

RESEARCH

Open Access

# On the complexity-performance trade-off in soft-decision decoding for unequal error protection block codes

Rebecca C de Albuquerque<sup>1</sup>, Daniel C Cunha<sup>2\*</sup> and Cecilio Pimentel<sup>3</sup>

## Abstract

Unequal error protection (UEP) codes provide a selective level of protection for different blocks of the information message. The effectiveness of two sub-optimum soft-decision decoding algorithms, namely generalized Chase-2 and weighted erasure decoding, is evaluated in this study for each protection class of UEP block codes. The performances of both algorithms are compared to that of the maximum likelihood algorithm in order to evaluate the performance loss of each protection class provided by less complex algorithms as well as their complexities are evaluated according to the number of arithmetic operations performed at each decoding step. Finally, numerical results and examples are provided which indicate that a trade-off between performance and complexity for each protection class is obtained. The results of this study can be used to select appropriate UEP coding and decoding schemes in applications that demand low energy consumption.

**Keywords:** Block codes, Bit error probability, Unequal error protection, Soft-decision decoding, Computational complexity

## 1 Introduction

One of the main challenges in the design of battery-supplied wireless devices is the minimization of their energy consumption [1-4]. It is known that forward error correction (FEC) decoders are responsible for a large part of energy consumption of such devices [5,6]. Since maximum likelihood (ML) decoding is often infeasible due to its complexity of exponential order, it is of interest to investigate sub-optimum decoding techniques in search of less complex alternatives.

Concerning block codes, a class of sub-optimum algorithms that deserves attention is composed of reliability-based soft-decision decoding techniques [7]. In this category, Chase-2 and weighted erasure decoding (WED) algorithms are recognized by their ease of implementation and reduced complexity when compared to the ML algorithm. The performance of the Chase-2 decoding

algorithms applied to Bose–Chaudhuri–Hocquenghem codes is analyzed in [8].

In a number of wireless protocols, the importance of different bits in the information sequence often varies and certain blocks of this sequence need higher protection level than other blocks. This property is called unequal error protection (UEP) and can be obtained either by hierarchical modulation techniques [9,10] or by FEC schemes [11,12]. Such UEP methods have been applied to wireless and mobile computing applications [13-15], apart from several video and image coding standards as set partitioning hierarchical trees [16], ITU-T H.264 [17], and its extensions [18], and joint photographic expert group 2000 (JPEG 2000) [19]. Concerning UEP coding, the analysis of suboptimal decoding algorithms applied to UEP block codes has not been considered in the literature.

In this study, the effectiveness of sub-optimum soft-decision decoding algorithms (generalized Chase-2 algorithm [20] and WED algorithm [21]) for each protection class of UEP block codes is evaluated using binary transmission over an additive white Gaussian noise (AWGN) channel. Performances of both algorithms are compared

\*Correspondence: dcunha@cin.ufpe.br

<sup>2</sup>Department of Computing Systems, Centro de Informática, Federal University of Pernambuco (CIn/UFPE), 50740-560, Recife-PE, Brazil  
Full list of author information is available at the end of the article

to that of the ML algorithm in order to evaluate the performance loss of each protection class provided by less complex algorithms. We also analyze the arithmetic decoding complexity of each algorithm to decode a received sequence. In addition, an analysis of the trade-off between performance and complexity of the algorithms for each protection class is done. Based on this analysis, we discuss the choice of the parameters of the decoder with the best complexity-performance trade-off, such as the number of test patterns, the error-correcting capability of the binary decoder, and the number of quantization levels.

The remainder of this article is structured as follows: In Section 2, concepts related to UEP coding are described. The soft-decision decoding algorithms are defined in Section 3, while the analysis of their decoding complexity in terms of mathematical operations is presented in Section 4. Section 5 presents simulation results. A trade-off between performance and complexity for both decoding algorithms is established in this section. Finally, conclusions are drawn in Section 6.

## 2 UEP block codes

Consider a binary linear code  $C_j(n, k, d)$  in which  $n$  is the codeword length,  $k$  is the dimension of the code, and  $d$  is the minimum Hamming distance of  $C_j$ . The generator matrix of  $C_j$  is denoted by  $\mathbf{G}_j$ . Assume that  $w(\mathbf{u}\mathbf{G}_j)$  is the Hamming weight of the codeword  $\mathbf{x} = \mathbf{u}\mathbf{G}_j$  related to the information vector  $\mathbf{u}$ . The separation vector of  $C_j$ ,  $\mathbf{s}_j = [s_j^0, \dots, s_j^i, \dots, s_j^{k-1}]$ , measures the UEP provided by the code  $C_j$  for ML decoding [22]. The  $i$ th position of  $\mathbf{s}_j$  is given by [22]

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

where  $\text{GF}(2)$  is the binary Galois field. The smallest element of  $\mathbf{s}_j$  is the minimum Hamming distance of  $C_j$ . A code  $C_j$  is said to have equal error protection capability if all elements of  $\mathbf{s}_j$  are equal, otherwise  $C_j$  has the UEP property. The error-correcting capability of the code  $C_j$  is denoted by  $t_j^*$ .

To illustrate these concepts, consider the linear block codes  $C_1(16, 5, 5)$  and  $C_2(25, 8, 5)$  with generator matrices  $\mathbf{G}_1$  and  $\mathbf{G}_2$  (calculated using the method proposed in [23]) given by

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

$$\mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (3)$$

Their separation vectors are  $\mathbf{s}_1 = [8, 8, 5, 5, 5]$  and  $\mathbf{s}_2 = [12, 12, 5, 5, 5, 5, 5, 5]$ , respectively. Thus, we can say that both codes are UEP codes with two distinct protection classes, denoted by  $\mathcal{C}_{P_1}$  (higher protection class) and  $\mathcal{C}_{P_2}$  (lower protection class).

## 3 Soft-decision decoding algorithms

Two decoding algorithms that deal with the least reliable positions of the received sequence, namely the generalized Chase-2 (GC-2) [20] and the WED [21] algorithms, are described in this section.

### 3.1 Generalized Chase-2 decoding algorithm

The GC-2 algorithm uses the sequence of real values observed at the output of the matched filter,  $\mathbf{r} = [r_0, r_1, \dots, r_{n-1}]$ , and the binary sequence  $\mathbf{y}$  obtained by a hard quantization of  $\mathbf{r}$ . For the AWGN channel, the real values of the sequence  $\mathbf{r}$  correspond to the reliabilities  $\alpha_i$  such that  $\alpha_i = |r_i|$ . Thus, the higher the value of  $\alpha_i$ , the lower the probability that the corresponding symbol had been strongly affected by the noise.

