

Decodificação por Apagamento Ponderado Aplicada a Códigos com Proteção Desigual de Erros

Rebecca C. de Albuquerque¹, Daniel C. Cunha^{2,3} e Cecilio Pimentel⁴

Resumo—Este trabalho realiza uma análise, por meio de simulações computacionais, do algoritmo de decodificação por apagamento ponderado (WED, do inglês *weighted erasure decoding*) 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 WED é comparado com o algoritmo de decodificação por máxima verossimilhança e uma avaliação do compromisso entre desempenho e complexidade do algoritmo WED é realizada para cada classe de proteção.

Palavras-Chave—proteção desigual de erros, decodificação por decisão suave, complexidade computacional.

Abstract—This work analyzes, through computer simulations, the effectiveness of the weighted erasure decoding (WED) algorithm applied to unequal error protection block codes using binary transmission over an additive white Gaussian noise channel. The performance of the WED algorithm is compared to maximum likelihood decoding one and an evaluation of the tradeoff between performance and complexity of the WED algorithm for each class of protection is provided.

Keywords—unequal error protection, soft decision decoding, computational complexity.

I. INTRODUÇÃO

Mobilidade, suporte a dados multimídia e rápido processamento têm sido requisitos decisivos para o oferecimento de equipamentos e serviços de telecomunicações com boa qualidade. No tocante à transmissão de dados multimídia em redes sem fio, a percepção de qualidade do usuário está relacionada à ausência de erros, principalmente em determinadas porções da mensagem [1]. Esquemas de transmissão, como o ATM (do inglês, *asynchronous transfer mode*), também apresentam maior nível de degradação quando os erros ocorrem na região onde se localizam as informações de roteamento, o cabeçalho da célula [2]. Em ambos os casos, esquemas com proteção desigual de erros (UEP, do inglês *unequal error protection*) [3] são utilizados para adequar as técnicas de proteção contra erros às características da rede ou da mensagem. As técnicas UEP têm sido investigadas pela comunidade científica na proposição de esquemas de codificação de rede aplicados a sistemas de comunicações cooperativas [4].

¹ Terceiro Centro Integrado de Defesa Aérea e Controle de Tráfego Aéreo (CINDACTA III), Recife-PE, Brasil.

² Departamento de Sistemas de Computação, Centro de Informática, Universidade Federal de Pernambuco (UFPE), Recife-PE, Brasil.

³ Prog. de Pós-Grad. em Engenharia de Sistemas, UPE, Recife-PE, Brasil.

⁴ Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco (UFPE), Recife-PE, Brasil.

E-mails: rebeccacamile@gmail.com, dcunha@cin.ufpe.br e cecilio@ufpe.br. Este trabalho foi parcialmente financiado pela FACEPE (APQ-1060-3.04/10) e pelo CNPq (302535/2010-1).

No que diz respeito à utilização de códigos corretores de erros UEP, a velocidade com que os dados são processados no receptor está associada, dentre outros fatores, ao algoritmo de decodificação utilizado. Outro aspecto importante, diretamente ligado ao algoritmo de decodificação empregado no receptor, é o consumo de energia, principalmente nas aplicações em que os nós da rede possuem vida útil limitada, como as redes de sensores sem fio. Esforços no sentido de otimizar o consumo de energia em sistemas codificados tem sido realizados nos últimos anos [5]–[7].

Os algoritmos de decodificação por decisão suave subótimos tem o propósito de fornecer um bom desempenho a um custo de processamento inferior ao do algoritmo de máxima verossimilhança (MV). Pertencentes a essa classe, os algoritmos de Chase-2 Generalizado (CG-2) [8] e WED (do inglês, *weighted erasure decoding*) [9], [10] são algoritmos de baixa complexidade computacional, quando comparados ao algoritmo MV. A análise de desempenho do algoritmo CG-2 operando em códigos UEP foi realizada em [11]. Não existe na literatura estudo semelhante para o algoritmo WED.

Esse trabalho realiza uma avaliação do algoritmo de decodificação WED aplicado a códigos UEP para cada classe de proteção (cp_i), considerando transmissão binária em um canal com ruído aditivo Gaussiano branco e usando como referência o desempenho do algoritmo MV. Na seção II, o algoritmo WED é descrito e a sua complexidade computacional é estimada em termos do número de operações matemáticas realizadas. Na Seção III, resultados de simulação são expostos. Por fim, a seção IV apresenta as conclusões do trabalho.

