Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 280831, 6 pages
doi:10.1155/2008/280831

Research Article

Generalized Punctured Convolutional Codes with Unequal Error Protection

Cecilio Pimentel,[1] Richard Demo Souza,[2] Bartolomeu F. Uchôa-Filho,[3] and Marcelo Eduardo Pellenz[4]

[1] *Department of Electronics and Systems, Federal University of Pernambuco (UFPE), Recife, 50711-970, Brazil*
[2] *Department of Electronics, Federal Technological University of Paraná (UTFPR), Curitiba, 80230-901, Brazil*
[3] *Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianópolis, 88040-900, Brazil*
[4] *Graduate Program in Computer Science, Pontifical Catholic University of Paraná (PUCPR), Curitiba, 80215-901, Brazil*

Correspondence should be addressed to Richard Demo Souza, richard@utfpr.edu.br

Received 22 July 2008; Accepted 29 October 2008

Recommended by Azzedine Zerguine

We conduct a code search restricted to the recently introduced class of generalized punctured convolutional codes (GPCCs) to find good unequal error protection (UEP) convolutional codes for a prescribed minimal trellis complexity. The trellis complexity is taken to be the number of symbols per information bit in the "minimal" trellis module for the code. The GPCC class has been shown to possess codes with good distance properties under this decoding complexity measure. New good UEP convolutional codes and their respective effective free distances are tabulated for a variety of code rates and "minimal" trellis complexities. These codes can be used in several applications that require different levels of protection for their bits, such as the hierarchical digital transmission of video or images.

1. INTRODUCTION

In several applications of digital transmission (e.g., video [1], images [2], voice [3], and data [4]), the importance of different bits in the input sequence of the channel encoder often varies and certain blocks of this sequence need higher protection level than other blocks, when this sequence is transmitted through a noisy channel. An error-correcting code which provides a selective level of protection to the information bits possesses a property called unequal error protection (UEP). UEP can be obtained either by using separate coding schemes with distinct error-correction capability for each level of protection or by using a single code with UEP capability. The latter case can be achieved by block [5–7], convolutional [8–11], coded modulation schemes [12, 13], turbo [14], or LDPC [15] codes.

The convolutional codes of rate $R = k/n$ with UEP capabilities considered in the literature [8–11] are represented by their conventional trellis module, denoted by M_{conv}. This module consists of one trellis section with 2^{ν} initial states and 2^{ν} final states; each initial state is connected by 2^k directed edges to final states, and each edge is labeled with n bits. In general, a trellis module M for an (n, k) convolutional code C consists of n' trellis sections, 2^{ν_t} states at depth t, 2^{b_t} edges emanating from each state at depth t, and l_t bits labeling each edge from depth t to depth $t + 1$, for $0 \le t \le n' - 1$. The semi-infinite trellis used by the Viterbi algorithm to decode C consists of a concatenation of infinitely many copies of the trellis modules M. The trellis complexity of the module M for the code C, denoted by $\text{TC}(M)$, is defined as [16]

$$\text{TC}(M) = \frac{1}{k}\sum_{t=0}^{n'-1} l_t 2^{\nu_t + b_t} \qquad (1)$$

symbols per bit. In particular, $\text{TC}(M_{\text{conv}}) = (n/k)2^{\nu+k}$ symbols per bit.

The "minimal" trellis module, $\widetilde{M}$, for convolutional codes was developed in [16, 17]. This "minimal" structure has n sections and $l_t = 1$ bit per branch for all t. The state complexity ν_t and the branch complexity b_t at depth t will be denoted by $\widetilde{\nu}_t$ and $\widetilde{b}_t$, respectively. The state and the branch complexity profiles of the "minimal" trellis module are denoted by $\widetilde{\boldsymbol{\nu}} = (\widetilde{\nu}_0, \ldots, \widetilde{\nu}_{n-1})$ and $\widetilde{\mathbf{b}} = (\widetilde{b}_0, \ldots, \widetilde{b}_{n-1})$, respectively. It has been shown in those references that for many convolutional codes the trellis complexity $TC(\widetilde{M})$ of the "minimal" trellis module is considerably smaller than the trellis complexity $TC(M_{conv})$ of the conventional trellis module.