Let  $p$  be the number of least reliable positions of the sequence  $\mathbf{r}$ , i.e., the positions that have the least values of  $\alpha_i$ . The value of  $p$  determines the set of test patterns  $S_b = \{\mathbf{b}_i\}$ ,  $i = 0, \dots, 2^p - 1$ , with cardinality  $|S_b| = 2^p$ . At first, the GC-2 algorithm applies a binary decoder (with error-correcting capability  $t$ ) to find an error pattern  $\mathbf{z}$  associated to the sequence  $\mathbf{y}^i = \mathbf{y} \oplus \mathbf{b}_i$ , in which  $\oplus$  represents sum modulo-2. If an error pattern  $\mathbf{z}$  is obtained by the binary decoder,<sup>a</sup> it is added to the test pattern  $\mathbf{b}_i$ , resulting the pattern  $\mathbf{z}^i = \mathbf{z} \oplus \mathbf{b}_i$ . After that, the analog weight  $W_\alpha$  of the pattern  $\mathbf{z}^i$  is figured out according to

$$W_\alpha(\mathbf{z}^i) = \sum_{k=0}^{n-1} \alpha_k z_k^i. \quad (4)$$

If  $\mathbf{z}$  is not found by binary decoding, the next test pattern  $\mathbf{b}_i$  is selected. The objective of the GC-2 algorithm is to find the pattern  $\mathbf{z}^{i^*}$  with minimum analog weight  $W_\alpha$  to estimate the transmitted codeword  $\mathbf{x}$ , as  $\hat{\mathbf{x}} = \mathbf{y} \oplus \mathbf{z}^{i^*}$ . When a pattern  $\mathbf{z}^i$  is not selected (for all test patterns), then  $\hat{\mathbf{x}} = \mathbf{y}$ .

A detailed description of the GC-2 algorithm is found in [20,24] and a summary of its steps is presented in Table 1.

For a better understanding of the GC-2 algorithm, we consider the following example.

**Table 1 Description of the steps of the GC-2 algorithm**

Step	Description
1	Get the sequence $\mathbf{y}$ from the sequence $\mathbf{r}$ . Do $i = 0$ . Select a test pattern $\mathbf{b}_i$
2	Obtain the sequence $\mathbf{y}' = \mathbf{y} \oplus \mathbf{b}_i$
3	Compute the syndrome associated with the sequence $\mathbf{y}'$ (binary decoding) and search the corresponding error pattern $\mathbf{z}$
4	If $\mathbf{z}$ is found, then get the pattern $\mathbf{z}^i = \mathbf{z} \oplus \mathbf{b}_i$ , compute its analog weight $W_\alpha$ , store the pattern $\mathbf{z}^*$ that has the minimum analog weight, and go to the Step 5. Otherwise, go to Step 5
5	If there is still test patterns to generate, do $i = i + 1$ , select $\mathbf{b}_i$ , and go to Step 2. Otherwise, go to Step 6
6	If a pattern $\mathbf{z}^*$ was stored, obtain the estimate $\hat{\mathbf{x}} = \mathbf{y} \oplus \mathbf{z}^*$ . Otherwise, $\hat{\mathbf{x}} = \mathbf{y}$

**Example 1.** Consider the Hamming code  $C(7, 4, 3)$  whose error-correcting capability is equal to one. Assume that the codeword  $\mathbf{x} = [1, 0, 0, 1, 0, 1, 1]$  is BPSK modulated and is transmitted over the AWGN channel. Suppose that the received sequence is  $\mathbf{r} = [1.5, 0.05, -0.8, 2.2, 0.1, 1.2, 0.3]$ .

According to the first step of the GC-2 algorithm, the sequence  $\mathbf{y} = [1, 1, 0, 1, 1, 1, 1]$  is obtained by hard quantization of  $\mathbf{r}$ . Let us assume that  $p = 2$ , so the two least reliable positions are the second and the fifth ones. Thus, considering all the combinations of 0's and 1's in these two least reliable positions, we have four test patterns  $\mathbf{b}$  according to the set

$$S_b = \{\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\} = \{0000000, 0100000, 0000100, 0100100\}.$$

To obtain the sequence  $\mathbf{y}^0$ , the test pattern  $\mathbf{b}_0$  is selected, resulting in  $\mathbf{y}^0 = \mathbf{y} \oplus \mathbf{b}_0 = [1, 1, 0, 1, 1, 1, 1]$ . After computing the syndrome associated to this  $\mathbf{y}^0$ , we get the error pattern  $\mathbf{z} = [0, 0, 1, 0, 0, 0, 0]$ . Since the error pattern  $\mathbf{z}$  exists, the sequence  $\mathbf{z}^0 = \mathbf{z} \oplus \mathbf{b}_0 = [0, 0, 1, 0, 0, 0, 0]$  is achieved and its analog weight  $W_\alpha(\mathbf{z}^0)$  is 0.8, according to (4). Repeating this procedure with the other test patterns from  $S_b$ , the algorithm stores  $\mathbf{z}^{i^*} = \mathbf{z}^2 = [0, 1, 0, 0, 1, 0, 0]$  as the sequence with the minimum analog weight ( $W_\alpha(\mathbf{z}^2) = 0.15$ ). Finally, the estimate  $\hat{\mathbf{x}} = \mathbf{y} \oplus \mathbf{z}^2 = [1, 0, 0, 1, 0, 1, 1]$  is obtained, characterizing the correct codeword.

### 3.2 WED Algorithm

The WED algorithm is based on the quantization of the sequence  $\mathbf{r}$  in  $Q = 2^m$  regions that are uniformly spaced

by the quantization step  $\delta$ . Figure 1 illustrates the quantization regions (denoted by  $R_{D_j}, 0 \leq j \leq Q - 1$ ) for  $Q = 8$  ( $m = 3$ ). The optimal value of  $\delta$ , denoted by  $\delta_{op}$ , that minimizes the bit error probability, can be obtained algebraically [25] or through computer simulations.

Given  $\mathbf{r}$  and  $Q$ , two sequences ( $\mathbf{v}$  and  $\mathbf{q}$ ) are obtained. First, consider the sequence  $\mathbf{v} = [v_0, \dots, v_\ell, \dots, v_{m-1}]$ , where each component  $v_\ell$  is given by

$$v_\ell = \frac{2^{m-\ell-1}}{Q-1}, \quad 0 \leq \ell \leq m-1. \quad (5)$$

The  $Q$ -ary sequence  $\mathbf{q} = [q_0, q_1, \dots, q_i, \dots, q_{n-1}]$ ,  $q_i \in \{0, \dots, Q-1\}$  is defined such that  $q_i = j$ , if  $r_i \in R_{D_j}$ . Then, a matrix  $\mathbf{A}$  of dimensions  $m \times n$  is determined such that the  $i$ th column of  $\mathbf{A}$  is the binary representation of  $q_i$ .