II. ALGORITMO DE DECODIFICAÇÃO WED

Considere um código UEP linear binário $C(n, k, d)$ 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. Define-se o vetor de separação $s = [s_1, \dots, s_i, \dots, s_k]$ do código $C(n, k, d)$ de maneira que a i -ésima posição de s é dada por [12]

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

em que $w(\mathbf{uG})$ é o peso de Hamming da palavra-código $\mathbf{x} = \mathbf{uG}$ e $GF(2)$ é o campo de Galois binário.

O vetor de separação s mede a proteção desigual de erros fornecida pelo código C para a decodificação MV e suas posições distintas permitem a obtenção do vetor de capacidade de correção t do código C , tal que

$$t_j = \lfloor \frac{s_j - 1}{2} \rfloor, j = 1, \dots, m_d, \quad (2)$$

em que m_d é o número de posições distintas do vetor de separação \mathbf{s} . O menor elemento de \mathbf{t} é a capacidade de correção do código C e será denotada por t^* .

Dada uma sequência \mathbf{r} , obtida na entrada do receptor a partir da transmissão de uma palavra-código, o algoritmo WED é baseado na quantização dos valores obtidos na saída dos filtros casados em Q regiões conforme ilustra a Figura 1. As regiões são uniformemente espaçadas pelo fator δ . O valor ótimo de δ , denotado por δ_{ot} e que minimiza a probabilidade de erro de bit P_b para um determinado valor de relação sinal-ruído (RSR) E_b/N_0 , pode ser obtido algebricamente [13] ou por meio de simulações computacionais.

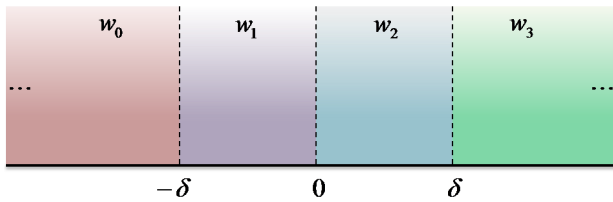


Fig. 1. Distribuição das regiões de quantização para $Q = 4$.

A cada uma das Q regiões são associados pesos w_i que estão relacionados à probabilidade de transição de um dos dois níveis de entrada para um dos Q níveis de saída [10]. Os pesos w_i são dados por [9]

$$w_i = \frac{i}{(Q-1)}, \quad 0 \leq i \leq Q-1. \quad (3)$$

Supondo $Q = 4$, os pesos w_i que obedecem à Eq. (3) são $w_0 = 0$, $w_1 = 0,333$, $w_2 = 0,666$ e $w_3 = 1$.

Define-se a sequência $\mathbf{v} = [v_0, \dots, v_j, \dots, v_{m-1}]$, com $m = \log_2 Q$, tal que cada componente v_j é dada por

$$v_j = \frac{2^{m-j-1}}{Q-1}, \quad 0 \leq j \leq m-1. \quad (4)$$

Para $Q = 4$, os valores de v_j , obtidos a partir da Eq. (4), são $v_0 = 0,666$ e $v_1 = 0,333$.

As regiões de quantização são associadas a sequências \mathbf{a}_i de comprimento m , tal que $0 \leq i \leq Q-1$. Estas sequências são geradas a partir da representação binária dos índices que identificam os pesos w_i associados às regiões de quantização. Os bits que compõem as sequências \mathbf{a}_i podem ser obtidos por meio da inspeção da equação

$$w_i = \sum_{j=0}^{m-1} a_i^j v_j, \quad (5)$$

para $0 \leq i \leq Q-1$ e $0 \leq j \leq m-1$. Ainda para $Q = 4$, pode-se obter, por exemplo, a sequência \mathbf{a}_1 conforme explicado a seguir. Uma vez que $w_1 = 0,333$, $v_0 = 0,666$ e $v_1 = 0,333$, verifica-se, usando a Eq. (5), que $a_1^0 = 0$ e $a_1^1 = 1$. As demais sequências \mathbf{a}_i tal que $i = 0, 2, 3$, são obtidas por inspeção semelhante.