Recently, there has been a noticeable interest in the search of convolutional codes with fixed $TC(\widetilde{M})$ [18–21], as opposed to specifying a complexity measure usually associated to the conventional trellis module (e.g., the encoder memory). Such interest is motivated by the reduced trellis complexity of the "minimal" trellis module. For this purpose, a class of convolutional codes called *generalized punctured convolutional codes* (GPCCs) has been proposed in [19], and further code search results have been tabulated in [21]. The way GPCCs are described by a "matrix module" facilitates the code search procedure under the "minimal" trellis complexity. It should be mentioned that none of the codes in [19, 21] has been designed to have the UEP property.

In this work, we are interested in the behavior of the UEP capability of convolutional codes under a "minimal" trellis complexity measure. For that sake, we carry out a code search to obtain good GPCCs with the UEP property for a given "minimal" trellis complexity $TC(\widetilde{M})$. The standard methodology, as, for example, in [9], consists of listing good codes with the UEP property for a given memory order. The general idea is to have convolutional codes, of different code rates, with different "minimal" (decoding) trellis complexities and different dispositions of error protection levels among the information bits to serve a wide range of applications.

The remainder of this paper is organized as follows. In Section 2, we present some notations and give some definitions towards the UEP property of convolutional codes. Also, the class of GPCCs is briefly described. In Section 3, we describe the search procedure for good GPCCs with the UEP property and having $TC(\widetilde{M})$ as the decoding trellis complexity. Tables containing good GPCCs with different code rates, trellis complexities, and UEP profiles are given. Finally, Section 4 concludes the paper.

2. PRELIMINARIES

Consider a binary (n, k) convolutional code represented by a trellis module M (or by a trellis constructed from M), with $k \geq 2$. At time instant t, exactly k information bits $\mathbf{x}_t = x_t^0 \cdots x_t^{k-1}$ are absorbed by the encoder and the output sequence $\mathbf{y}_t = y_t^0 \cdots y_t^{n-1}$ with n bits is generated. The transmitted codeword is $\mathbf{y} = \mathbf{y}_0\mathbf{y}_1\mathbf{y}_2,\ldots$ when the information sequence $\mathbf{x} = \mathbf{x}_0\mathbf{x}_1\mathbf{x}_2,\ldots$ is fed into the convolutional encoder. Given an initial state in the trellis obtained from a trellis module M, each possible input sequence gives rise to a unique path (sequence of states) in the trellis which is associated to a codeword.

Let $\mathcal{S}$ be the set of all paths in the trellis that diverge from the all-zero path (leave the state 0), at a fixed time instant t, say $t = 0$, and remerge into the all-zero path exactly once at some time later. Each path in $\mathcal{S}$ constitutes an error event. Given a path σ in $\mathcal{S}$, we denote by $w_H(\sigma)$ the Hamming weight of the codeword corresponding to this path. Let $\mathbf{x}^i = x_0^i x_1^i x_2^i,\ldots$ be a sequence of information symbols at the ith position of $\mathbf{x}_t$, for $t = 0, 1, 2, \ldots$. The UEP capability of a convolutional encoder is measured by the effective free distance vector [8, 9] of length k (the number of possible UEP protection levels).

Definition 1. Let $\mathcal{S}^{(i)}$ be a subset of $\mathcal{S}$ formed by paths induced by information sequences such that $\mathbf{x}^i$ is a nonzero sequence. The effective free distance vector, denoted by $\mathbf{d}_{\text{eff}}$, of a convolutional encoder with trellis module M is

$$\mathbf{d}_{\text{eff}} = (d_0, d_1, \ldots, d_{k-1}), \tag{2}$$

where the ith effective free distance d_i is

$$d_i = \min_{\sigma \in \mathcal{S}^{(i)}} w_H(\sigma). \tag{3}$$

Clearly, the free Hamming distance of the convolutional code, namely d_{free}, is equal to the least entry of the vector $\mathbf{d}_{\text{eff}}$. This vector depends on the mapping between the information sequences and the codewords and, therefore, is related to the encoder. An encoder is said to possess an equal error protection (EEP) capability if all entries of $\mathbf{d}_{\text{eff}}$ are equal.

