

# Uma Avaliação do Algoritmo de Chase-2 Generalizado Aplicado a Códigos de Bloco com Proteção Desigual de Erros

Rebecca C. de Albuquerque<sup>1</sup>, Daniel C. Cunha<sup>1</sup> e Cecilio Pimentel<sup>2</sup>

**Resumo**—Este trabalho realiza uma avaliação, através de simulações computacionais, do algoritmo de Chase-2 generalizado aplicado a códigos de bloco com proteção desigual de erros usando transmissão binária em um canal com ruído aditivo Gaussiano branco. O desempenho do algoritmo de Chase-2 generalizado é comparado com o algoritmo de decodificação por máxima verossimilhança e uma análise do compromisso entre desempenho e complexidade do algoritmo de Chase-2 é realizada para cada classe de proteção.

**Palavras-Chave**—Proteção desigual de erros, algoritmo de Chase, complexidade de decodificação.

**Abstract**—This work evaluates, through computer simulations, the effectiveness of the generalized Chase-2 algorithm applied to unequal error protection block codes using binary transmission over an additive white Gaussian noise channel. The performance of the generalized Chase-2 algorithm is compared to maximum likelihood decoding algorithm and an analysis of the tradeoff between performance and complexity of this algorithm for each class of protection is provided.

**Keywords**—Unequal error protection, Chase algorithm, decoding complexity.

## I. INTRODUÇÃO

A transmissão de sinais digitais, como por exemplo, imagem e vídeo, por meio da *Internet* e através das redes sem fio, fixas ou móveis, tem crescente importância prática. Uma técnica efetiva para esta transmissão consiste em oferecer níveis de proteção distintos a diferentes partes dos dados de informação. A esta técnica dá-se o nome de codificação com proteção desigual de erros (UEP, do inglês *unequal error protection*) [1], [2]. O emprego de técnicas UEP tem sido alvo de investigação da comunidade científica nos últimos anos, seja na construção de códigos corretores de erros [3], como na proposição de esquemas de modulação hierárquica aplicados a sistemas de comunicação cooperativa [4], [5].

Uma vez que a decodificação de máxima verossimilhança (MV) possui complexidade de ordem exponencial, torna-se interessante a investigação de outras técnicas de decodificação aplicadas a códigos com proteção desigual de erros em busca de alternativas menos complexas. Em [6], foram propostas modificações no algoritmo *maximum a posteriori*

(MAP) aplicado a códigos turbo convolucionais UEP. Em relação aos códigos de bloco UEP, uma categoria de algoritmos que merece atenção é formada pelos algoritmos de decodificação por decisão suave baseados em confiabilidade (probabilísticos) [7]. Nessa categoria, o algoritmo de Chase e suas variantes se destacam pela facilidade de implementação, menor complexidade em relação ao algoritmo MV e pela aplicação em códigos produto [8].

Este trabalho realiza uma avaliação, através de simulações computacionais, de um algoritmo de decodificação por decisão suave (algoritmo de Chase-2 generalizado [9]) para cada classe de proteção de códigos de bloco UEP com transmissão binária em um canal com ruído aditivo Gaussiano branco. O desempenho do algoritmo de Chase-2 generalizado é comparado com o algoritmo de decodificação MV com o objetivo de avaliar a perda de desempenho de cada classe de proteção proporcionada por um algoritmo menos complexo. Adicionalmente, é realizada uma análise do compromisso entre desempenho e complexidade do algoritmo de Chase-2 generalizado para cada classe de proteção. Na seção II, conceitos relativos à codificação UEP são descritos. O algoritmo de Chase-2 generalizado é descrito na Seção III, assim como a análise de sua complexidade de decodificação em termos de operações matemáticas. Na Seção IV, resultados de simulação são apresentados. Por fim, a seção V apresenta as conclusões do trabalho.

## II. CÓDIGOS COM PROTEÇÃO DESIGUAL DE ERROS

Considere um código linear binário  $C_j(n, k, d)$  em que  $n$  é o comprimento das palavras-código,  $k$  é o tamanho do vetor de informação e  $d$  é a distância de Hamming mínima do código. A matriz geradora do código  $C_j(n, k, d)$  é denotada por  $\mathbf{G}_j$ . Assumindo que  $w(\mathbf{uG}_j)$  é o peso de Hamming da palavra-código  $\mathbf{x} = \mathbf{uG}_j$  relativa ao vetor de informação  $\mathbf{u}$ , define-se o vetor de separação  $\mathbf{s}_j = [s_j^1, \dots, s_j^i, \dots, s_j^k]$  do código  $C_j(n, k, d)$  de maneira que a  $i$ -ésima posição de  $\mathbf{s}_j$  é dada por [10]

$$s_j^i = \min\{w(\mathbf{uG}_j) : \mathbf{u} \in GF(2)^k, u_i \neq 0\}, i = 1, \dots, k, \quad (1)$$

em que  $GF(2)$  é o corpo de Galois binário. O menor elemento de  $\mathbf{s}_j$  é a distância mínima do código. Um código  $C_j$  é dito ter proteção igual de erros se todos os elementos de  $\mathbf{s}_j$  forem iguais, caso contrário, o código possui a propriedade UEP.