Next, a matrix  $\mathbf{A}'$  having the same dimensions of  $\mathbf{A}$  is obtained from the binary decoding of the rows of  $\mathbf{A}$ . The syndrome of each row of  $\mathbf{A}$  is computed in order to find its associated error pattern. If an error pattern is found, it is added to the row of  $\mathbf{A}$  to generate the row of  $\mathbf{A}'$ . Otherwise, the row of  $\mathbf{A}'$  is equal to the row of  $\mathbf{A}$ .

We also define the vector  $\mathbf{f} = [f_0, \dots, f_\ell, \dots, f_{m-1}]$ , where each component  $f_\ell$  is the number of positions of the  $\ell$ th row of  $\mathbf{A}'$  that differs from the  $\ell$ th row of  $\mathbf{A}$ . Using  $\mathbf{f}$ , the reliability  $R_\ell$  of the  $\ell$ th row of  $\mathbf{A}'$  is computed as [21]

$$R_\ell = \max \{0, d - 2f_\ell\}. \quad (6)$$

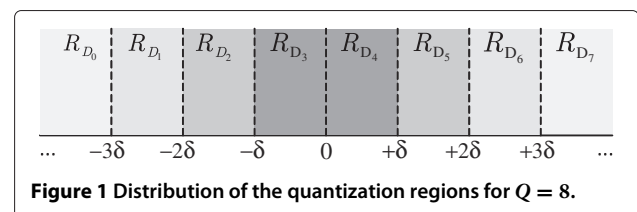
In the WED algorithm proposed in [21], the error-correcting capability of the binary decoder is  $t = t^* = \lfloor (d-1)/2 \rfloor$ . To allow the use of an arbitrary value of  $t$ , we propose a new reliability  $R'_\ell$  given by

$$R'_\ell = \max \{0, 2t + 1 - 2f_\ell\}. \quad (7)$$

It is assumed that  $R'_\ell = 0$  if the binary decoder cannot find the error pattern associated with the syndrome of the  $\ell$ th row of  $\mathbf{A}$ . This consideration is intended to reduce the reliability of sequences in which the high number of errors has made impossible the binary decoding. Also, the candidate sequences with fewer errors are favored.

Let  $S_0^i$  corresponds to the set of indices of the rows of  $\mathbf{A}'$  containing the bit 0 in  $i$ th column and  $S_1^i$ , the set of indices for the presence of the bit 1 in  $i$ th column. The  $i$ th bit is decoded as 0 if

$$\sum_{\ell \in S_0^i} R'_\ell v_\ell > \sum_{\ell \in S_1^i} R'_\ell v_\ell$$



**Figure 1 Distribution of the quantization regions for  $Q = 8$ .**

or as 1 if

$$\sum_{\ell \in S_0^i} R'_\ell v_\ell < \sum_{\ell \in S_1^i} R'_\ell v_\ell.$$

If  $\sum_{\ell \in S_0^i} R'_\ell v_\ell = \sum_{\ell \in S_1^i} R'_\ell v_\ell$ , the  $i$ th bit is obtained by hard-decision decoding of the component  $r_i$ .

A detailed description of the WED algorithm is found in [21], and a summary of its steps is presented in Table 2. For a better understanding of this algorithm, we consider in the following example the same code, transmitted codeword, and received sequence of Example 1.

**Example 2.** Assume the mapping of  $\mathbf{r}$  into four quantization regions ( $Q = 4$ ) with  $\delta = 0.2$ . According to the first step of the WED algorithm, the sequences  $\mathbf{q} = [3, 2, 0, 3, 2, 3, 3]$  and  $\mathbf{v} = [0.666, 0.333]$  are obtained. Given  $\mathbf{q}$ , the matrix  $\mathbf{A}$  is obtained as

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (8)$$

Next, applying a binary decoding (with  $t = 1$ ) to each row of  $\mathbf{A}$ , we obtain the matrix  $\mathbf{A}'$  as

$$\mathbf{A}' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (9)$$

From  $\mathbf{A}$  and  $\mathbf{A}'$ , we obtain  $\mathbf{f} = [f_0, f_1] = [1, 0]$ . Assuming  $t = 1$ , the reliabilities of the rows of the  $\mathbf{A}'$  are  $R'_0 = 1$  and  $R'_1 = 3$ . For the first column of  $\mathbf{A}'$ , we have  $S_0^0 = \{\emptyset\}$  and  $S_1^0 = \{0, 1\}$ , resulting in  $\hat{x}_0 = 0$ . For the second column of  $\mathbf{A}'$ ,  $S_0^1 = \{1\}$  and  $S_1^1 = \{0\}$ . Since  $R'_1 v_1 > R'_0 v_0$  ( $1 > 0.666$ ), we have  $\hat{x}_1 = 0$ . Continuing with the decoding, the estimate  $\hat{\mathbf{x}} = [1, 0, 0, 1, 0, 1, 1]$  is obtained. This is the correct codeword as it was obtained in the GC-2 algorithm.

**Table 2 Description of the steps of the WED algorithm**

Step	Description
1	Quantize the sequence $\mathbf{r}$ in $Q$ levels, obtaining the sequences $\mathbf{v}$ and $\mathbf{q}$
2	Construct the matrix $\mathbf{A}$ according to the binary representation of the sequence $\mathbf{q}$
3	Decode the $m$ rows of the matrix $\mathbf{A}$ by binary decoding obtaining the matrix $\mathbf{A}'$
4	Get the vector $\mathbf{f}$ and compute $R'_\ell$
5	Get the sets $S_0^i$ and $S_1^i$ . Make comparisons $\sum_{\ell \in S_0^i} R'_\ell v_\ell \gtrless \sum_{\ell \in S_1^i} R'_\ell v_\ell$ , obtaining $\hat{\mathbf{x}}$

#### 4 Arithmetic complexity of the GC-2 and WED algorithms

The complexity of both algorithms considered in this article is evaluated according to the number of arithmetic operations performed at each decoding step.<sup>b</sup> Consider  $N_s$ ,  $N_g$ ,  $N_m$ , and  $N_c$ , the number of additions, additions modulo-2, multiplications and comparisons, respectively.

Table 3 indicates the number of operations performed at each step of the GC-2 algorithm, as described in Table 1, for each decoded sequence. In Step 3, multiplications and additions modulo-2 correspond to the syndrome computing. Also in Step 3, we assume that there is no arithmetic operations associated to the search of an error pattern  $\mathbf{z}$  (a lookup table may be used for this purpose). Operations related to Steps 1, 5, and 6 are omitted because either they are not performed for each test pattern or do not require arithmetic operations. Thus, they represent a very small percentage of the whole operations.