Para um melhor entendimento do algoritmo WED, vamos verificar um exemplo simples. Considere a transmissão de uma sequência do código $C(7, 4, 3)$ e a recepção da sequência $\mathbf{r} = [-0,05 \quad -1,0 \quad 1,2 \quad -0,4 \quad -0,03 \quad -0,7 \quad -1,6]$. O mapeamento da sequência \mathbf{r} em quatro regiões, assumindo

que $\delta = 0,1$, permite construir a sequência \mathbf{q} com os índices dos pesos w_i decorrentes da quantização, tal que $\mathbf{q} = [1 \quad 0 \quad 3 \quad 0 \quad 1 \quad 0 \quad 0]$.

A partir daí, uma matriz \mathbf{A} , de dimensões $m \times n$, é construída de forma que sua l -ésima coluna seja a sequência \mathbf{a}_i , em que $i = q_l$ e $0 \leq l \leq n-1$. Para a sequência \mathbf{q} obtida anteriormente e utilizando as sequências \mathbf{a}_i calculadas para $Q = 4$, temos que

$$\mathbf{A} = [\mathbf{a}_{q_l}] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (6)$$

Observe que a primeira linha da matriz \mathbf{A} corresponde à decodificação por decisão abrupta da sequência \mathbf{r} .

Em seguida, a decodificação binária de cada linha da matriz \mathbf{A} dá origem à matriz \mathbf{A}' , também de dimensões $m \times n$. A decodificação binária de cada linha da matriz \mathbf{A} , dada por (6), resulta em

$$\mathbf{A}' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (7)$$

A estimativa da palavra transmitida é gerada a partir das decisões tomadas para cada coluna da matriz \mathbf{A}' . Para isso, define-se o vetor $\mathbf{f} = [f_0, \dots, f_j, \dots, f_m]$, em que cada uma das m componentes f_j corresponde ao número de posições em que a j -ésima linha de \mathbf{A}' difere da j -ésima linha de \mathbf{A} . Para as matrizes \mathbf{A} e \mathbf{A}' , dadas por (6) e (7), respectivamente, obtém-se os valores de $f_0 = 1$ e $f_1 = 1$.

De posse do vetor \mathbf{f} , calcula-se a confiabilidade R_j da j -ésima linha de \mathbf{A}' tal que

$$R_j = \max \{0, d - 2f_j\}, \quad (8)$$

O algoritmo WED proposto em [9] não considera a alteração do número de erros que o decodificador binário é capaz de corrigir, denotado por t . Para permitir a utilização de valores de t diferentes da capacidade de correção do código, $t^* = \lfloor (d-1)/2 \rfloor$, propõe-se o emprego da confiabilidade R'_j , dada por

$$R'_j = \max \{0, 2t + 1 - 2f_j\}. \quad (9)$$

Assume-se que $R'_j = 0$, caso o decodificador binário não encontre o padrão de erro associado à síndrome calculada para a j -ésima linha da matriz \mathbf{A} . Essa consideração tem o propósito de reduzir a confiabilidade de palavras nas quais a quantidade elevada de erros tenha inviabilizado a decodificação binária, além de privilegiar as palavras candidatas com menor quantidade de erros. As confiabilidades das linhas da matriz \mathbf{A}' , dadas por (7), são $R'_0 = R'_1 = 1$.

Assuma que S_0^l corresponde ao conjunto de índices das linhas de \mathbf{A}' que contém o bit 0 na l -ésima coluna e que S_1^l , analogamente, corresponde ao conjunto de índices para a presença do bit 1. O l -ésimo bit é decodificado como 0 se

$$\sum_{S_0^l} R'_j v_j > \sum_{S_1^l} R'_j v_j \quad (10)$$

ou como 1 se

$$\sum_{S_0^l} R'_j v_j < \sum_{S_1^l} R'_j v_j. \quad (11)$$

Caso $\sum_{S_0^l} R_j' v_j = \sum_{S_1^l} R_j' v_j$, o l -ésimo bit será a decodificação por decisão abrupta da componente r_l .