O vetor de separação  $\mathbf{s}_j$  mede a proteção desigual de erros fornecida pelo código  $C_j$  para a decodificação MV. A

<sup>1</sup> Núcleo de Pesquisa em Telecomunicações, POLI, Universidade de Pernambuco (UPE), Recife-PE, Brasil.

<sup>2</sup> Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco (UFPE), Recife-PE, Brasil.

E-mails: rebeccacamile@upe.poli.br, dccunha@upe.poli.br e cecilio@ufpe.br. Este trabalho foi parcialmente financiado pelo PFA-2010/UPE, pela FACEPE (APQ-1060-3.04/10) e pelo CNPq (302535/2010-1).

decodificação MV busca a palavra-código  $\mathbf{x}$  com distância Euclidiana quadrática mínima em relação à sequência  $\mathbf{r}$ , considerando o uso de modulação binária. Isto equivale a selecionar o vetor  $\mathbf{x}$  que possui máxima correlação em relação a sequência  $\mathbf{r}$ , definida por

$$c(\mathbf{r}, \mathbf{x}) = \sum_{i=1}^n r_i \cdot x_i. \quad (2)$$

A decodificação correta da  $i$ -ésima posição do vetor  $\mathbf{u}$  ocorrerá caso menos de  $\lfloor \frac{s_j^i - 1}{2} \rfloor$  erros ocorram na palavra transmitida. Uma vez que cada posição do vetor de informação possui um nível de proteção individual, é possível agrupar as posições que possuem o mesmo nível de proteção em uma determinada classe de proteção. Além disso, é possível estabelecer o vetor de capacidade de correção  $\mathbf{t}_j$  do código UEP  $C_j$  com base no vetor de separação  $\mathbf{s}_j$ . Inicialmente, define-se o vetor  $\mathbf{h}_j = [h_j^1, \dots, h_j^k]$  de maneira que a  $i$ -ésima posição de  $\mathbf{h}_j$  é dada por

$$h_j^i = \lfloor \frac{s_j^i - 1}{2} \rfloor, i = 1, \dots, k. \quad (3)$$

Em seguida, define-se o vetor de capacidade de correção  $\mathbf{t}_j = [t_j^1, \dots, t_j^m]$ , em que  $m$  é o número de elementos distintos de  $s_j$  e representa o número de classes de proteção distintas do código  $C_j$ . O menor elemento de  $\mathbf{t}_j$  é a capacidade de correção do código  $C_j(n, k, d)$  e será denotada por  $t_j^*$ .

Para ilustrar os conceitos apresentados, considere o código de bloco linear  $C_a(10, 5, 3)$ , cuja matriz geradora  $\mathbf{G}_a$  é dada por

$$\mathbf{G}_a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (4)$$

e que possui vetor de separação  $\mathbf{s}_a = [5, 3, 3, 3, 3]$ . O primeiro bit de informação possui um nível de proteção  $s_a^1 = 5$ , enquanto os demais bits de informação possuem nível de proteção  $s_a^j = 3, j = 2, \dots, 5$ . Desta maneira, podemos afirmar que  $C_a(10, 5, 3)$  é um código UEP que possui duas classes de proteção distintas, denotadas por  $\mathcal{C}_{P_1}$  (classe de maior proteção) e  $\mathcal{C}_{P_2}$  (classe de menor proteção), vetor de capacidade de correção  $\mathbf{t}_a = [2, 1]$  e capacidade de correção  $t_a^* = 1$ .

### III. ALGORITMO DE CHASE-2 GENERALIZADO

Neste trabalho, o algoritmo de decodificação por decisão suave considerado é o algoritmo de Chase-2 generalizado [9], técnica de decodificação que lida com as posições menos confiáveis da sequência recebida  $\mathbf{r}$ . Por questões de simplificação, daqui por diante, o algoritmo de Chase-2 generalizado será denominado apenas de algoritmo CG-2.