It is noteworthy that the operations in Step 4 depend on the result obtained in Step 3, i.e., depend on the success of the binary decoder in the search for an error pattern  $\mathbf{z}$  associated with the sequence  $\mathbf{y}^i$ . Thus, it is necessary to estimate the average value of the operations performed in Step 4. For this, we define the relative frequency of computing  $W_\alpha$  as  $f_A = N_W/2^p$ , in which  $N_W$  is the number of times that the analog weight  $W_\alpha$  is computed in the main loop of the algorithm. This value is evaluated via computer simulations in the next section (see also [26]).

In the case of the WED algorithm, defined  $Q$  and  $\delta$ , the implementation of the algorithm follows the steps described in Table 2. The computing of the sequence  $\mathbf{v}$  depends only on  $Q$  and does not need to be executed for each received sequence. Therefore, these operations are not considered in Table 4 that relates the number of operations required to implement the WED algorithm for each decoded sequence. In Step 1, it is considered the binary tree mapping [27], in which, for  $Q$  regions,  $m$  comparisons are needed to the quantization of a component  $r_i$ . Step 2 is omitted, because it does not require mathematical operations.

Finally, in Step 5, depending on the sequence that is being decoded, it may be necessary either  $(m - 1)$  or  $(m - 2)$  additions to perform the comparison  $\sum_{\ell \in S_0^i} R'_\ell v_\ell \gtrless$

**Table 3 Number of mathematical operations performed in the GC-2 algorithm for each decoded sequence**

Step	$N_s$	$N_g$	$N_m$	$N_c$
2	-	$n2^p$	-	-
3	-	$2^p(n - k)(n - 1)$	$2^p(n - k)n$	-
4	$f_A 2^p(n - 1)$	$f_A 2^p n$	$f_A 2^p n$	$f_A 2^p$

**Table 4 Number of mathematical operations performed in the WED algorithm for each decoded sequence**

Step	$N_s$	$N_g$	$N_m$	$N_c$
1	-	-	-	$mn$
3	-	$m[n + (n - k)(n - 1)]$	$mn(n - k)$	-
4	$m(n + 1)$	$mn$	$2m$	$m$
5	$n(m - 1)$	-	$mn$	$n(m + 1)$

$\sum_{\ell \in S_1^i} R'_\ell v_\ell$ . It is considered the worst case for all  $n$  positions, totalizing  $n(m - 1)$  additions per decoded sequence.

## 5 Numerical results

The performance of three decoding algorithms (ML, GC-2, and WED) is evaluated via computer simulations for the two UEP codes defined in Section 2 using binary transmission over the AWGN channel. Various configurations of the GC-2 and WED algorithms are considered by changing their parameters ( $t$  and  $p$  for GC-2;  $t$  and  $Q$  for WED), in order to compare their performance to that of the ML algorithm for each protection class. Using these results together with the operations in Tables 3 and 4, a trade-off between performance and complexity for both decoding algorithms is also established. In the following sections, the GC-2 and the WED algorithms will be denoted by GC-2( $t, p$ ) and WED( $t, Q$ ), respectively.

### 5.1 GC-2 decoding algorithm

Figure 2 shows the curves of the bit error probability ( $P_b$ ) versus signal-to-noise ratio (SNR)  $E_b/N_0$ , in which  $E_b$  is the energy per information bit and  $N_0$  is the power spectral density of the noise, of the GC-2(2, 2) and GC-2(3, 4) algorithms for both classes of the UEP code  $C_1$ . For this code, the maximum value of the error-correcting capability of the binary decoder ( $t$ ) is assumed equal to 3 and the maximum value of  $p$  is such that the cardinality of  $S_b$  is always lower than the search set of the ML algorithm ( $|S_b| < 2^k - 1$ ).

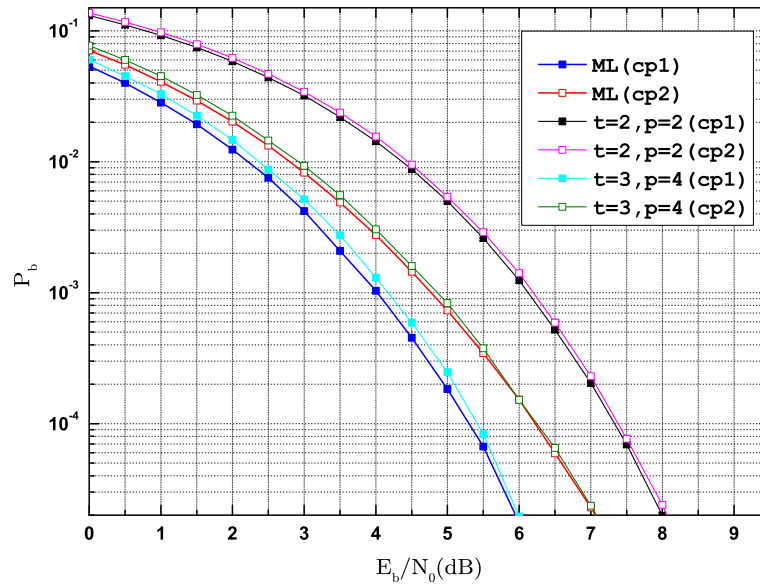
For the GC-2(2, 2) algorithm, we observe that there is virtually no performance difference between the classes  $cp_1$  and  $cp_2$ . In addition, considering  $P_b = 10^{-4}$ , the SNR difference compared to the ML algorithm is approximately 2 and 1.1 dB for the classes  $cp_1$  and  $cp_2$ , respectively. For the GC-2(3, 4) algorithm, the SNR difference to the ML algorithm is 0.1 dB ( $cp_1$ ) and 0.03 dB ( $cp_2$ ).

To assess the complexity of the GC-2 algorithm, it is necessary to evaluate  $f_A$ , as mentioned in Section 4. Figure 3 illustrates the values of  $f_A$  as a function of  $E_b/N_0$  for GC-2(2, 2), GC-2(2, 7), GC-2(3, 2), GC-2(3, 7), GC-2(6, 2), and GC-2(6, 7) algorithms applied to the UEP

code  $C_2$ . For  $p = 2$  and considering  $t = 2$ ,  $t = 3$ , and  $t = 6$ ,  $f_A$  reaches its maximum value (it is estimated for all test patterns  $\mathbf{b}_i$ ), when  $E_b/N_0 = 9.5, 8$ , and 4 dB, respectively. The reduction of SNR occurs due to the increased possibility of an error pattern  $\mathbf{z}$  be found, a consequence of the increasing of the error-correcting capability of the binary decoder. We observe that for  $t = 2$  and  $t = 3$  ( $p = 7$ ), there are test patterns  $\mathbf{b}$  which do not produce calculations of  $W_\alpha$  ( $f_A < 1$ ), even in regions of high SNR ( $E_b/N_0 > 7.5$  dB). For example, considering the GC-2(2, 7) algorithm and  $E_b/N_0 > 7.5$  dB, it is very probable that the bit inversion resulting from the addition of test patterns  $\mathbf{b}_i$  causes errors in the sequence  $\mathbf{y}$ . As the binary decoder used is able to correct only 2 errors in this algorithm, 31 estimates of  $W_\alpha$  ( $N_W = 31$ ) occur, which represents the sum of all test patterns of weight less than 2, resulting in  $f_A = 31/128 \simeq 0.242$ .