Para a primeira coluna da matriz \mathbf{A}' , temos que $S_0^0 = 0$ e $S_1^0 = 1$. Como $v_0 > v_1$ ($0,666 > 0,333$), tem-se que $\hat{x}_0 = 0$. Prosseguindo a decodificação para as demais colunas, verifica-se que $\hat{\mathbf{x}}$, a estimativa da palavra-código transmitida, é a sequência toda nula.

A. Complexidade do Algoritmo WED

Com os valores de Q e δ definidos, a execução do algoritmo WED segue as etapas descritas na Tabela I. Os cálculos de w_i , v e \mathbf{a}_i compõem o pré-processamento do algoritmo e não precisam ser reexecutados a cada palavra recebida. Sendo assim, essas operações não são consideradas na análise de complexidade.

TABELA I
DESCRIÇÃO DAS ETAPAS DO ALGORITMO WED.

Etapa	Descrição
1	Quantizar a sequência \mathbf{r} em Q níveis, obtendo a sequência \mathbf{q} .
2	Construir a matriz \mathbf{A} , a partir das sequências \mathbf{a}_i , de acordo com a sequência \mathbf{q} .
3	Decodificar as m linhas da matriz \mathbf{A} com o decodificador binário obtendo a matriz \mathbf{A}' .
4	Obter o vetor \mathbf{f} . Calcular R_j' .
5	Obter os conjuntos S_0^l e S_1^l . Realizar as comparações $\sum_{S_0^l} R_j' v_j \gtrless \sum_{S_1^l} R_j' v_j$, obtendo $\hat{\mathbf{x}}$.

A Tabela II relaciona a quantidade de operações matemáticas necessárias para a execução do algoritmo WED, em que N_a , N_{a2} , N_{mt} e N_c são, respectivamente, o número de adições, adições módulo-2, multiplicações e comparações. Na Etapa 1, foi considerado o mapeamento por árvore binária [14], no qual, para Q regiões, são necessárias m comparações para a quantização de uma componente da sequência \mathbf{r} . A Etapa 2 foi omitida, pois não necessita de operações matemáticas, uma vez que se trata de uma indexação a uma das sequências \mathbf{a}_i inicialmente computadas e armazenadas.

Finalmente, na Etapa 5, dependendo da sequência que estiver sendo decodificada, é possível que sejam necessárias $(m-1)$ ou $(m-2)$ adições para se realizar uma comparação $\sum_{S_0^l} R_j' v_j \gtrless \sum_{S_1^l} R_j' v_j$. Para fins de análise, foi considerado o pior caso para todas as n posições, totalizando $n(m-1)$ adições por palavra decodificada.

III. RESULTADOS NUMÉRICOS

O compromisso entre desempenho e complexidade do algoritmo WED foi investigado em decorrência da variação dos parâmetros t e Q para os códigos 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 $\mathbf{s}_2 = [12, 12, 5, 5, 5, 5, 5, 5]$, além dos vetores de capacidade de correção $\mathbf{t}_1 = [t_1^0, t_1^1] = [3, 2]$ e $\mathbf{t}_2 = [t_2^0, t_2^1] = [5, 2]$, respectivamente, obtidos em [11]. Desta forma, ambos os

códigos possuem duas classes de proteção, denotadas por cp_1 e cp_2 .

Para permitir a variação do parâmetro t , a confiabilidade R_j' foi empregada conforme definida em (9). Nas simulações, t variou de t^* até $t_\alpha^0 + 1$ (capacidade de correção associada à classe cp_1 do código C_α acrescida de 1). Em relação ao níveis de quantização, foram escolhidos os valores de $Q = 4, 16$ e 1024 . O desempenho do algoritmo WED para cada combinação de parâmetros t e Q , denotado por $\text{WED}(t, Q)$, foi comparado ao desempenho dos algoritmo MV suave.

A obtenção das curvas $P_b \times E_b/N_0$ para o algoritmo $\text{WED}(t, Q)$ necessitou da seleção do passo de quantização ótimo (δ_{ot}). Os valores de δ_{ot} foram gerados por meio de simulações computacionais para as classes cp_1 e cp_2 de cada código UEP. A Figura 2 mostra a variação de δ para o algoritmo $\text{WED}(2, 4)$ aplicado ao código C_1 , assumindo que $E_b/N_0 = 4$ dB. Para esse caso, foi obtido $\delta_{ot} = 0,39$ para a classe cp_1 e $\delta_{ot} = 0,41$ para a classe cp_2 .