O algoritmo CG-2 utiliza a sequência de valores reais observados na saída do canal (saídas dos filtros casados do receptor), denotada por  $\mathbf{r} = [r_1, r_2, \dots, r_n]$ , e a palavra binária  $\mathbf{y}$  gerada pela quantização abrupta da sequência  $\mathbf{r}$ . Para o caso do canal com ruído aditivo Gaussiano branco, os valores reais da sequência  $\mathbf{r}$  correspondem aos valores

de confiabilidade  $\alpha_i$  do algoritmo, tal que  $\alpha_i = |r_i|$ . Desta forma, quanto maior o valor de  $\alpha_i$ , menor a probabilidade de o símbolo correspondente ter sido afetado fortemente pelo ruído.

O algoritmo CG-2 é completamente especificado pelos parâmetros  $t$ , número máximo de erros que o decodificador binário empregado é capaz de corrigir, e  $p$ , número de posições menos confiáveis da sequência recebida  $\mathbf{r}$ , ou seja, as posições que possuem os menores valores de  $\alpha_i$  [11]. O valor de  $p$  determina o conjunto de padrões de teste  $\mathbf{b}$ , cuja cardinalidade vale  $|S_b| = 2^p$ . Inicialmente, o algoritmo CG-2 emprega um decodificador binário para decodificar a sequência binária  $\mathbf{y} \oplus \mathbf{b}$ , em que  $\oplus$  representa a operação de soma módulo-2. Caso um padrão de erro  $\mathbf{z}$  seja obtido na decodificação binária<sup>1</sup>, este é adicionado ao padrão de teste  $\mathbf{b}$ , originando o padrão  $\mathbf{z}^b = \mathbf{z} \oplus \mathbf{b}$ . Em seguida, o peso analógico  $W_\alpha$  do padrão  $\mathbf{z}^b$  é calculado segundo a equação

$$W_\alpha(\mathbf{z}^b) = \sum_{i=1}^n \alpha_i z_i^b. \quad (5)$$

Caso o padrão de erro  $\mathbf{z}$  não seja encontrado na decodificação binária, o próximo padrão de teste  $\mathbf{b}$  é selecionado. O objetivo do algoritmo CG-2 é selecionar o padrão  $\mathbf{z}^b$  com menor peso analógico  $W_\alpha$  para estimar a palavra-código transmitida  $\mathbf{x}$  a partir da sequência  $\mathbf{y}$ . Quando um padrão  $\mathbf{z}^b$  não é selecionado, a sequência  $\mathbf{y}$  é assumida como estimativa da palavra transmitida  $\mathbf{x}$ .

Uma descrição detalhada do algoritmo CG-2 pode ser obtida em [9], [11] e um resumo das etapas que compõem o algoritmo CG-2 é descrito na Tabela I.

TABELA I  
DESCRIBÇÃO DAS ETAPAS DO ALGORITMO DE CHASE-2 GENERALIZADO.

Etapa	Descrição
1	Obter a sequência $\mathbf{y}$ a partir da sequência $\mathbf{r}$ . Gerar um padrão de teste $\mathbf{b}$ .
2	Selecionado um padrão de teste $\mathbf{b}$ , obter a sequência $\mathbf{y} \oplus \mathbf{b}$ .
3	Buscar um padrão de erro $\mathbf{z}$ associado à sequência $\mathbf{y} \oplus \mathbf{b}$ (decodificação binária).
4	Caso $\mathbf{z}$ seja encontrado, obter o padrão $\mathbf{z}^b$ e calcular seu peso analógico $W_\alpha$ . Armazenar o padrão $\mathbf{z}^b$ de peso analógico mínimo. Caso $\mathbf{z}$ não seja encontrado e ainda haja padrões de teste a gerar, selecionar outro padrão $\mathbf{b}$ e voltar à etapa 2.
5	Caso todos os padrões de teste tenham sido gerados e um padrão $\mathbf{z}^b$ tenha sido armazenado, obter a estimativa $\hat{\mathbf{x}} = \mathbf{y} \oplus \mathbf{z}^b$ . Caso contrário, $\hat{\mathbf{x}} = \mathbf{y}$ .

#### A. Análise de Complexidade

Um aspecto relevante na avaliação de algoritmos de decodificação é o estabelecimento do compromisso entre desempenho e complexidade. A complexidade do algoritmo

<sup>1</sup>Padrão de erro  $\mathbf{z}$  associado à síndrome da sequência  $\mathbf{y} \oplus \mathbf{b}$ .

CG-2 pode ser avaliada em função do número de operações matemáticas executadas em cada etapa do algoritmo. Considere  $N_s$ ,  $N_g$ ,  $N_m$  e  $N_c$ , o número de somas, somas módulo-2, multiplicações e comparações, respectivamente. A Tabela II indica o número de operações matemáticas realizadas por padrão de teste **b** para as etapas 2,3 e 4 do algoritmo CG-2, descritas na Tabela I.