An (n, k) GPCC [19] is a periodically time-varying convolutional code of period n defined by the "matrix module" (defined in [16, equation (2.4)]) (only the nonzero rows are shown):

$$\begin{bmatrix} g^0_{\hat{\nu}_0} & & & & & & \\ \vdots & & g^{p-1}_{\hat{\nu}_{p-1}} & & & & \\ g^0_1 & \ddots & \vdots & g^p_{\hat{\nu}_p} & g^{p+1}_{\hat{\nu}_{p+1}} & & \\ g^0_0 & & g^{p-1}_1 & \vdots & \vdots & & g^{n-1}_{\hat{\nu}_{n-1}} \\ & & g^{p-1}_0 & g^p_1 & g^{p+1}_1 & \ddots & \vdots \\ & & & g^p_0 = 0 & g^{p+1}_0 & & g^{n-1}_1 \\ & & & & & & g^{n-1}_0 \end{bmatrix}, \tag{4}$$

where g_i^t are the binary generator scalars of the GPCC, for the phase index $t = 0, 1, \ldots, n-1$ and for $i = 0, 1, \ldots, \hat{\nu}_t$, with $\hat{\nu}_t$ being the memory order at phase t. Shown in (4) is a phase p which carries no information ($g_0^p = 0$). There are exactly $n - k$ such phases in the "matrix module" of a GPCC. This, and some other restrictions, which apply to $\hat{\nu}_t$, must be imposed in order for the convolutional code to be a GPCC [19]. As shown in [19], we can easily fix the value of certain generator scalars in (4) in such a way that the positions of the leading and trailing 1's in each row of the so-called scalar generator matrix [16], denoted by G_{scalar}, yield naturally to the minimal-span form [16]. This allows the predetermination of the value of $TC(\widetilde{M})$ for an ensemble of GPCCs. For further details on the GPCC class refer to [19].

Table 1: Good convolutional codes of rate 2/3 and their effective free distances.

$\text{TC}(\widetilde{M})$	d_{free}	$\mathbf{d}_{\text{eff}}$	$\widetilde{\mathbf{b}}$	$\widetilde{\boldsymbol{\nu}}$	$G(D)$
10	2	2, 4	(1, 0, 1)	(2, 2, 2)	[1 1 3; 2 0 1]
	3	3, 3	(1, 1, 0)	(2, 2, 2)	[3 1 0; 0 3 1]
20	2	2, 5	(1, 1, 0)	(3, 3, 3)	[1 2 0; 4 3 3]
	4	4, 4	(1, 1, 0)	(3, 3, 3)	[3 2 1; 4 3 2]
32	2	2, 6	(1, 1, 0)	(3, 4, 4)	[1 0 2; 6 7 1]
	4	4, 5	(1, 1, 0)	(3, 4, 4)	[1 2 3; 4 5 3]
40	4	4, 6	(1, 1, 0)	(4, 4, 4)	[5 1 2; 2 7 3]
	5	5, 5	(1, 1, 0)	(4, 4, 4)	[7 3 2; 4 7 1]

3. CODE SEARCH RESULTS

We conduct a refined code search within the class of GPCCs to obtain good convolutional codes with the UEP property for a prescribed "minimal" trellis complexity $\text{TC}(\widetilde{M})$. The state complexity $\widetilde{\nu}_t$ and branch complexity $\widetilde{b}_t$ of the "minimal" trellis module are calculated from the "matrix module" (or actually the scalar generator matrix) of the code in the minimal-span form, following the procedure described in [16]. Then $\text{TC}(\widetilde{M})$ is obtained from (1), with $l_t = 1$ for all t. For a given code rate and $\text{TC}(\widetilde{M})$, we provide distinct configurations of $\mathbf{d}_{\text{eff}}$ satisfying the UEP property. As an example, to illustrate the code search conducted in this work, consider the (4, 3) GPCC with matrix module defined by

$$\begin{bmatrix} \overline{1} & 0 & 0 & 0 \\ 0 & 1 & \overline{1} & 0 \\ \underline{1} & 1 & 0 & \overline{1} \\ 0 & \underline{1} & 0 & 0 \\ 0 & 0 & \underline{1} & 1 \end{bmatrix}, \tag{5}$$

where only the nonzero rows are shown. The underlined (leading) and the overlined (trailing) 1's in (5) are, respectively, the first nonzero entry and last nonzero entry in each row of G_{scalar}, which for this code is given by

$$G_{\text{scalar}} = \begin{bmatrix} \underline{1}\ 1\ 0\ \overline{1} & 0\ 0\ 0\ 0 & & \\ 0\ \underline{1}\ 0\ 0 & \overline{1}\ 0\ 0\ 0 & & \\ 0\ 0\ \underline{1}\ 1 & 0\ 1\ \overline{1}\ 0 & & \\ & \underline{1}\ 1\ 0\ \overline{1} & 0\ 0\ 0\ 0 & \\ & 0\ \underline{1}\ 0\ 0 & \overline{1}\ 0\ 0\ 0 & \\ & 0\ 0\ \underline{1}\ 1 & 0\ 1\ \overline{1}\ 0 & \\ & & \underline{1}\ 1\ 0\ \overline{1} & \\ & & 0\ \underline{1}\ 0\ 0 & \ddots \\ & & 0\ 0\ \underline{1}\ 1 & \end{bmatrix}. \tag{6}$$

Notice that no column in (6) contains more than one leading 1's or more than one underlined 1's, thus this matrix has the LR property, which is equivalent to being in the minimal span-form [22, Theorem 6.11].