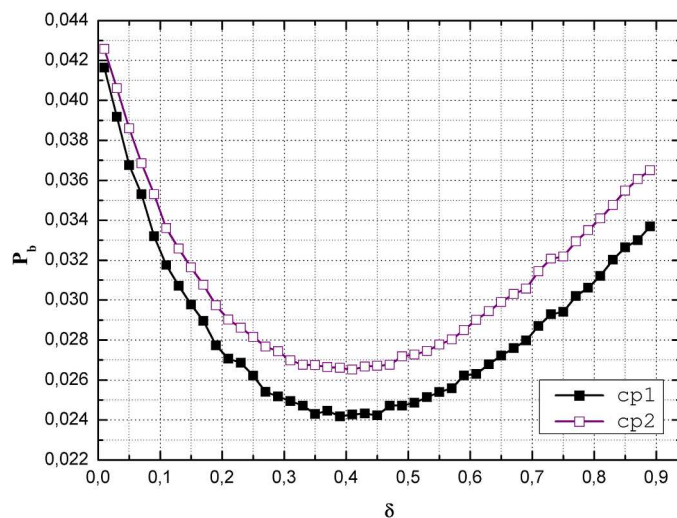


Fig. 2. Probabilidade de erro de bit em função de δ para o algoritmo $\text{WED}(2, 4)$ aplicado ao código UEP $C_1(16, 5, 5)$ para $E_b/N_0 = 4$ dB.

Em geral, o aumento do número de níveis de quantização provoca uma diminuição do valor de δ_{ot} , reduzindo cada vez mais o espaçamento entre os níveis adjacentes. Dessa forma, a faixa de valores considerada não sofre progressivos distanciamentos do zero com o aumento de Q . Esse comportamento, decorrente do aumento de Q , pode ser observado na Tabela III para o algoritmo $\text{WED}(2, Q)$. Além disso, a Tabela III apresenta a variação de δ_{ot} para a classe cp_1 em função de E_b/N_0 . Foram escolhidos os valores de δ_{ot} para a classe cp_1 , por ser a classe de maior relevância. Para ambos os códigos e os valores de Q considerados, δ_{ot} aumentou com o crescimento de E_b/N_0 , embora, para $Q = 1024$, esse aumento tenha sido apenas na quarta casa decimal. O aumento de δ_{ot} expande a faixa de valores considerada na quantização correspondente a um certo valor de Q . Dessa forma, é possível acompanhar o crescimento, em módulo, das amplitudes máximas e mínimas observadas no sinal recebido.

Após a obtenção de δ_{ot} para a classe cp_1 , as curvas de desempenho do algoritmo $\text{WED}(t, Q)$ foram geradas para os códigos C_1 e C_2 . A Figura 3 apresenta o desempenho dos

TABELA II
NÚMERO DE OPERAÇÕES MATEMÁTICAS REALIZADAS NO ALGORITMO WED POR PALAVRA DECODIFICADA.

Etapa	N_a	N_{a2}	N_{mt}	N_c
1	-	-	-	mn
3	-	$m[n + (n - k)(n - 1)]$	$mn(n - k)$	-
4	$m(n + 1)$	mn	$2m$	m
5	$n(m - 1)$	-	mn	$n(m + 1)$

TABELA III

VALORES DE δ_{ot} DA CLASSE cp_1 PARA O ALGORITMO WED(2, Q) EM FUNÇÃO DE E_b/N_0 .