TABELA II

NÚMERO DE OPERAÇÕES MATEMÁTICAS REALIZADAS NO ALGORITMO CG-2 POR PADRÃO DE TESTE **b**.

Etapa	$N_s$	$N_g$	$N_m$	$N_c$
2	-	$n$	-	-
3	-	$(n-k)(n-1)$	$(n-k)n$	-
4	$(n-1)$	$n$	$n$	1

Cabe ressaltar que as operações realizadas na etapa 4 dependem do resultado obtido na etapa 3, ou seja, dependem do êxito do decodificador binário na busca por um padrão de erro  $\mathbf{z}$  associado à sequência  $\mathbf{y} \oplus \mathbf{b}$ . Desta forma, é necessário estimar o valor médio das operações executadas na etapa 4 do algoritmo CG-2. Para isso, define-se a frequência relativa de execução de cálculo de  $W_\alpha$  por  $f_A = N_W/2^p$ , em que  $N_W$  é o número de vezes que o peso analógico  $W_\alpha$  é calculado.

#### IV. RESULTADOS NUMÉRICOS

Foram realizadas simulações computacionais do algoritmo de decodificação MV e do algoritmo CG-2 para os códigos de bloco UEP  $C_1(16, 5, 5)$  e  $C_2(25, 8, 5)$ , cujos vetores de separação são  $\mathbf{s}_1 = [8, 8, 5, 5, 5]$  e  $\mathbf{s}_2 = [12, 12, 5, 5, 5, 5, 5, 5]$ , respectivamente. Desta forma, os vetores de capacidade de correção para os códigos  $C_1$  e  $C_2$  são  $\mathbf{t}_1 = [3, 2]$  e  $\mathbf{t}_2 = [5, 2]$ . As regras de construção das matrizes geradoras para os códigos  $C_1$  e  $C_2$  foram obtidas em [12]. Foi considerado o uso de modulação binária e canal com ruído aditivo Gaussiano branco. Para cada classe de proteção, curvas de probabilidade de erro de bit ( $P_b$ ) versus relação sinal-ruído (RSR)  $E_b/N_0$  foram obtidas, em que  $E_b$  é a energia por bit de informação e  $N_0$  é a densidade espectral de potência do ruído Gaussiano.

Várias configurações do algoritmo CG-2 foram testadas, por meio da mudança de seus parâmetros ( $t$  e  $p$ ), com o objetivo de compará-las com o desempenho do algoritmo MV. O parâmetro  $t$  variou de 1 até  $(t_j^1+1)$ , ou seja, até o máximo valor dos elementos de  $\mathbf{t}_j$  acrescido de 1. Em relação ao conjunto de padrões de teste  $S_b$ , foi assumido um valor máximo de  $p$  tal que  $|S_b|$  fosse sempre menor do que o conjunto de busca do algoritmo MV ( $2^k - 1$  palavras-código).

A Figura 1 ilustra o desempenho dos algoritmos MV e CG-2( $t, p$ ) para as duas classes do código UEP  $C_1(16, 5, 5)$ . Para o algoritmo CG-2, estão representados os casos CG-2(2, 2) e CG-2(3, 4). No algoritmo CG-2(2, 2), é possível observar que praticamente não houve diferença de desempenho entre as classes  $cp_1$  e  $cp_2$ . Ainda para o algoritmo CG-2(2, 2) e considerando  $P_b = 10^{-4}$ , a diferença de RSR em relação ao algoritmo MV foi de aproximadamente 2 dB e 1,1 dB para as classes  $cp_1$  e  $cp_2$ , respectivamente. Já para o algoritmo

CG-2(3, 4), a diferença de RSR para o algoritmo MV foi de 0,12 dB ( $cp_1$ ) e 0,03 dB ( $cp_2$ ). Para a classe  $cp_2$ , o algoritmo CG-2(3, 4) utilizou o parâmetro  $t$  acima da capacidade de correção da classe ( $t_1^2 = 2$ ), o que pode justificar o desempenho bastante próximo ao do algoritmo MV. Enquanto isso, para a classe  $cp_1$ , o parâmetro  $t$  do algoritmo CG-2(3, 4) não excedeu o valor máximo inicialmente estipulado, o que indica que é possível avançar um pouco mais em desempenho.