Finally, it is analyzed the compromise between performance and complexity of the GC-2 algorithms in terms of the SNR difference with respect to the ML algorithm related to the class  $cp_i$  (for  $P_b = 10^{-4}$ ), namely  $\Delta_i$  (dB), and the number of mathematical operations executed in the algorithm, defined as a 4-tuple MO =  $[N_s; N_g; N_m; N_c]$ . For the estimation of MO, it is necessary to determine in Step 4 of Table 3 the value of  $f_A$  used to weight the number of operations. Provided the GC-2 algorithm and the protection class  $cp_i, i = 1, 2$ , the SNR value corresponding to  $P_b = 10^{-4}$  is determined. With this SNR, we can identify the correspondent value of  $f_A$  (see Figure 3).

Tables 5 and 6 summarize the complexity-performance trade-off for various configurations of the GC-2 algorithm applied to the UEP codes  $C_1$  and  $C_2$ , respectively. For each intersection of a row ( $t$ ) with a column ( $p$ ), the values of  $\Delta_i$  (dB) and MO required to achieve  $P_b = 10^{-4}$  are shown for each protection class. For both codes, these results indicate that an increasing in  $p$  (for a fixed  $t$ ) provides better performance, however increases the complexity, since each operation shown in Table 3 grows exponentially with  $p$ . On the other hand, an increasing in  $t$  (for a fixed  $p$ ) also results in an improved performance, but with a smaller increasing of complexity. For example, considering the class  $cp_1$  of the code  $C_1$  and the GC-2(2, 2) algorithm, we have  $\Delta_1 = 2.0$  dB and MO =  $[N_s; N_g; N_m; N_c] \simeq [58.3; 790; 770; 3.89]$ . Moreover, the GC-2(3, 2) provides  $\Delta_1 = 0.9$  dB and MO =  $[N_s; N_g; N_m; N_c] \simeq [59.5; 790; 770; 3.96]$ , while the GC-2(2, 4) yields  $\Delta_1 = 0.8$  dB and MO =  $[N_s; N_g; N_m; N_c] \simeq [159.6; 3, 100; 3, 000; 10.6]$ , which represents a significant complexity increase related to the previous two cases, while the value of  $\Delta_1$  is approximately the same as obtained by the GC-2(3, 2) algorithm. This analysis led us to conclude that the increase

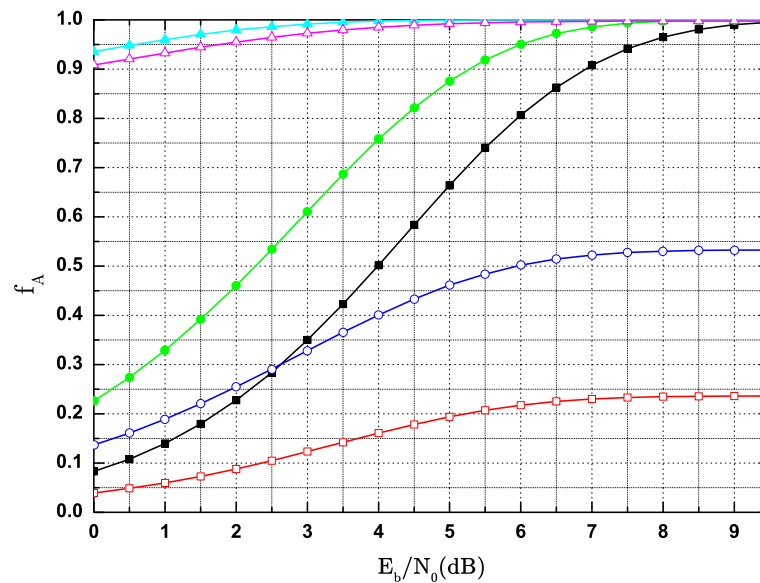


**Figure 2** Performance of ML and GC-2  $(t, p)$  decoding algorithms for UEP code  $C_1(16, 5)$  considering the use of binary antipodal modulation and AWGN channel.

of the error-correcting capability of the binary decoder is more advantageous than the increase of the number of test patterns of the GC-2 algorithm. Also, it is possible to observe (analyzing Tables 5 and 6) that, in most of the cases,  $\Delta_2$  is smaller than  $\Delta_1$ , indicating that the performance achieved by the class  $cp_2$  is closer to the ML one than the obtained by the class  $cp_1$ .

### 5.2 WED algorithm

The WED $(t, Q)$  algorithm uses the reliability  $R'_\ell$  defined in (7). The number of quantization regions considered is  $Q = 4, 16$  and  $1024$ . Table 7 illustrates the optimal value of the quantization step,  $\delta_{op}$ , (in the sense of minimizing  $P_b$ ) for the WED $(2, Q)$  algorithm (class  $cp_1$ ) applied to the codes  $C_1$  and  $C_2$ . In general, increasing the number of quantization regions causes a decrease



**Figure 3** Relative frequency of the execution of the calculation of the analog weight  $W_\alpha$  in the GC-2  $(t, p)$  algorithm applied to the code  $C_2(25, 8, 5)$  as a function of  $E_b/N_0$ . Parameters of the GC-2 algorithm: (black square symbol):  $t = 2, p = 2$ ; (red square symbol):  $t = 2, p = 7$ ; (green circle symbol):  $t = 3, p = 2$ ; (blue circle symbol):  $t = 3, p = 7$ ; (cyan triangle symbol):  $t = 6, p = 2$ ; (magenta triangle symbol):  $t = 6, p = 7$ .