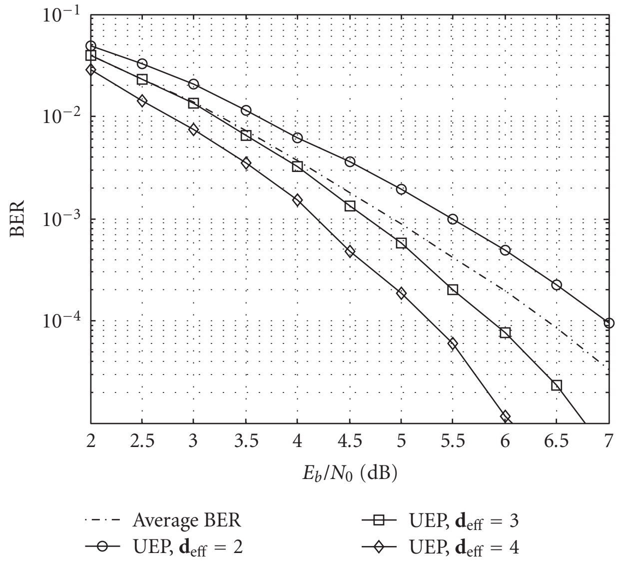

Figure 1: Simulated BER of each input bit versus E_b/N_0 over the AWGN channel for the (4, 3) GPCC with $\mathbf{d}_{\text{eff}} = (2, 3, 4)$. The average BER curve is also shown.

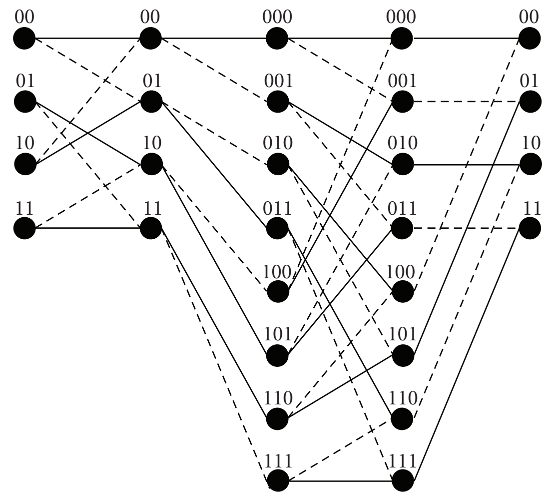

Figure 2: Minimum trellis module for the (4, 3) GPCC with $\mathbf{d}_{\text{eff}} = (2, 3, 4)$. The solid edges represent "0" codeword bits and the dashed edges represent "1" codeword bits.

According to the positions of the leading and trailing 1's, the "minimal" trellis module of this GPCC has state and branch complexity profiles given by $\widetilde{\boldsymbol{\nu}} = (2, 2, 3, 3)$ and $\widetilde{\mathbf{b}} = (1, 1, 1, 0)$, respectively, and trellis complexity $\text{TC}(\widetilde{M}) = 13.33$ symbols per bit. The trellis module for this GPCC, constructed according to the rules introduced in [22], is shown in Figure 2. The effective free distance vector of this code is $\mathbf{d}_{\text{eff}} = (2, 3, 4)$, therefore presenting the UEP property. In order to measure the bit-error rate (BER) corresponding to a particular encoder input, we simulated this code over the AWGN channel with one-sided noise power spectral density N_0, considering BPSK modulation with transmitted energy per information bit E_b and soft decision decoding using

Table 2: Good convolutional codes of rate 2/5 and their effective free distances.