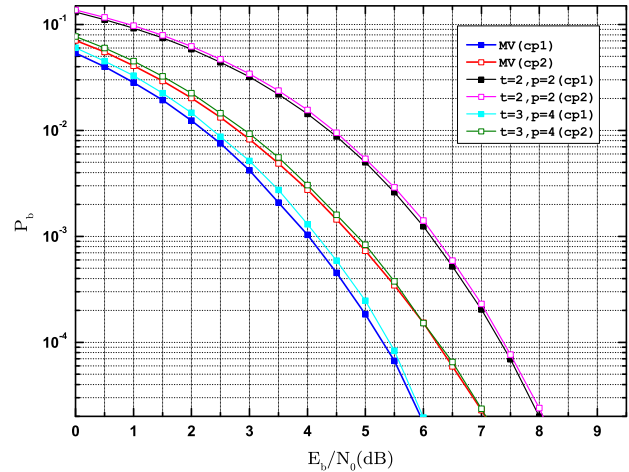


Fig. 1. Desempenho dos algoritmos de decodificação MV e CG-2( $t, p$ ) para o código UEP  $C_1(16, 5, 5)$  considerando o uso de modulação binária e canal com ruído aditivo Gaussiano branco.

A Figura 2 ilustra o desempenho dos algoritmos MV ( $cp_1$  e  $cp_2$ ), CG-2(2, 2) ( $cp_1$  e  $cp_2$ ), CG-2(4, 7) ( $cp_1$ ) e CG-2(3, 7) ( $cp_2$ ) para o código UEP  $C_2(25, 8, 5)$ . Para o algoritmo CG-2(2, 2), nota-se que, apesar da classe  $cp_1$  possuir maior nível de proteção ( $t_1^2 = 5$ ), o desempenho da classe  $cp_2$  foi melhor em toda a faixa de RSR analisada. Uma possível justificativa pode estar no número de bits de informação em cada classe (dois na primeira e seis na segunda). Assumindo mais uma vez  $P_b = 10^{-4}$ , a diferença de RSR do algoritmo CG-2(2, 2) para o algoritmo MV foi de aproximadamente 4 dB e 1,9 dB para as classes  $cp_1$  e  $cp_2$ , respectivamente. Para o algoritmo CG-2(4, 7) ( $cp_1$ ), esta diferença atingiu 0,17 dB, enquanto para o algoritmo CG-2(3, 7) ( $cp_2$ ), este valor foi de 0,03 dB. Assim como no código  $C_1$ , o uso de  $t = 3$  permitiu que o desempenho do algoritmo CG-2(3, 7) ( $cp_2$ ) se aproximasse bastante do desempenho do algoritmo MV, porém, para  $E_b/N_0 < 5$  dB, esta aproximação não ocorreu de maneira semelhante ao caso do algoritmo CG-2(3, 4) ( $cp_2$ ) aplicado ao código  $C_1$  (vide Figura 1).

Para avaliação da complexidade, foi necessário estimar a frequência relativa de cálculos de  $W_\alpha$ , conforme mencionado na subseção III-A. A Figura 3 ilustra os valores de  $f_A$  em função de  $E_b/N_0$  para os algoritmos CG-2(2, 2), CG-2(2, 7), CG-2(3, 2), CG-2(3, 7), CG-2(6, 2) e CG-2(6, 7) aplicados ao código  $C_2$ . Para  $p = 2$ , as curvas de  $f_A$ , considerando  $t = 2, 3$  e 6, atingiram seus valores máximos (cálculo de  $W_\alpha$  em todos os padrões de teste **b** utilizados) em  $E_b/N_0 = 9$  dB, 7,5 dB e 4 dB, respectivamente. A redução de RSR ocorreu devido a maior possibilidade de um padrão de erro  $\mathbf{z}$  ser encontrado, consequência do aumento do número máximo de erros que o decodificador binário pode corrigir.

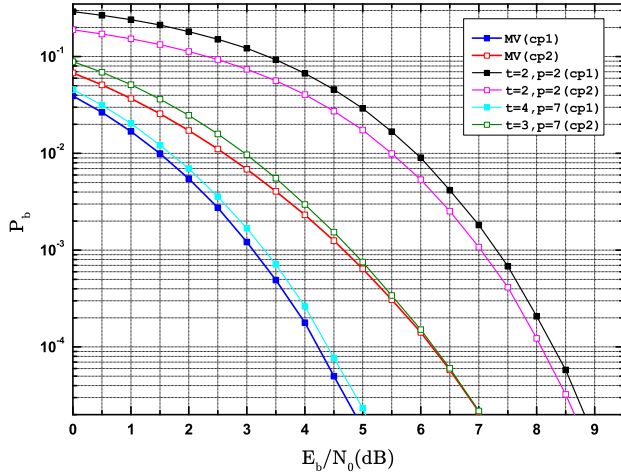


Fig. 2. Desempenho dos algoritmos de decodificação MV e CG-2( $t, p$ ) para o código UEP  $C_2(25, 8, 5)$  considerando o uso de modulação binária e canal com ruído aditivo Gaussiano branco.