**Table 5 Results of performance and arithmetic complexity of the GC-2( $t, p$ ) algorithms applied to the UEP code  $C_1(16, 5, 5)$**

		$p = 2$		$p = 3$		$p = 4$	
		$\Delta_i(\text{dB})$	MO [ $N_s - N_g$ ] [ $N_m - N_c$ ]	$\Delta_i(\text{dB})$	MO [ $N_s - N_g$ ] [ $N_m - N_c$ ]	$\Delta_i(\text{dB})$	MO [ $N_s - N_g$ ] [ $N_m - N_c$ ]
cp <sub>1</sub>	$t = 2$	2.0	[58.3 – 786.2] [766.2 – 3.9]	1.4	[101.4 – 1,556.1] [1,516.1 – 6.8]	0.8	[159.6 – 3,100.0] [3,000.0 – 10.6]
	$t = 3$	0.9	[59.5 – 787.4] [767.4 – 4.0]	0.5	[116.6 – 1,573.8] [1,533.8 – 7.8]	0.1	[218.6 – 3,130.0] [3,050.0 – 14.6]
	$t = 4$	0.6	[60.0 – 788.0] [768.0 – 4.0]	0.2	[119.6 – 1,575.6] [1,535.6 – 8.0]	0.1	[238.8 – 3,150.0] [3,070.0 – 15.9]
cp <sub>2</sub>	$t = 2$	1.1	[58.4 – 786.3] [766.3 – 3.9]	0.6	[102.0 – 1,556.8] [1,516.8 – 6.8]	0.3	[162.2 – 3,070.0] [3,000.0 – 10.8]
	$t = 3$	0.5	[59.7 – 787.7] [767.7 – 4.0]	0.1	[118.0 – 1,570] [1,530.0 – 7.9]	0.03	[223.2 – 3,130.0] [3,050.0 – 14.9]
	$t = 4$	0.4	[60.0 – 788.0] [768.0 – 4.0]	0.1	[119.9 – 1,575.8] [1,535.8 – 8.0]	0.0	[239.3 – 3,150.0] [3,070.0 – 15.9]

**Table 6 Results of performance and arithmetic complexity of the GC-2( $t, p$ ) algorithms applied to the UEP code  $C_2(25, 8, 5)$**

		$p = 2$		$p = 3$		$p = 5$	
		$\Delta_i(\text{dB})$	MO [ $N_s - N_g$ ] [ $N_m - N_c$ ]	$\Delta_i(\text{dB})$	MO [ $N_s - N_g$ ] [ $N_m - N_c$ ]	$\Delta_i(\text{dB})$	MO $0.8[N_s - N_g]$ [ $N_m - N_c$ ]
cp <sub>1</sub>	$t = 2$	4.1	[93.5 – 1,829.4] [1,797.4 – 3.9]	3.5	[126.3 – 3,595.6] [3,531.6 – 5.3]	2.7	[367.1 – 14,200] [13,980 – 15.3]
	$t = 3$	3.1	[95.0 – 1,831.0] [1,799.0 – 3.96]	2.6	[169.1 – 3,640.2] [3,576.2 – 7.0]	1.6	[577.5 – 14,400] [14,200 – 24.0]
	$t = 4$	2.2	[95.4 – 1,831.4] [1,799.4 – 3.98]	1.7	[183.2 – 3,654.8] [3,590.8 – 7.6]	0.8	[687.4 – 14,600] [14,300 – 28.6]
	$t = 5$	1.3	[95.7 – 1,831.7] [1,799.7 – 3.99]	0.9	[188.3 – 3,660.2] [3,596.2 – 7.8]	0.3	[742.7 – 14,630] [14,400 – 30.9]
	$t = 6$	1.0	[96.0 – 1,832.0] [1,800.0 – 4.0]	0.6	[191.4 – 3,663.4] [3,599.4 – 8.0]	0.2	[764.9 – 14,700] [14,400 – 31.9]
cp <sub>2</sub>	$t = 2$	1.9	[91.8 – 1,829.7] [1,797.7 – 3.9]	1.4	[126.2 – 3,593.5] [3,529.6 – 5.2]	0.6	[365.6 – 14,230] [14,000 – 15.2]
	$t = 3$	1.0	[94.7 – 1,830.8] [1,799.3 – 3.96]	0.6	[169.1 – 3,640.7] [3,576.6 – 7.0]	0.1	[596.7 – 14,500] [14,220 – 24.9]
	$t = 4$	0.6	[95.2 – 1,831.4] [1,800.6 – 3.99]	0.3	[182.9 – 3,655.0] [3,590.1 – 7.64]	0.06	[728.1 – 14,600] [14,400 – 30.3]
	$t = 5$	0.5	[95.5 – 1,832] [1,801.2 – 4.0]	0.3	[191.1 – 3,663.2] [3,598.8 – 7.9]	0.02	[763.4 – 14,600] [14,400 – 31.8]
	$t = 6$	0.5	[95.5 – 1,832] [1,801.2 – 4.0]	0.3	[191.2 – 3,663.9] [3,598.6 – 7.9]	0.02	[763.4 – 14,600] [14,400 – 31.8]

**Table 7 Values of optimal quantization step  $\delta_{op}$  of the class  $cp_1$  for the WED(2, Q) algorithm and different values of  $E_b/N_0$**

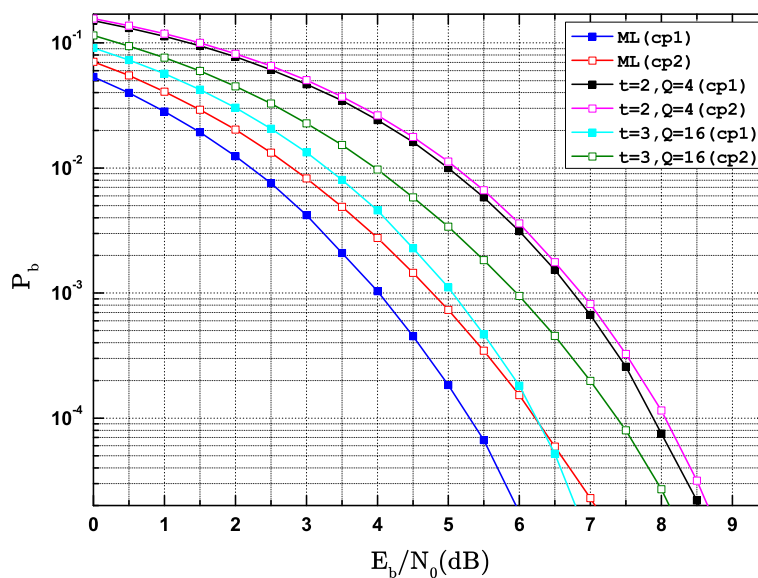
$E_b/N_0$ (dB)	$\delta_{op}$					
	$C_1(16, 5, 5)$			$C_2(25, 8, 5)$		
	Q = 4	Q = 16	Q = 1024	Q = 4	Q = 16	Q = 1024
0	0.31	0.07	0.0011	0.27	0.05	0.0011
1	0.37	0.09	0.0013	0.23	0.07	0.0011
2	0.37	0.09	0.0013	0.27	0.07	0.0011
3	0.37	0.09	0.0013	0.27	0.07	0.0011
4	0.39	0.09	0.0013	0.33	0.07	0.0011
5	0.45	0.09	0.0015	0.29	0.07	0.0011
6	0.47	0.11	0.0015	0.35	0.09	0.0013
7	0.51	0.11	0.0015	0.37	0.09	0.0013
8	0.67	0.11	0.0017	0.49	0.11	0.0017
9	0.67	0.15	0.0017	0.57	0.09	0.0019

in the value of  $\delta_{op}$ , reducing even more the spacing between adjacent regions. In addition, we can observe that the higher the value of Q, the lower the variation of  $\delta_{op}$  over the range of SNR considered. This behavior indicates that as Q increases, the optimal value of the quantization step becomes approximately constant.

