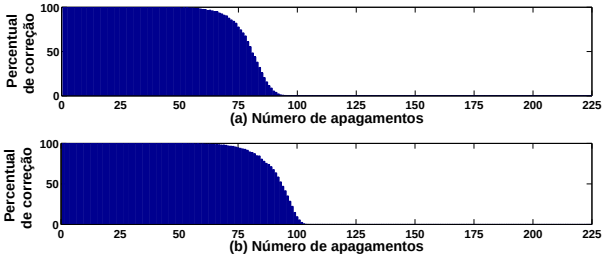
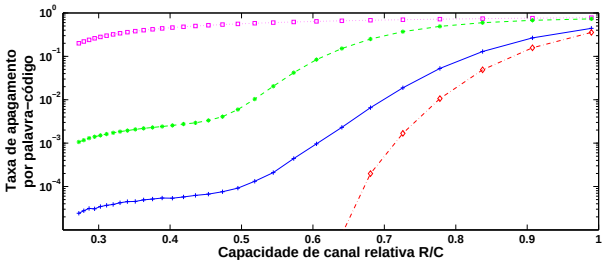



Fig. 4. Percentual de bits corretamente recuperados versus o o número de apagamentos em uma palavra-código recebida para o código produto $(225, 121, 9)$. a) sem permutação cíclica, b) com permutação cíclica.

Pode-se observar na Fig. 4 que, para padrões cujo número de apagamentos gerados pelo canal é próximo de 75, o decodificador recuperou praticamente todos os bits corretamente. Esse número de apagamentos corrigidos é bem maior que o máximo garantido, que é $d - 1 = 8$. Outra observação é que alguns padrões cujo número de apagamentos por palavra-código é muito alto, perto de 100 apagamentos, só foram corrigidos quando foi utilizada a permutação cíclica.

### C. *Código produto* $(225, 49, 25)$

O comportamento deste código, ilustrado na Fig. 5, difere dos anteriores pelo fato de que equações de verificação de paridade adicionais poderiam ser usadas e não o foram.




·· □ ·· *bounded-distance*, — ∗ — *belief propagation*, — + — *belief propagation* com permutação cíclica, 9 iterações, · — ◇ — código virtual MDS $(225, 49, 177)$.

Fig. 5. CER para o código produto $(225, 49, 25)$ versus a capacidade de canal relativa $(R/C)$.

Observa-se na Fig. 5 que o algoritmo BP combinado com permutação cíclica apresenta o melhor resultado. A Fig. 6 mostra a distribuição do percentual de bits recuperados corretamente sem permutação cíclica (Fig. 6a) e com permutação cíclica (Fig. 6b). Na Fig. 6 fica claro o ganho no desempenho de correção de apagamento quando utilizamos decodificação iterativa em duas dimensões usando permutação cíclica em comparação com a decodificação em duas dimensões sem utilizar permutação cíclica.

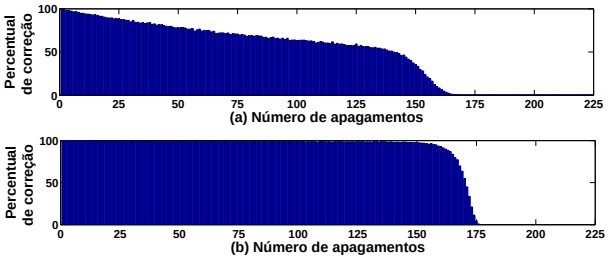



Fig. 6. Percentual de bits corretamente recuperados versus o o número de apagamentos em uma palavra-código recebida para o código produto $(225, 49, 25)$. a) sem permutação cíclica, b) com permutação cíclica.

## V. Conclusões

Neste artigo foi investigado um procedimento de baixa complexidade para implementação do algoritmo BP com permutação cíclica em canais BEC. Foi mostrado que é possível realizar a decodificação iterativa em duas dimensões utilizando apenas uma única equação de verificação de paridade e seus deslocamentos cíclicos nas linhas e nas colunas do arranjo, com um desempenho satisfatório. Para códigos de Hamming idênticos nas duas dimensões este procedimento é ótimo, no sentido de que todas as possíveis equações de verificação de paridade são empregadas.

## Referências

[1] P. Elias, "Coding for two noisy channels", *in Proc. 3rd London Symp. Information Theory*, London, U.K., 1955, pp. 61 - 76.

[2] R. E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press, 2003.

[3] S. Changuel, R. Le Bidan and R. Pyndiah, "Iterative decoding of product codes over binary erasure channel", *Electronics Letters*, vol.46 , no. 7, 2010, pp. 503 - 505.

[4] E. Rosnes and O. Ytrehus, "Turbo decoding on the binary erasure channel: Finite-length analysis and turbo stopping sets", *IEEE Trans. on Information Theory*, Vol. 53, No. 11, novembro de 2007, pp. 4059 - 4075.

[5] P. Elias, "Error-free coding", *IRE Trans. Inform. Theory*, vol. IT-16, setembro de 1970, pp. 624 - 627.

[6] R. M. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes", *IEEE Trans. on Commun.*, vol. 46, agosto de 1998, pp. 1003 - 1010.

[7] M. Schwartz, P. H. Siegel and A. Vardy, "On the asymptotic performance of iterated subcode-decoding of product codes", *IEEE Int. Symp. on Information Theory*, ISIT2005, Adelaide, Austrália, setembro de 2005, pp. 1758-1762.

[8] S. Lin and D. J. Costello, Jr., *Error Control Coding*, Second Edition, Pearson, Prentice Hall, New Jersey, USA, 2004.

[9] M. Tomlinson, C. Tjhai, J. Cai and M. Ambroze, "Analysis of the distribution of the number of erasures correctable by a binary linear code and the link to low-weight codewords", *IET Commun.*, vol.1, no. 3, junho de 2007, pp. 539 - 548.

[10] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, 1995.

[11] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, New York, USA, 2008.

[12] N. Sendrier, "Codes correcteurs d'erreurs à haut pouvoir de correction", *Ph.D. Thesis*, Université Paris VI, Dec. 1991.

[13] R. G. Gallager, "Low-density parity-check codes", *IRE Trans. Inform. Theory*, vol. 8,no. 1, janeiro de 1962, pp. 21 - 28, .

[14] M. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. A. Spielman, D. A., "Efficient erasure correcting codes", *IEEE Trans. Inform. Theory*, vol. 47, no. 2, fevereiro de 2001, pp. 569 - 584.

[15] T. Hehn, O. Milenkovic, S. Laendner and J. B. Huber, "Permutation decoding and the stopping redundancy hierarchy of linear block codes", *Proc IEEE Int. Symp. Inform. Theory ISIT'07*, France, junho de 2007, pp. 2926 - 2930.