Mantendo-se  $t$  fixo, o aumento de  $p$  provocou uma diminuição de  $f_A$  e isto foi notado em praticamente todos os casos apresentados. Todavia, para avaliar mais detalhadamente o comportamento da etapa 4 do algoritmo CG-2 quanto à variação de  $p$ , a diferença de  $f_A$  ( $\Delta f_A$ ) entre os algoritmos CG-2( $t, 2$ ) e CG-2( $t, 7$ ) foi observada em função da RSR. Com base na Figura 3 e assumindo  $t = 2$ , o aumento de  $p$  fez com que  $\Delta f_A$  aumentasse com o crescimento da RSR. Isto também ocorreu para o caso  $t = 3$ , porém de forma menos acentuada em comparação ao caso  $t = 2$ . Finalmente, para  $t = 6$ , o aumento de  $p$  alterou o comportamento de  $\Delta f_A$ , que passou a diminuir com o aumento da RSR. Com isso, foi possível notar que para  $t = 2$  e  $3$ , houve padrões de teste  $\mathbf{b}$  que não ocasionaram cálculos de  $W_\alpha$  ( $f_A < 1$ ), mesmo em regiões de alta RSR ( $E_b/N_0 > 7,5$  dB). Por exemplo, no caso do algoritmo CG-2( $2, 7$ ) e considerando  $E_b/N_0 > 7,5$  dB, é muito provável que os erros ocorridos na sequência  $\mathbf{y}$  tenham sido gerados pelas inversões de bit provocadas pela adição do padrão de teste  $\mathbf{b}$ . Como o decodificador binário utilizado foi capaz de corrigir apenas 2 erros, ocorreram 31 cálculos de  $W_\alpha(N_W = 31)$ , o que representa a soma de todos os padrões de teste de peso menor do que 2, resultando em  $f_A = 31/128 \approx 0,242$ .

Por fim, foi analisado o compromisso entre desempenho e complexidade dos algoritmos CG-2 em termos da diferença de RSR para o algoritmo MV (em cada classe de proteção) e do número de operações matemáticas (OM) para se atingir o desempenho mencionado. Para o cálculo de OM, foram consideradas todas as operações referentes às etapas 2, 3 e 4 do algoritmo CG-2 (vide Tabela II). No caso da etapa 4, foi necessário determinar o valor de  $f_A$  utilizado para ponderar o número de operações executadas. Estabelecido o algoritmo CG-2, foi determinado o valor da RSR correspondente a  $P_b = 10^{-4}$ . Com o valor de  $E_b/N_0$  obtido, identificou-se o valor de  $f_A$  correspondente (vide Figura 3). As operações referentes às etapas 1 e 5 foram omitidas por representarem um percentual muito pequeno do valor total de operações (no máximo 2%, no caso do algoritmo CG-2( $1, 2$ ) aplicado a  $C_1$ ).

As Tabelas III e IV resumem a avaliação realizada para

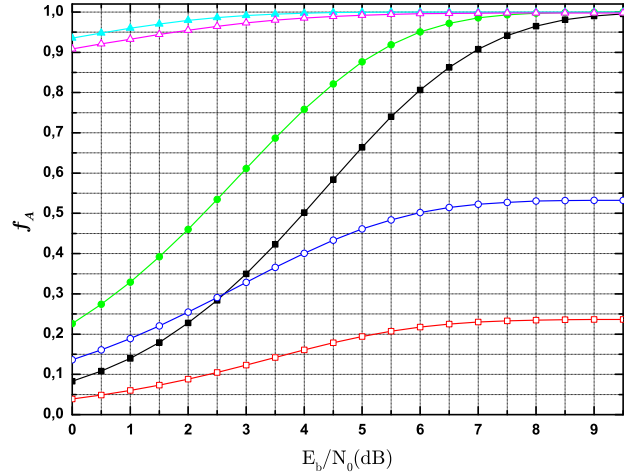


Fig. 3. Frequência relativa de execução do cálculo do peso analógico  $W_\alpha$  no algoritmo CG-2( $t, p$ ) aplicado ao código  $C_2(25, 8, 5)$  em função da RSR. Parâmetros indicados: ■ :  $t = 2, p = 2$ ; □ :  $t = 2, p = 7$ ; ● :  $t = 3, p = 2$ ; ○ :  $t = 3, p = 7$ ; ▲ :  $t = 6, p = 2$ ; △ :  $t = 6, p = 7$ .

