

Unequal Error Protection Punctured Convolutional Codes

Cecilio Pimentel, Richard Demo Souza, Bartolomeu F. Uchôa-Filho,
and Marcelo Eduardo Pellenz

Abstract—In this paper, we take the total number of edge symbols per information bit in the minimal trellis module representing a convolutional code as the measure of complexity to get it decoded by the Viterbi algorithm. We conduct a code search restricted to the recently introduced class of generalized punctured convolutional codes (GPCC) to find good unequal error protection (UEP) convolutional codes for a prescribed trellis complexity. New good UEP convolutional codes and their respective effective free distances are tabulated for a variety of code rates and “minimal” trellis complexities. The low decoding complexity, resulting from the adoption of the “minimal” trellis for the code, makes these codes attractive for practical applications.

Index Terms—Convolutional codes, decoding complexity, puncturing, Viterbi algorithm.

I. INTRODUCTION

An error correcting code which provides a selective level of protection to the information bits possesses a property called unequal error protection (UEP) [1]–[3]. It thus embraces a number of important applications that require different levels of protection for their bits [4]–[6]. The convolutional codes of rate $R = k/n$ with UEP capabilities considered in the literature [7]–[9] 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. Punctured convolutional codes (PCCs) [10], on the other hand, form a special class of (n, k) convolutional codes that can be described by a low-complexity trellis module. For rate $R > 1/2$, PCCs can be obtained by puncturing a rate $1/2$ periodically time-varying convolutional code [11] called *mother* code. Amat *et al.* [12] have proposed recursive $(n, n-1)$ convolutional codes with excellent distance spectrum and high decoding complexity, as noted by Tang *et al.* [13].

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 $0 \leq t \leq n'-1$. 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 . McEliece and Lin [14] stated that the computational effort required by the Viterbi algorithm to decode a convolutional code is proportional to the total number of edge symbols in the trellis module representing the code. This is said to be the *trellis complexity of the module M* for the convolutional code C , denoted by $TC(M)$, and according to [14] it is defined as

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

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

Almost four decades after the inception of convolutional codes by Elias [15], and when PCCs were about fifteen years old, a theory of minimal trellis for convolutional codes was developed by Sidorenko and Zyablov [16] and McEliece and Lin [14]. Unique (up to isomorphism), the minimal trellis module, \tilde{M} , for the convolutional code C minimizes, among various complexity measures, the number of states at each depth and the total number of branches [17]. For this minimal structure, the state complexity ν_t and the branch complexity b_t at depth t will be denoted by $\tilde{\nu}_t$ and \tilde{b}_t , respectively. The *minimal trellis module* for the (n, k) convolutional code C consists of $n' = n$ trellis sections, k of which has $\tilde{b}_t = 1$ and the remaining $(n-k)$ trellis sections are informationless, i.e., a single edge leaves each state ($\tilde{b}_t = 0$). There are $2^{\tilde{\nu}_t}$ states at depth t , and $l_t = 1$ bit per branch for all t . The state and the branch complexity profiles of the “minimal” trellis module are denoted by $\tilde{\nu} = (\tilde{\nu}_0, \dots, \tilde{\nu}_{n-1})$ and $\tilde{\mathbf{b}} = (\tilde{b}_0, \dots, \tilde{b}_{n-1})$, respectively.

It has been shown in [14] that for many convolutional codes the trellis complexity $TC(\tilde{M})$ of the “minimal” trellis module is considerably smaller than the trellis complexity $TC(M_{conv})$ of the conventional trellis module. Bearing in mind that the minimal trellis module is inherent to the convolutional code, not to a particular way of encoding or decoding it, we henceforth refer to $TC(\tilde{M})$ as the *trellis complexity of the convolutional code C* .