E_b/N_0	δ_{ot}			
	$C_1(16, 5, 5)$		$C_2(25, 8, 5)$	
	$Q = 4$	$Q = 1024$	$Q = 4$	$Q = 1024$
0	0,31	0,0011	0,27	0,0011
1	0,37	0,0013	0,23	0,0011
2	0,37	0,0013	0,27	0,0011
3	0,37	0,0013	0,27	0,0011
4	0,39	0,0013	0,33	0,0011
5	0,45	0,0015	0,29	0,0011
6	0,47	0,0015	0,35	0,0013
7	0,51	0,0015	0,37	0,0013
8	0,67	0,0017	0,49	0,0017
9	0,67	0,0017	0,57	0,0019

algoritmos MV, WED(2, 4) e WED(3, 16) para o código C_1 . O aumento de Q e t provocou melhoria de desempenho nas duas classes de proteção. Para $P_b = 10^{-4}$, os ganhos obtidos foram de 1,72 dB e 1,15 dB para as classes cp_1 e cp_2 , respectivamente. Para o algoritmo WED(3, 16), o desempenho da classe cp_2 foi melhor que o desempenho da classe cp_1 até $E_b/N_0 = 3,67$ dB. Entretanto, houve uma aproximação ao desempenho do algoritmo MV para ambas as classes. A diferença de E_b/N_0 , para $P_b = 10^{-4}$, entre os desempenhos dos algoritmos WED(3, 16) e MV foi de 0,94 dB (cp_1) e de 0,79 dB (cp_2).

Para o código C_2 , os desempenhos dos algoritmos MV, WED(2, 4) e WED(5, 16) são apresentados na Figura 4. No algoritmo WED(2, 4), a classe cp_2 obteve melhor desempenho que a classe cp_1 , de forma semelhante ao que fora observado para o algoritmo CG-2 aplicado ao código C_2 [11]. Para $P_b = 10^{-4}$ e considerando a classe cp_2 , a diferença entre as RSRs dos algoritmos WED(2, 4) e MV (2,42 dB) é menor do que a diferença de RSR observada para a classe cp_1 (4,53 dB).

A alteração dos parâmetros para $Q = 16$ e $t = 5$ mostrou-se eficaz para adequar o desempenho das classes ao comportamento indicado pelo grau de proteção observado no vetor de separação s_2 . Para $P_b = 10^{-4}$, a classe cp_1 do algoritmo WED(5, 16) apresentou uma diferença de 1,04 dB em relação à classe cp_1 do algoritmo MV, enquanto a diferença para as classes cp_2 dos mesmos algoritmos foi de 1,37 dB.

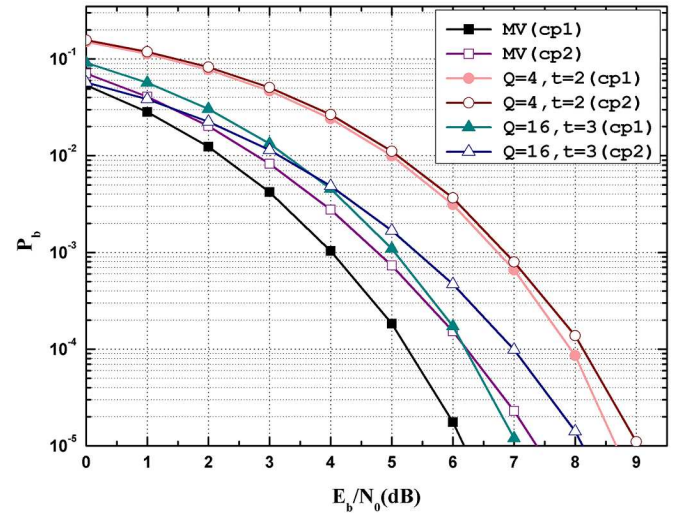


Fig. 3. Desempenho dos algoritmos de decodificação MV e WED(t, Q) para o código UEP $C_1(16, 5, 5)$ considerando o uso de modulação binária e canal com ruído aditivo Gaussiano branco.

A Tabela IV apresenta os resultados de desempenho e complexidade do algoritmo WED(t, Q) por classe de proteção. O desempenho é indicado por Δ_α , diferença (em dB) entre as RSRs dos algoritmos WED(t, Q) e MV para o código C_α e considerando $P_b = 10^{-4}$. Por exemplo, o valor de Δ_1 para os algoritmos WED(2, 4) e MV é de 2,68 dB. Já a complexidade é representada pelo número de operações matemáticas necessárias para realizar a decodificação completa de uma palavra recebida, sendo denotada por N_{om} (vide Tabela II). Admitiu-se que as quatro operações consideradas possuem o mesmo custo computacional.