diversas configurações do algoritmo CG-2 aplicado aos códigos  $C_1$  e  $C_2$ , respectivamente. Para cada interseção de uma linha ( $t$ ) com uma coluna ( $p$ ), dois valores são exibidos, quais sejam,  $\Delta_i$ (dB), a diferença de RSR em relação ao algoritmo MV (para  $P_b = 10^{-4}$ ), e o coeficiente  $OM_i$  (fundo cinza), ambos referentes à classe  $cp_i$ . Para ambos os códigos, os resultados apresentados indicam que, apesar do aumento de  $p$  (para um  $t$  fixo) proporcionar uma melhoria de desempenho e reduzir o valor de  $f_A$  (para alguns valores de  $t$ ), o aumento de complexidade associado cresce bastante, uma vez que  $|S_b|$  aumenta exponencialmente. Por outro lado, o aumento de  $t$  (para um  $p$  fixo) também ocasionou melhoria de desempenho, porém com um custo associado (aumento de complexidade) menor. Por exemplo, considerando a classe  $cp_1$  do código  $C_1$ , foi possível atingir  $\Delta_1 = 0,95$  dB utilizando o algoritmo CG-2( $3, 2$ ) com  $OM_1 \approx 1.618$ . Por outro lado, o algoritmo CG-2( $2, 4$ ) forneceu  $\Delta_1 = 0,85$  dB e necessitou da execução de aproximadamente 6.223 operações. Já para a classe  $cp_2$  do código  $C_2$ , os algoritmos CG-2( $4, 2$ ) e CG-2( $2, 5$ ) forneceram praticamente o mesmo  $\Delta_1$  (0,63 dB e 0,62 dB, respectivamente), porém com complexidades bem diferentes. Enquanto no primeiro caso foram executadas aproximadamente 3.731 operações nas etapas intermediárias, no segundo foram necessárias em torno de 28.598 operações.

Desta forma, os resultados apresentados nas Tabelas III e IV indicam que, para estabelecer um ponto de operação ótimo para o algoritmo CG-2, o aumento do número de erros que o decodificador binário é capaz de corrigir ( $t$ ) se mostra mais eficiente do que o aumento no conjunto de padrões de teste ( $|S_b| = 2^p$ ) no que se refere ao compromisso entre desempenho e complexidade.

## V. CONCLUSÕES

Este trabalho considerou a avaliação do algoritmo de Chase-2 generalizado para cada classe de proteção de códigos de bloco UEP com transmissão binária em um canal com ruído aditivo Gaussiano branco. Foi verificado o desempenho do algoritmo de Chase-2 frente ao algoritmo de máxima verossimilhança. Adicionalmente, foi investigado o

TABELA III  
 RESULTADOS DE DESEMPENHO E COMPLEXIDADE DOS ALGORITMOS CG-2( $t, p$ ) APLICADOS AO CÓDIGO  $C_1(16, 5, 5)$ .

	$t$	$p = 2$		$p = 3$		$p = 4$	
		$\Delta_i(\text{dB})$	$\text{OM}_i$	$\Delta_i(\text{dB})$	$\text{OM}_i$	$\Delta_i(\text{dB})$	$\text{OM}_i$
cp <sub>1</sub>	$t = 1$	3,30	1.569,50	2,66	3.044,16	2,12	5.947,01
	$t = 2$	2,03	1.614,62	1,42	3.180,48	0,85	6.222,72
	$t = 3$	0,95	1.618,27	0,46	3.229,25	0,11	6.411,65
	$t = 4$	0,65	1.620	0,17	3.238,85	0,10	6.476,16
cp <sub>2</sub>	$t = 1$	2,42	1.569,70	1,72	3.044,16	1,28	5.947,78
	$t = 2$	1,15	1.614,82	0,65	3.182,40	0,28	6.231,47
	$t = 3$	0,47	1.619,04	0,13	3.233,47	0,03	6.426,24
	$t = 4$	0,40	1.620	0,10	3.239,62	0,00	6.477,70

TABELA IV  
 RESULTADOS DE DESEMPENHO E COMPLEXIDADE DOS ALGORITMOS CG-2( $t, p$ ) APLICADOS AO CÓDIGO  $C_2(25, 8, 5)$ .