A study of *equivalent* convolutional codes recently presented by Tang *et al.* [13] reinforces the appropriateness of this measure of complexity, as opposed to the memory size of the encoder; the equivalence is in the sense that the minimal trellis modules of equivalent codes are cyclic shift versions of one another, this all codes have the same $TC(\tilde{M})$, although they have been generated by minimal encoders of distinct memory sizes.

Cecilio Pimentel is with CODEC/DES/UFPE, Recife-PE, Brazil, email: cecilio@ufpe.br. Richard Demo Souza is with DAELN/CPGEI/UTFPR, Curitiba-PR, Brazil, email: richard@cpgei.cefetpr.br. Bartolomeu F. Uchôa-Filho is with GPqCom/EEL/CTC/UFSC, Florianópolis-SC, Brazil, email: uchoa@eel.ufsc.br. Marcelo E. Pellenz is with PPGIA/PUCPR, Curitiba-PR, Brazil, email: marcelo@ppgia.pucpr.br.

This work was supported by CNPq (Brazil) under grants 301253/2004-8, 303938/2007-2, and 472977/2007-5, and by FACEPE/CNPq under grant APQ-0630-1.05/06.

In [18], the authors have found good convolutional codes by introducing the class of *generalized punctured convolutional codes* (GPCCs). It was shown that the class of GPCCs is sufficiently broad to contain good codes, e.g. it encompasses the class of PCCs and the majority of codes found in [19]. Also, we can easily control the spanlength of each row of the GPCC generator matrix, a property that allows us to search for GPCCs with fixed $TC(\widetilde{M})$.

In this work, we are interested in the behavior of the UEP capability of convolutional codes under a “minimal” trellis complexity measure. We then propose GPCCs and modify the code search to endow these codes with UEP capability.

II. 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.

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 $\mathbf{x}^i = x_0^i x_1^i x_2^i \cdots$ be a sequence of information symbols at the i th position of \mathbf{x}_t , for $t = 0, 1, 2, \dots$. The UEP capability of a convolutional encoder is measured by the effective free distance vector [7], [8] 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 non-zero sequence. The effective free distance vector, denoted by \mathbf{d}_{eff} , of a convolutional encoder with trellis module M is

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

where the i th effective free distance d_i is

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

Clearly, the free Hamming distance of the convolutional code, namely d_{free} , is equal to the least entry of the vector \mathbf{d}_{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}_{eff} are equal.

An (n, k) GPCC [18] is a periodically time-varying convolutional code of period n defined by the “matrix module” [14, eq. (2.4)] (only the non-zero rows are shown)

$$\left[\begin{array}{cccccc} g_{\hat{\nu}_0}^0 & & & & & \\ \vdots & & & & & \\ g_1^0 & \cdots & \vdots & g_{\hat{\nu}_p}^p & g_{\hat{\nu}_{p+1}}^{p+1} & \\ g_0^0 & g_1^{p-1} & \vdots & \vdots & \vdots & g_{\hat{\nu}_{n-1}}^{n-1} \\ & g_0^{p-1} & g_1^p & g_1^{p+1} & \ddots & \vdots \\ & & g_0^p = 0 & g_0^{p+1} & & g_1^{n-1} \\ & & & & & g_0^{n-1} \end{array} \right] \quad (4)$$

where g_i^t are the binary generator scalars of the GPCC, for the phase index $t = 0, 1, \dots, n-1$ and for $i = 0, 1, \dots, \hat{\nu}_t$, 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. For reasons explained in [18], these parameters are required to satisfy the restrictions:

- $g_0^t = 0$ for all $t \in J$, and $g_0^t = 1$ for all $t \in I \setminus J$, where J is some subset of size $n-k$ of the set $I = \{0, 1, \dots, n-1\}$ and it represents the indexes of the informationless phases (this is illustrated in (4) with phase $p \in J$).
- $\hat{\nu}_{t+1} \leq \hat{\nu}_t + \hat{b}_t$, for $t = 0, 1, 2, \dots, n-2$, and $\hat{\nu}_0 \leq \hat{\nu}_{n-1} + \hat{b}_{n-1}$, where $\hat{b}_t = 0$ if $t \in J$, and $\hat{b}_t = 1$ if $t \in I \setminus J$.