Figure 4 shows the curves of  $P_b$  versus  $E_b/N_0$  of the WED(2, 4) and WED(3, 16) algorithms for both classes of the UEP code  $C_1$ . Similarly to observed in the GC-2(2, 2) algorithm, there is no performance difference between the two classes for the case  $t = 2$ , while for  $t = 3$ , the SNR difference to the ML algorithm is 0.9 dB ( $cp_1$ ) and 1.2 dB ( $cp_2$ ).

Table 8 summarizes the complexity-performance trade-off for various configurations of the WED algorithm applied to the UEP codes  $C_1$  and  $C_2$ . For each intersection of a row ( $t$  and  $cp_i$ ) with a column (Q and a code  $C_j$ ), we have  $\Delta_i$  (dB) for  $P_b = 10^{-4}$ . For code  $C_1$ , the error-correcting capability of the WED algorithm is  $t = 2, 3$  and 4, while for code  $C_2$ ,  $t = 2, 3, 4, 5$  and 6, as it was considered for the GC-2 algorithm. Thus, considering the complexity-performance trade-off, it is more advantageous to increase  $t$ , as in the GC-2( $t, p$ ) algorithm, than increase the number of quantization regions Q.

Finally, we compare both soft-decision decoding algorithms for a specific protection class, such as the higher



**Figure 4 Performance of ML and WED ( $t, Q$ ) decoding algorithms for UEP code  $C_1(16, 5, 5)$  considering the use of binary antipodal modulation and AWGN channel.**



**Table 8 Results of performance ( $\Delta_i$  in dB) and arithmetic complexity (MO) of the WED( $t, Q$ ) algorithms applied to the UEP codes  $C_1(16, 5, 5)$  and  $C_2(25, 8, 5)$**

MO	Q	$C_1(16, 5, 5)$			$C_2(25, 8, 5)$		
		4	16	1024	4	16	1024
		$[N_s - N_g]$	$[116 - 788]$	$[314 - 1,970]$	$[77 - 916]$	$[179 - 1,832]$	$[485 - 4,580]$
	$[N_m - N_c]$	$[388 - 82]$	$[776 - 148]$	$[1,940 - 346]$	$[904 - 127]$	$[1,808 - 229]$	$[4,520 - 535]$
cp <sub>1</sub>	$t = 2$	2.7	2.1	1.7	4.5	4.1	3.7
	$t = 3$	1.2	0.9	0.8	3.4	3.0	2.5
	$t = 4$	1.3	0.9	0.9	2.3	1.9	1.4
	$t = 5$	-	-	-	1.3	1.0	1.0
	$t = 6$	-	-	-	1.2	1.1	1.1
cp <sub>2</sub>	$t = 2$	1.9	1.4	1.2	2.4	2.0	1.8
	$t = 3$	1.0	0.8	0.7	1.7	1.4	1.3
	$t = 4$	1.5	0.8	0.7	1.5	1.3	1.2
	$t = 5$	-	-	-	1.6	1.4	1.3
	$t = 6$	-	-	-	1.6	1.4	1.3

protection one (cp<sub>1</sub>). To do this, we define a binary decoding ratio, denoted by  $\gamma$ , as

$$\gamma = \frac{2^p}{\log_2 Q}. \quad (10)$$

The parameters  $p$  and  $Q$  are associated with the number of binary decodings that the GC-2( $t, p$ ) and WED( $t, Q$ ) algorithms, respectively, execute. When decoding a received sequence, the GC-2( $t, p$ ) algorithm does  $2^p$  binary decodings, while the WED( $t, Q$ ) algorithm does  $\log_2 Q$  ones. Thus, for making a fair comparison of the algorithms, we choose configurations where  $\gamma \cong 1$ . In this case, the WED algorithm can offer a performance closer to the ML curve (for the higher protection class), but at the price of increased complexity. For  $\gamma = 1$  and code  $C_2$ , we can see this comparing GC-2(5, 2) and WED(5, 16) algorithms (see Tables 6 and 8). For the GC-2(5, 2) algorithm,  $\Delta_1 = 1.3$  dB and MO  $\simeq [95.7; 1,800; 1,800; 3.99]$ , while for the WED(5, 16) one,  $\Delta_1 = 1.0$  dB and MO  $\simeq [179; 1,832; 1,808; 229]$ . Another example is verified if the GC-2(4, 3) and WED(4, 1024) are compared ( $\gamma = 0.8$  and code  $C_2$ ). For the GC-2(4, 3) algorithm,  $\Delta_1 = 1.7$  dB and MO  $\simeq [183; 3,655; 3,591; 7.63]$ , while for the WED one,  $\Delta_1 = 1.4$  dB and MO  $\simeq [485; 4,580; 4,520; 535]$ . It should also be observed in Table 8 that the performance of the WED algorithm degrades when  $t$  is high. The authors conjecture that this behavior is due to some limitation of the reliability  $R'_\ell$  adopted.

## 6 Conclusions

In this study, the effectiveness of two sub-optimum soft-decision decoding algorithms (GC-2( $t, p$ ) and WED( $t, Q$ ) algorithms) was investigated for each protection class of UEP block codes using binary transmission over an

AWGN channel. It was verified the performance of both algorithms compared to that of the ML one. The behavior of the GC-2 algorithm was investigated for estimating the analog weight (Step 4) according to the variation of its parameters ( $t$  and  $p$ ), while the WED algorithm was examined for a new proposed reliability according to the variation of its parameters ( $t$  and  $Q$ ). To estimate the complexity of each algorithm, it was computed the number of arithmetic operations per decoded sequence. An analysis of the trade-off between performance and complexity of the algorithms was performed for each protection class assuming various configuration options. These analyses led us to conclude that, when choosing the parameters of the algorithms, the increase of the error-correcting capability of the binary decoder ( $t$ ) was more advantageous in both cases. In addition, choosing the values of  $p$  and  $Q$  such that  $\gamma$  is close to one (for a fixed value of  $t$ ), it was verified that the GC-2 algorithm is less complex, while the WED algorithm can offer (depending on the code adopted) a performance closer to the ML one.

## Endnotes

<sup>a</sup>Error pattern  $\mathbf{z}$  associated with the syndrome of the sequence  $\mathbf{y}^i$ .