Observa-se que, para o código C_2 , o aumento de Q , para t fixo, proporcionou uma melhoria de desempenho do algoritmo WED(t, Q) com o custo do aumento da complexidade. Entretanto, o aumento de t , para Q fixo, não implicou no aumento de N_{om} , mas fez o desempenho do algoritmo WED(t, Q) se aproximar do desempenho do algoritmo MV. Por exemplo, o algoritmo WED(2, 4) necessitou de 2024 operações matemáticas para a decodificação de uma palavra recebida com $\Delta_1 = 4,53$ dB. Já o algoritmo WED(5, 4) necessitou da mesma quantidade de operações, mas para $\Delta_1 = 1,32$ dB.

Ainda na Tabela IV, a análise dos resultados referentes à classe cp_1 permite observar que, para o código C_1 , não houve melhoria de desempenho quando $t > 3$. Para o código C_2 , com exceção do caso ($t = 6, Q = 4$), também ocorreu aumento de Δ_2 para $t > 5$, o que representa degradação de desempenho. Assim, percebe-se que para valores de t superiores a t_α^0 , em

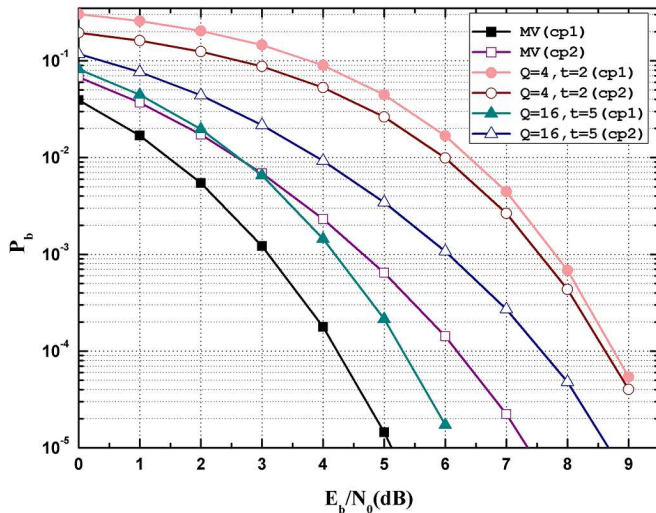


Fig. 4. Desempenho dos algoritmos de decodificação MV e WED(t, Q) para o código UEP $C_2(25, 8, 5)$ considerando o uso de modulação binária e canal com ruído aditivo Gaussiano branco.

TABELA IV

RESULTADOS DE DESEMPENHO (Δ_α) E COMPLEXIDADE (N_{om}) DOS ALGORITMOS WED(t, Q) APLICADOS AOS CÓDIGOS $C_1(16, 5, 5)$ E $C_2(25, 8, 5)$.

Q	$C_1(16, 5, 5)$			$C_2(25, 8, 5)$			
	4	16	1024	4	16	1024	
N_{om}	914	1828	4570	2024	4048	10120	
CP ₁	$t = 2$	2,68	2,10	1,67	4,53	4,12	3,67
	$t = 3$	1,25	0,94	0,83	3,36	2,98	2,49
	$t = 4$	1,33	0,94	0,88	2,33	2,98	1,42
	$t = 5$	-	-	-	1,32	1,04	0,98
	$t = 6$	-	-	-	1,19	1,12	1,15
CP ₂	$t = 2$	1,90	1,38	1,22	2,42	2,04	1,79
	$t = 3$	1,02	0,79	0,68	1,69	1,42	1,34
	$t = 4$	1,51	0,81	0,70	1,53	1,31	1,23
	$t = 5$	-	-	-	1,59	1,37	1,28
	$t = 6$	-	-	-	1,61	1,40	1,35

geral, o desempenho do algoritmo WED(t, Q) não melhora para as duas classes de proteção. É possível que a alteração do parâmetro R'_j permita ganhos de desempenho quando $t > t_\alpha^0$ ou que essa seja realmente uma limitação do algoritmo WED(t, Q).

Por fim, cabe ressaltar que a modificação do decodificador binário (t) não altera a complexidade do algoritmo WED(t, Q), o que não acontece quando o número de níveis de quantização (Q) é alterado.