This class of convolutional codes is broad enough to contain good codes and yet has structural properties that facilitate the code search; for instance, one can easily control the spanlength of each row of the GPCC generator matrix, a property that allows us to search for GPCCs with fixed $TC(\widetilde{M})$.

III. 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 $TC(\widetilde{M})$. Then $TC(\widetilde{M})$ is obtained from (1), with $l_t = 1$ for all t . For a given code rate and $TC(\widetilde{M})$, we provide distinct configurations of \mathbf{d}_{eff} satisfying the UEP property. By placing the leading and trailing “ones” of each row of G_{scalar} in specific positions, while others are set free to assume any binary value, we can define ensembles of GPCCs with generator matrix in trellis oriented form [20, p. 343] with a particular trellis complexity. Consider the template (only the non-zero rows are shown)

$$\left[\begin{array}{cccc} \overline{1} & 0 & 0 & 0 \\ * & * & \overline{1} & 0 \\ \underline{1} & * & * & \overline{1} \\ 0 & \underline{1} & * & * \\ 0 & 0 & \underline{1} & * \end{array} \right] \quad (5)$$

for the matrix module of a (4,3) GPCC, where the nontrivial binary entries are marked by asterisk. According to the positions of the leading (underlined) and trailing (overlined) 1’s of each row of (5), the state and the branch complexity profiles of the minimal trellis module are $\bar{\nu} = (2, 2, 3, 3)$ and $\bar{b} = (1, 1, 1, 0)$. Therefore, all GPCCs within this ensemble have the same trellis complexity $TC(\widetilde{M}) = 13.33$ symbol per bit. The effective free distance vector of this code is $\mathbf{d}_{\text{eff}} = (2, 3, 4)$, therefore presenting the UEP property. We may vary in each row of (5) the entries between the underlined and overlined 1’s (while these are kept fixed) and search through this template 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 (5) with $TC(\widetilde{M}) = 13.33$ symbols per bit. This code has $\mathbf{d}_{\text{eff}} = (3, 3, 3)$. Similar results can be obtained for different code rates and trellis complexities.

In Tables I-III, a refined list of good GPCCs, and their respective effective distances 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 I. The second code in the table is an EEP code with $d_{\text{eff}} = (3, 3)$, while the first one has $d_{\text{eff}} = (2, 4)$. 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.

Finally, it is important to mention that d_{eff} in this paper has been obtained from the conventional, time-invariant trellis of the convolutional codes. In [13], the authors showed that the distance spectrum of the same convolutional code obtained from different trellis representations agree only for the first few terms, and may be slightly different for the other terms. It is argued in [13] that a more accurate distance spectrum of a convolutional code is obtained from its minimal trellis. However, as already mentioned, to reduce the computational effort in our code search we avoid the actual construction of the minimal trellis.

IV. 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. For this purpose, we have proposed the recently introduced class of generalized punctured convolutional codes and we have taken McEliece and Lin’s 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.

REFERENCES

- [1] B. Masnick and J. K. Wolf, “On linear unequal error protection codes,” *IEEE Trans. Inform. Theory*, vol. 13, no. 4, pp. 600-607, Oct. 1967.
- [2] W. J. van Gils, “Linear unequal error protection codes from shorter codes,” *IEEE Trans. Inform. Theory*, vol. 30, no. 3, pp. 544-546, May 1984.
- [3] Z. Chen, P. Fan, and F. Jin, “New results on self-orthogonal unequal error protection codes,” *IEEE Trans. Inform. Theory*, vol. 36, no. 5, pp. 1141-1144, Sept. 1990.