$TC(\widetilde{M})$	d_{free}	$\mathbf{d}_{eff}$	$\widetilde{\mathbf{b}}$	$\widetilde{\boldsymbol{\nu}}$	$G(D)$
14	4	4, 6	(1, 0, 1, 0, 0)	(2, 2, 2, 2, 2)	[3 1 0 1 0; 2 2 3 1 1]
	5	5, 5	(1, 0, 1, 0, 0)	(2, 2, 2, 2, 2)	[3 1 1 1 0; 0 2 3 1 1]
28	6	6, 8	(1, 0, 1, 0, 0)	(3, 3, 3, 3, 3)	[3 3 2 1 0; 4 2 3 3 3]
	7	7, 7	(1, 0, 1, 0, 0)	(3, 3, 3, 3, 3)	[3 3 3 1 0; 4 2 1 3 3]
48	6	6, 9	(1, 0, 1, 0, 0)	(3, 4, 4, 4, 4)	[1 1 2 2 3; 4 6 5 3 3]
	7	7, 8	(1, 0, 1, 0, 0)	(3, 4, 4, 4, 4)	[1 1 3 3 3; 4 6 5 1 3]
72	8	8, 10	(1, 0, 1, 0, 0)	(4, 5, 4, 5, 4)	[7 6 6 7 0; 4 6 1 3 3]
	9	9, 9	(1, 0, 1, 0, 0)	(4, 5, 4, 5, 4)	[5 7 5 7 0; 6 4 3 3 3]

Table 3: Good convolutional codes of rate 3/4 and their effective free distances.

$TC(\widetilde{M})$	d_{free}	$\mathbf{d}_{eff}$	$\widetilde{\mathbf{b}}$	$\widetilde{\boldsymbol{\nu}}$	$G(D)$
9.33	2	2, 2, 4	(1, 1, 1, 0)	(2, 2, 2, 2)	[1 0 1 0; 2 1 0 0; 2 2 1 1]
	2	2, 3, 3	(1, 1, 1, 0)	(2, 2, 2, 2)	[1 0 1 0; 2 1 1 0; 0 2 1 1]
13.33	2	2, 3, 4	(1, 1, 1, 0)	(2, 2, 3, 3)	[1 1 0 1; 2 1 0 0; 0 2 3 1]
	3	3, 3, 3	(1, 1, 1, 0)	(2, 2, 3, 3)	[1 1 0 1; 2 1 1 0; 0 0 3 1]
37.33	2	2, 4, 5	(1, 1, 1, 0)	(4, 4, 4, 4)	[1 3 0 1; 0 1 2 0; 4 0 3 3]
	3	3, 3, 5	(1, 1, 1, 0)	(4, 4, 4, 4)	[3 3 1 0; 0 1 3 0; 4 2 3 3]
	4	4, 4, 4	(1, 1, 1, 0)	(4, 4, 4, 4)	[1 3 1 0; 0 3 2 1; 6 2 1 3]
42.67	2	2, 5, 5	(1, 1, 1, 0)	(4, 4, 4, 5)	[1 2 0 0; 0 3 2 3; 6 0 3 1]
	4	4, 4, 5	(1, 1, 1, 0)	(4, 4, 4, 5)	[3 2 0 1; 0 3 3 2; 4 0 1 3]

the Viterbi algorithm. Figure 1 shows the BER versus E_b/N_0 associated with each one of the three encoder inputs as well as the average BER. From the figure, we can see that the three inputs present quite different error protection levels. The input with effective free distance $d_2 = 4$ requires an E_b/N_0 of around 6 dB to achieve a BER of 10^{-5}, which is almost 1 dB better than the performance of the input with effective free distance $d_1 = 3$. By its turn, this input with $d_1 = 3$ performs around 1.5 dB better than the input with effective free distance $d_0 = 2$, at the same BER level.

By keeping fixed the underlined and overlined 1's in (5) or (6) we maintain the same $TC(\widetilde{M})$, even if we vary the other row elements between these underlined and overlined 1's. By doing so, then we can search through this template (which is defined by the fixed positions of the underlined and overlined 1's) for new GPCCs with the same $TC(\widetilde{M})$. Now consider the best (4, 3) GPCC in terms of d_{free} for the template in (6) or (5) with $TC(\widetilde{M}) = 13.33$ symbols per bit. This code has $\mathbf{d}_{eff} = (3, 3, 3)$. For such an EEP code, the error performances for the three inputs are very similar, requiring an E_b/N_0 of around 7 dB for achieving a BER of 10^{-5}. Therefore, the proposed GPCC with the UEP property can be directly applied to hierarchical sources not only with the same complexity cost but also without requiring any modifications in the topology of the "minimal" trellis used for decoding, increasing the implementation flexibility.