	$t$	$p = 2$		$p = 4$		$p = 5$		$p = 7$	
		$\Delta_i(\text{dB})$	$\text{OM}_i$	$\Delta_i(\text{dB})$	$\text{OM}_i$	$\Delta_i(\text{dB})$	$\text{OM}_i$	$\Delta_i(\text{dB})$	$\text{OM}_i$
cp <sub>1</sub>	$t = 1$	5,16	3.653,40	4,15	14.095,20	3,79	27.897,60	3,12	110.409,60
	$t = 2$	4,1	3.724,20	3,09	14.517,60	2,66	28.603,20	1,88	111.916,80
	$t = 3$	3,08	3.729	2,12	14.785,20	1,65	29.260,80	0,85	114.278,40
	$t = 4$	2,16	3.730,20	1,19	14.872,80	0,76	29.604	0,17	116.649,60
	$t = 5$	1,32	3.731,10	0,47	14.905,20	0,31	29.776,80	0,04	118.452,40
	$t = 6$	1,03	3.732	0,28	14.924,40	0,20	29.846,40	0,03	119.289,60
cp <sub>2</sub>	$t = 1$	2,94	3.651,90	1,97	14.092,80	1,59	27.892,80	1,01	110.409,60
	$t = 2$	1,88	3.722,10	0,97	14.511,60	0,62	28.598,40	0,21	111.974,40
	$t = 3$	1,05	3.728,70	0,35	14.798,40	0,14	29.320,80	0,04	114.691,20
	$t = 4$	0,63	3.731,10	0,15	14.905,20	0,06	29.731,20	0,00	117.408,00
	$t = 5$	0,55	3.732	0,14	14.924,40	0,02	29.841,60	0,00	118.934,40
	$t = 6$	0,55	3.732	0,14	14.924,40	0,02	29.841,60	0,00	118.934,40

comportamento do algoritmo de Chase-2 em relação ao cálculo do peso analógico (etapa 4) quanto à variação dos parâmetros do algoritmo. Por fim, uma análise do compromisso entre desempenho e complexidade do algoritmo de Chase-2 foi realizada para cada classe de proteção para diversas opções de configuração do algoritmo. Em relação ao compromisso entre desempenho e complexidade, a alteração do número de erros que o decodificador binário empregado é capaz de corrigir se mostrou mais vantajosa do que o aumento do número de padrões de teste do algoritmo de Chase-2.

REFERÊNCIAS

[1] M. Trott, "Unequal Error Protection Codes: Theory and Practice". in *Proc. of the IEEE Inf. Theory Workshop (ITW 1996)*, Haifa, Israel, 1996.

[2] S. Borade, B. Nakiboglu and L. Zheng, "Unequal Error Protection: An Information-Theoretic Perspective". *IEEE Trans. Inf. Theory*, vol. 55, n. 12, pp. 5511-5539, Dec 2009.

[3] C. Pimentel, R. D. Souza, B. F. Uchôa-Filho and M. E. Pellenz, "Generalized Punctured Convolutional Codes with Unequal Error Protection". *EURASIP Journal on Advances in Signal Processing*, vol. 2008, ID 280831, 6 pp, Oct 2008.

[4] T. Q. Duong, U. Engelke and H.-J. Zepernick, "Unequal Error Protection for Wireless Multimedia Transmission in Decode-and-Forward Relay Networks," in *Proc. of the IEEE Radio and Wireless Symposium (RWS 2009)*, San Diego, EUA, 2009.

[5] H. X. Nguyen, H. H. Nguyen and T. Le-Ngoc, "Signal Transmission With Unequal Error Protection in Wireless Relay Networks". *IEEE Trans. Veh. Technol.*, vol. 59, n. 5, pp. 2166-2178, Jun 2010.

[6] T. Gnanasekaran and K. Duraiswamy, "Performance of Unequal Error Protection Using a Maximum A-Posteriori Algorithm and Modified MAP in Additive White Gaussian Noise and Fading Channel". *Journal of Computer Science*, vol. 4, n. 7, pp. 585-590, 2008.

[7] M. Fossorier, S. Lin and J. Snyders, "Reliability-Based Syndrome Decoding of Linear Block Codes". *IEEE Trans. Inf. Theory*, vol. IT - 44, pp. 388-398, Jan 1998.

[8] R. Pyndiah, "Near Optimum Decoding of Product Codes: Block Turbo Codes," *IEEE Trans. Commun.*, vol. 46, n.8, pp. 1003-1010, Aug 1998.

[9] N. N. Tendolkar and C. R. P. Hartmann "Generalization of Chase Algorithms for Soft Decision Decoding of Binary Linear Codes". *IEEE Trans. Inf. Theory*, vol. IT - 30, n. 5, pp. 714-721, Sep 1984.

[10] B. Masnick and J. Wolf, "On Linear Unequal Error Protection Codes". *IEEE Trans. Inf. Theory*, vol. IT - 3, n. 4, pp. 600-607, Oct 1967.

[11] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information". *IEEE Trans. Inf. Theory*, vol. IT - 18, n. 1, pp. 170-182, Jan 1972.

[12] W. J. van Gils, "On Linear Unequal Error Protection Codes," Diss. de Mestrado, Eindhoven Univ. of Technol., Eindhoven, Holanda, 1982.