- [4] 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 Trans. Multimedia*, vol. 8, no. 5, pp. 1033-1044, Oct. 2006.
- [5] A. Nazer and F. Alajaji, “Unequal error protection and source-channel decoding of CELP speech,” *Electro. Letters*, vol. 38, no. 7, pp. 347-349, Mar. 2002.
- [6] 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, Aug. 2007.
- [7] R. Palazzo Jr., “Linear unequal error protection convolutional codes,” in *Proc. IEEE Int. Symp. Inform. Theory*, Brighton, U.K., Jun. 1985, pp. 88-89.
- [8] R. Palazzo Jr., “On the linear unequal error protection convolutional codes,” in *Proc. IEEE Global Telecom. Conf.*, Houston, TX, Dec. 1986, pp. 1367-1371.
- [9] V. Pavlushkov, R. Johannesson, and V. V. Zyablov, “Unequal error protection for convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 700-708, Feb. 2006.
- [10] I. E. Bocharova and B. D. Kudryashov, “Rational rate punctured convolutional codes for soft-decision Viterbi decoding,” *IEEE Trans. Inform. Theory*, vol. 43, no. 4, pp. 1305-1313, July 1997.
- [11] R. D. Souza and B. F. Uchôa-Filho, “On trellis modules for convolutional codes,” in *2002 IEEE International Symposium on Information Theory*, p. 417, Lausanne, Switzerland, July 2002.
- [12] A. G. i Amat, G. Montorsi, and S. Benedetto, “A new approach to the construction of high-rate convolutional codes,” *IEEE Communications Letters*, vol. 5, no. 11, pp. 453-455, Nov. 2001.
- [13] H.-H. Tang, M.-C. Lin, and B. F. Uchôa-Filho, “Minimal trellis modules and equivalent convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 2, no. 8, pp. 3738-3746, Aug. 2006.
- [14] R. J. McEliece and W. Lin, “The trellis complexity of convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1855-1864, Nov. 1996.
- [15] P. Elias, “Coding for noisy channels,” *IRE Conv. Rec.*, Part 4, pp. 37-47, 1955.
- [16] V. Sidorenko and V. Zyablov, “Decoding of convolutional codes using a syndrome trellis,” *IEEE Trans. Inform. Theory*, vol. 40, no. 5, pp. 1663-1666, Sept. 1994.
- [17] A. Vardy, “Trellis structure of codes,” in *Handbook of Coding Theory*, vol. II, (V.S. Pless and W.C. Huffman, eds.), North-Holland, Amsterdam, The Netherlands (1998).
- [18] B. F. Uchôa-Filho, R. D. Souza, C. Pimentel, and M.-C. Lin, “Generalized punctured convolutional codes,” *IEEE Commun. Letters*, vol. 9, no. 12, pp. 1070-1072, Dec. 2005.
- [19] H.-H. Tang and M.-C. Lin, “On $(n, n-1)$ convolutional codes with low trellis complexity,” *IEEE Trans. Commun.*, vol. 50, no. 1, pp. 37-47, Jan. 2002.
- [20] S. Lin and D. J. Costello, Jr., *Error Control Coding*, Prentice Hall, 2nd edition, 2004.

TABLE I
GOOD CONVOLUTIONAL CODES OF RATE 2/3 AND THEIR EFFECTIVE FREE DISTANCES.

$TC(\tilde{M})$	d_{free}	d_{eff}	\tilde{b}	$\tilde{\nu}$	$G(D)$
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]

TABLE II
GOOD CONVOLUTIONAL CODES OF RATE 3/4 AND THEIR EFFECTIVE FREE DISTANCES.

$TC(\tilde{M})$	d_{free}	d_{eff}	\tilde{b}	$\tilde{\nu}$	$G(D)$
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]

TABLE III
GOOD CONVOLUTIONAL CODES OF RATE 4/5 AND THEIR EFFECTIVE FREE DISTANCES.

$TC(\tilde{M})$	d_{free}	d_{eff}	\tilde{b}	$\tilde{\nu}$	$G(D)$
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]