IV. CONCLUSÕES

Este trabalho realizou, por meio de simulações computacionais, uma análise do algoritmo de decodificação WED para cada classe de proteção de códigos de bloco UEP, considerando o uso de transmissão binária em um canal com ruído aditivo Gaussiano branco. O desempenho do algoritmo

WED foi comparado ao desempenho do algoritmo de máxima verossimilhança. Para tanto, o passo de quantização ótimo do algoritmo WED foi obtido para cada código UEP considerado em uma faixa de valores de RSR e para diferentes níveis de quantização. Foi proposta uma mudança na confiabilidade que permitiu a variação do decodificador binário presente no algoritmo WED. Essa alteração no algoritmo proporcionou ganhos de desempenho que só seriam obtidos com um número elevado de níveis de quantização, o que implicaria em um alto custo computacional. Além disso, foi observado que a complexidade computacional do algoritmo WED não varia com a mudança do decodificador binário empregado. Todavia, o ganho de desempenho obtido foi limitado, o que pode ser consequência da confiabilidade proposta. Os autores pressupõem que uma investigação acerca da métrica de confiabilidade do algoritmo possa resultar na obtenção de desempenhos melhores sem que seja necessário empregar muitos níveis de quantização.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado pela Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERÊNCIAS

- [1] T. H. Yeo, W. C. Wong and D. Y. Huang, "Soft decision unequal error protection scheme for MPEG advanced audio coding". In *Proc. of the Int. Conf. on Multimedia and Expo (ICME 2003)*, 2003.
- [2] J. Dou, N. Wang and J. Wang, "A LDPC coding scheme for improving the satellite ATM links performance". In *Proc. of the WRI Int. Conf. on Communications and Mobile Computing*, 2009.
- [3] S. Borade, B. Nakiboglu and L. Zheng, "Unequal error protection: an information-theoretic perspective". *IEEE Trans. Inf. Theory*, v. 55, n. 12, pp. 5511-5539, 2009.
- [4] M. Iezzi, M. Di Renzo and F. Graziosi, "Network code design from unequal error protection coding: channel-aware receiver design and diversity analysis." In *Proc. of the IEEE Int. Conf. on Communications (ICC 2011)*, pp. 1-6, 2011.
- [5] S. L. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: When is ECC energy-efficient?". *EURASIP Journal on Wireless Communications and Networking*, 2006(2):1-14, 2006.
- [6] G. Balakrishnan, M. Yang, Y. Jiang and Y. Kim, "Performance analysis of error control codes for wireless sensor networks". In *Proc. of the Int. Conf. on Information Technology*, pp. 876-879, 2007.
- [7] S. Chouhan, M. Balakrishnan and R. Bose, "A framework for energy consumption based design space exploration for wireless sensor nodes". In *Proc. of the Int. Symposium on Low Power Electronics and Design (ISPLD'08)*, pp. 329-334, 2008.
- [8] N. N. Tendolkar and C. R. P. Hartmann "Generalization of Chase algorithms for soft decision decoding of binary linear codes". *IEEE Trans. Inf. Theory*, v. IT-30, n. 5, pp. 714-721, 1984.
- [9] E. Weldon Jr, "Decoding binary block codes on Q -ary output channels". *IEEE Trans. Inf. Theory*, v. 17, n. 6, pp. 713-718, 1971.
- [10] Y. Hong and S. Tsai, "Simulation and analysis of the weighted erasure decoding". *IEEE Trans. on Commun.*, v. 27, n. 2, pp. 483-488, 1979.
- [11] R. C. Albuquerque, D. C. Cunha e C. Pimentel, "An evaluation of the generalized Chase-2 algorithm applied to unequal error protection block codes". In *Proc. of the IEEE Latin-American Conf. on Communications (LATINCOM 2011)*, 2011.
- [12] 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.
- [13] W. H. J. Chen, M. P. C. Fossorier and S. Lin, "Optimum quantizer design for the weighted erasure decoding algorithm". In *Proc. of the IEEE Int. Conf. on Communications (ICC'99)*, v. 2, pp. 838-842, 1999.
- [14] A. M. Tanenbaum et al., *Estruturas de dados usando C*. Pearson Makron Books, 2004.