<sup>b</sup>The complexity of decoding algorithms should be taken into consideration additional factors besides the arithmetic operations (like memory reads and writes). Since these factors are architecture dependent, we omit their contribution in this article.

## Competing interests

The authors declare that they have no competing interests.

## Author details

<sup>1</sup>Third Center for Integrated Air Defense and Air Traffic Control (CINDACTA III), 51250-020, Recife-PE, Brazil. <sup>2</sup>Department of Computing Systems, Centro de

Informática, Federal University of Pernambuco (CIn/UFPE), 50740-560, Recife-PE, Brazil. <sup>3</sup>Department of Electronics and Systems, Federal University of Pernambuco (DES/UFPE), 50740-530, Recife-PE, Brazil.

#### Acknowledgements

This study was supported in part by the State of Pernambuco Research Foundation (FACEPE) under Grant APQ-1060-3.04/10 and the Brazilian Council for Scientific and Technological Development (CNPq) under Grant 302535/2010-1.

Received: 6 August 2012 Accepted: 1 February 2013

Published: 21 February 2013

#### References

1. T-H Lin, WJ Kaiser, GJ Pottie, Integrated low-power communication system design for wireless sensor networks. *IEEE Commun. Mag.* **42**(12), 142–150 (2004)
2. E Niewiadomska-Szynkiewicz, P Kwasniewski, I Windyga, Comparative study of wireless sensor networks energy-efficient topologies and power save protocols. *J. Telecommun. Inf. Technol.* **3**, 68–75 (2009)
3. J Gómez-Vilardebó, Al Pérez-Neira, M Nájjar, Energy efficient communications over the AWGN relay channel. *IEEE Trans. Wirel. Commun.* **9**(1), 32–37 (2010)
4. Y Zhu, W Wu, J Pan, Y Tang, An energy-efficient data gathering algorithm to prolong lifetime of wireless sensor networks. *Comput. Commun.* **33**(5), 639–647 (2010)
5. SL Howard, C Schlegel, K Iniewski, Error control coding in low-power wireless sensor networks: when is ECC energy-efficient? *EURASIP. J. Wirel. Commun. Netw.* **2**, 1–14 (2006)
6. F Kienle, N Wehn, H Meyr, On complexity, energy- and implementation-efficiency of channel decoders. *IEEE Trans. Commun.* **59**(12), 3301–3310 (2011)
7. M Fossorier, S Lin, J Snyders, Reliability-based syndrome decoding of linear block codes. *IEEE Trans. Inf. Theory* **IT-44**, 388–398 (1998)
8. J Singh, D Pesch, Application of energy efficient soft-decision error control in wireless sensor networks. *Telecommun. Syst.* (Springer Netherlands) (2011) doi:10.1007/s11235-011-9588-z, pp. 1–11
9. YC Chang, SW Lee, R Komiya, A low complexity hierarchical QAM symbol bits allocation algorithm for unequal error protection of wireless video transmission. *IEEE Trans. Consum. Electron.* **55**(3), 1089–1097 (2009)
10. HX Nguyen, HH Nguyen, T Le-Ngoc, Signal transmission with unequal error protection in wireless relay networks. *IEEE Trans. Veh. Technol.* **59**(5), 2166–2178 (2010)
11. C Pimentel, RD Souza, BF Uchôa-Filho, ME Pellenz, Generalized punctured convolutional codes with unequal error protection. *EURASIP J. Adv. Signal Process.* **2008**, Art. ID 280831, 1–6 (2008)
12. S Borade, B Nakiboglu, L Zheng, Unequal error protection: an information-theoretic perspective. *IEEE Trans. Inf. Theory.* **55**(12), 5511–5539 (2009)
13. S Zhang, VKN Lau, A novel unequal error protection (UEP) scheme using D-STTD for multicast service. *IEEE Trans. Wirel. Commun.* **8**(2), 978–984 (2009)
14. SS Arslan, PC Cosman, LB Milstein, Coded hierarchical modulation for wireless progressive image transmission. *IEEE Trans. Veh. Technol.* **60**(9), 4299–4313 (2011)
15. K Kang, WJ Jeon, Differentiated protection to video layers to improve perceived quality. *IEEE Trans. Mobi. Comput.* **11**(2), 292–304 (2012)
16. N Thomos, NV Boulgouris, MG Strintzis, Wireless image transmission using turbo codes and optimal unequal error protection. *IEEE Trans. Image Process.* **14**(11), 1890–1901 (2005)
17. Q Qu, JW Modestino, An adaptive motion-based unequal error protection approach for real-time video transport over wireless IP networks. *IEEE Trans. Multimed.* **8**(5), 1033–1044 (2006)
18. H Ha, C Yim, Layer-weighted unequal error protection for scalable video coding extension of H.264/AVC. *IEEE Trans. Consum. Electron.* **54**(2), 736–744 (2008)
19. W Zhang, X Shao, M Torki, A HajShirMohammadi, IV Bajic, Unequal error protection codes for JPEG2000 images using short block length turbo codes. *IEEE Commun. Lett.* **15**(6), 659–661 (2011)
20. NN Tendolkar, CRP Hartman, Generalization of Chase algorithms for soft decision decoding of binary linear codes. *IEEE Trans. Inf. Theory.* **IT-30**(5), 714–721 (1984)
21. E Weldon Jr, Decoding binary block codes on Q-ary output channels. *IEEE Trans. Inf. Theory.* **17**(6), 713–718 (1971)
22. B Masnick, J Wolf, On linear unequal error protection codes. *IEEE Trans. Inf. Theory.* **IT-3**(4), 600–607 (1967)
23. WJ van Gils, On linear unequal error protection codes. EUT-Rep-82-WSK-02, Department of Mathematical and Computing Science. Eindhoven University of Technology, 1982
24. D Chase, A class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inf. Theory.* **IT-18**(1), 170–182 (1972)
25. WHJ Chen, MPC Fossorier, S Lin, in *Proceedings of the IEEE International Conference on Communications (ICC)*. Optimum quantizer design for the weighted erasure decoding algorithm (Vancouver, Canada, June 1999), pp. 838–842
26. RC Albuquerque, DC Cunha, C Pimentel, in *Proceedings of the IEEE 3rd Latin-American Conference on Communications (LATINCOM)*. An evaluation of the generalized Chase-2 algorithm applied to unequal error protection block codes (Belém-PA, Brazil, October 2011), pp. 1–6
27. AM Tenenbaum, Y Langsam, MJ Augenstein, *Data Structures Using C*. (Facsimile edition, Prentice Hall, 1989)

doi:10.1186/1687-6180-2013-28

Cite this article as: de Albuquerque et al.: On the complexity-performance trade-off in soft-decision decoding for unequal error protection block codes. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:28.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)