Similar results can be obtained for different code rates and trellis complexities. In Tables 1, 2, 3, 4, 5, a refined list of good GPCCs, and their respective effective distances $\mathbf{d}_{eff}$, is shown for different code rates and different values of $TC(\widetilde{M})$. The codes are specified by the polynomial generator matrix $G(D)$, which is given in octal form, where the highest power in D is in the most significant bit of the representation (e.g., $6 \equiv D + D^2$). Note that some of the codes in the tables are EEP codes, since in our search procedure we did not restrict ourselves to UEP codes. So, when an EEP code is listed in a table, it means that no GPCC with the UEP property could be found for that d_{free} and that $TC(\widetilde{M})$. From the analysis of the results listed in the tables we can make some remarks. First, consider the two codes with $TC(\widetilde{M}) = 10$ symbols per bit listed in Table 1. The second code in the table is an EEP code with $\mathbf{d}_{eff} = (3, 3)$, while the first one has $\mathbf{d}_{eff} = (2, 4)$. Therefore, in going from the second code to the first code, the effective free distance d_0 is decreased while d_1 is improved. In other words, the overall d_{free} had to be reduced in this case to produce an UEP code with the same $TC(\widetilde{M})$. We can find many of these examples in the tables. However, doubling the "minimal" trellis complexity to $TC(\widetilde{M}) = 20$ symbols per bit in this case may either increase further the protection of the second input bit or increase the overall d_{free} with an EEP code. As far as the protection of the second bit is concerned, in going from the code with $\mathbf{d}_{eff} = (2, 4)$ to the code with $\mathbf{d}_{eff} = (2, 5)$, thus doubling $TC(\widetilde{M})$, yields an E_b/N_0 improvement of 0.7 dB at a BER of 10^{-5} (simulation not shown).

The comparison of a code with the UEP property with another one having the EEP property with the same $TC(\widetilde{M})$ can be done for other rates as well. There are cases where the adoption of the UEP code yields a significant reduction of some of the effective free distances. For example, the case of $TC(\widetilde{M}) = 56$ symbols per bit in Table 5. The first code has $\mathbf{d}_{eff} = (2, 2, 2, 5)$, while the second code has $\mathbf{d}_{eff} = (4, 4, 4, 4)$.

Table 4: Good convolutional codes of rate 3/5 and their effective free distances.

$\mathrm{TC}(\widetilde{M})$	d_{free}	$\mathbf{d}_{\mathrm{eff}}$	$\widetilde{\mathbf{b}}$	$\widetilde{\boldsymbol{\nu}}$	$G(D)$
10.67	2	2, 4, 4	(1, 0, 1, 0, 1)	(2, 2, 2, 2, 2)	[1 1 0 0 0; 2 0 1 1 1; 0 2 0 2 3]
	3	3, 3, 4	(1, 0, 1, 0, 1)	(2, 2, 2, 2, 2)	[1 1 1 0 0; 2 0 1 1 0; 2 0 0 2 3]
24.00	3	3, 6, 3	(1, 1, 0, 1, 0)	(3, 3, 4, 3, 3)	[3 0 0 0 1; 2 3 3 1 0; 2 0 0 3 1]
	4	4, 5, 5	(1, 1, 0, 1, 0)	(3, 3, 4, 3, 3)	[3 1 0 1 1; 0 3 3 0 1; 0 0 2 3 1]
37.33	4	4, 6, 6	(1, 1, 0, 1, 0)	(4, 4, 4, 3, 4)	[3 2 1 1 1; 0 3 1 2 3; 0 0 2 3 1]
	5	5, 5, 5	(1, 1, 0, 1, 0)	(4, 4, 4, 3, 4)	[3 3 0 1 0; 0 1 3 3 2; 2 0 2 3 1]
48.00	4	4, 4, 7	(1, 1, 0, 1, 0)	(4, 4, 5, 4, 4)	[1 2 2 1 0; 0 1 1 2 1; 6 0 2 3 3]
	6	6, 6, 6	(1, 1, 0, 1, 0)	(4, 4, 5, 4, 4)	[1 3 2 1 1; 2 1 3 3 0; 6 2 2 3 3]

Table 5: Good convolutional codes of rate 4/5 and their effective free distances.

$\mathrm{TC}(\widetilde{M})$	d_{free}	$\mathbf{d}_{\mathrm{eff}}$	$\widetilde{\mathbf{b}}$	$\widetilde{\boldsymbol{\nu}}$	$G(D)$
14	2	2, 3, 3, 3	(1, 1, 1, 1, 0)	(2, 2, 3, 3, 3)	[1 0 0 1 0; 0 1 0 1 1; 2 0 1 0 1; 0 2 2 1 1]
	2	2, 2, 2, 4	(1, 1, 1, 1, 0)	(2, 2, 3, 3, 3)	[1 0 0 1 0; 0 1 0 0 1; 2 0 1 0 0; 2 2 2 1 1]
18	2	2, 2, 4, 4	(1, 1, 1, 1, 0)	(3, 3, 3, 3, 3)	[1 0 0 1 0; 2 1 0 1 1; 0 2 1 0 0; 0 2 2 1 1]
	3	3, 3, 3, 3	(1, 1, 1, 1, 0)	(3,3,3,3,3)	[1 0 1 1 0; 2 1 0 0 1; 2 2 1 0 0; 0 0 2 1 1]
24	2	2, 4, 4, 4	(1, 1, 1, 1, 0)	(3, 3, 3, 4, 4)	[1 1 1 0 1; 2 1 0 0 0; 0 2 1 1 1; 0 0 2 3 1]
	3	3, 3, 3, 4	(1, 1, 1, 1, 0)	(3, 3, 3, 4, 4)	[1 1 1 0 1; 2 1 0 1 0; 2 2 1 1 0; 2 0 2 3 1]
56	2	2, 2, 2, 5	(1, 1, 1, 1, 0)	(4, 4, 5, 5, 5)	[3 0 1 1 0; 0 3 3 1 1; 2 0 3 2 1; 0 0 2 3 3]
	4	4, 4, 4, 4	(1, 1, 1, 1, 0)	(4, 4, 5, 5, 5)	[3 1 0 1 0; 0 3 2 0 1; 0 0 3 2 1; 2 2 0 3 3]

Thus, in going from the second code to the first code three of the input bits had their effective free distances decreased by two in order to have the effective free distance of the fourth input bit increase by only one. On the other hand, we can also find examples where most of the effective free distances are improved, as is the case in going from the code with $\mathbf{d}_{\mathrm{eff}} = (5,5,5)$ to the code with $\mathbf{d}_{\mathrm{eff}} = (4,6,6)$, both with $\mathrm{TC}(\widetilde{M}) = 37.33$ symbols per bit, in Table 4.

It is also possible to improve the error protection of one input bit without sacrificing the error protection of the other input bit by increasing $\mathrm{TC}(\widetilde{M})$. This is the case in going from the code with $\mathbf{d}_{\mathrm{eff}} = (4,5)$ to the code with $\mathbf{d}_{\mathrm{eff}} = (4,6)$ in Table 1. Anyhow, the UEP requirements of the particular application will tell which code is more appropriate. For example, the two codes with $\mathrm{TC}(\widetilde{M}) = 14$ symbols per bit and the same d_{free} in Table 5 have quite different error protection profiles.

When considering the minimal trellis complexity $\mathrm{TC}(\widetilde{M})$ in the search criterion several codes which are not listed in the literature can be found. That is because when considering the typical encoder memory order criterion the leaps between available decoding complexity values are very large (the complexity doubles for each additional memory in the encoder). The $\mathrm{TC}(\widetilde{M})$-based criterion allows for an increased granularity in the list of decoding complexity values, as can be seen in Tables 1–5 in the manuscript. Therefore, the proposed criterion and the resultant UEP codes give a much larger flexibility for the system designer.

Finally, it is important to mention that $\mathbf{d}_{\mathrm{eff}}$ in this paper has been obtained from the conventional, time-invariant trellis of the convolutional codes. It is easy to see that if the conventional trellis is obtained from the same "matrix module" in (4) (in the minimal-span form) used to obtain the "minimal" trellis, then the mapping from information bits to coded bits inferred from the edge labeling in the "minimal" trellis module (see [16]) is the same as the one in the conventional trellis of the convolutional code. Therefore, $\mathbf{d}_{\mathrm{eff}}$ can be obtained from the conventional trellis, avoiding the explicit construction of the "minimal" trellis module.

4. CONCLUSIONS

This paper has considered the problem of providing unequal error protection to a binary data stream transmitted over a noisy channel via a single error-correcting code with low decoding complexity. It thus embraces a number of important applications that require different levels of protection for their bits. For this purpose, we have proposed the recently introduced class of generalized punctured convolutional codes and we have taken McEliece and Lin's [16] decoding complexity measure, namely, the number of symbols per information bit in the "minimal" trellis module for the code. A code search has been conducted and new good convolutional codes endowed with unequal error protection have been tabulated for a variety of code rates and "minimal" trellis complexities. It is seen through computer simulations that the E_b/N_0 required to achive a bit-error rate of 10^{-5} with a simple rate 3/4 convolutional code can differ in as much as 2 dB, depending on which of the three encoder inputs we are referring to, and the most protected encoder input requires about 1.5 dB lesser E_b/N_0 than what is required to achieve the same bit-error rate averaging over all encoder inputs. This example gives an idea of the unequal error protection capability of the new codes. The low decoding complexity, resulting from the adoption of the "minimal" trellis for the code, makes these codes attractive for practical applications.

ACKNOWLEDGMENTS

This work was supported by CNPq (Brazil) under Grants 304173/2007-0, 303938/2007-2, 303181/2007-9, 472977/2007-5, and also by FACEPE/CNPq under Grant APQ-0630-1.05/06.

REFERENCES

[1] Q. Qu, Y. Pei, and J. W. Modestino, "An adaptive motion-based unequal error protection approach for real-time video transport over wireless IP networks," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 1033–1044, 2006.

[2] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the internet using unequal error protection," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 121–131, 2003.

[3] A. Nazer and F. Alajaji, "Unequal error protection and source-channel decoding of CELP speech," *Electronics Letters*, vol. 38, no. 7, pp. 347–349, 2002.

[4] Z. C. Pereira, M. E. Pellenz, R. D. Souza, and M. A. A. Siqueira, "Unequal error protection for LZSS compressed data using Reed-Solomon codes," *IET Communications*, vol. 1, no. 4, pp. 612–617, 2007.

[5] B. Masnick and J. K. Wolf, "On linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 600–607, 1967.

[6] W. J. van Gils, "Linear unequal error protection codes from shorter codes," *IEEE Transactions on Information Theory*, vol. 30, no. 3, pp. 544–546, 1984.

[7] Z. Chen, P. Fan, and F. Jin, "New results on self-orthogonal unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 1141–1144, 1990.

[8] R. Palazzo Jr., "Linear unequal error protection convolutional codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '85)*, pp. 88–89, Brighton, UK, June 1985.

[9] R. Palazzo Jr., "On the linear unequal error protection convolutional codes," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '86)*, pp. 1367–1371, Houston, Tex, USA, December 1986.

[10] D. G. Mills and D. J. Costello Jr., "Using a modified transfer function to calculate unequal error protection capabilities of convolutional codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '93)*, p. 144, San Antonio, Tex, USA, January 1993.

[11] V. Pavlushkov, R. Johannesson, and V. V. Zyablov, "Unequal error protection for convolutional codes," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 700–708, 2006.

[12] A. R. Calderbank and N. Seshadri, "Multilevel codes for unequal error protection," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1234–1248, 1993.

[13] R. H. Morelos-Zaragoza, T. Kasami, S. Lin, and H. Imai, "On block-coded modulation using unequal error protection codes over rayleigh-fading channels," *IEEE Transactions on Communications*, vol. 46, no. 1, pp. 1–4, 1998.

[14] G. Caire and E. Biglieri, "Parallel concatenated codes with unequal error protection," *IEEE Transactions on Communications*, vol. 46, no. 5, pp. 565–567, 1998.

[15] N. Rahnavard, H. Pishro-Nik, and F. Fekri, "Unequal error protection using partially regular LDPC codes," *IEEE Transactions on Communications*, vol. 55, no. 3, pp. 387–391, 2007.

[16] R. J. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, part 1, pp. 1855–1864, 1996.

[17] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1663–1666, 1994.

[18] H.-H. Tang and M.-C. Lin, "On $(n, n-1)$ convolutional codes with low trellis complexity," *IEEE Transactions on Communications*, vol. 50, no. 1, pp. 37–47, 2002.

[19] B. F. Uchôa-Filho, R. D. Souza, C. Pimentel, and M.-C. Lin, "Generalized punctured convolutional codes," *IEEE Communications Letters*, vol. 9, no. 12, pp. 1070–1072, 2005.

[20] E. Rosnes and Ø. Ytrehus, "On maximum length convolutional codes under a trellis complexity constraint," *Journal of Complexity*, vol. 20, no. 2-3, pp. 372–403, 2004.

[21] B. F. Uchôa-Filho, R. D. Souza, C. Pimentel, and M. Jar, "Convolutional codes under a minimal trellis complexity measure," *IEEE Transactions on Communications*, vol. 57, no. 1, 2009.

[22] R. J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Transactions on Information Theory*, vol. 42, no. 4, pp. 1072–1092, 1996.