Convolutional Codes Search Based on Minimal Trellis Complexity

Bartolomeu F. Uchôa-Filho, Richard Demo Souza, Cecilio Pimentel and Mao-Chao Lin

***Abstract*— This paper considers convolutional codes with low trellis complexity (i.e., low Viterbi decoding complexity) and good distance spectrum (i.e., good error performance). The trellis complexity of the convolutional code is defined as the total number of edge symbols per information bit in the minimal trellis module representing the code. We introduce a class of convolutional codes, called generalized punctured convolutional codes (GPCCs), which is broader than and encompasses the class of the standard punctured convolutional codes. A code in this class can be represented by a trellis module, the GPCC trellis module, whose topology resembles that of the minimal trellis module. The GPCC trellis module for a punctured convolutional code is shown to be isomorphic to the minimal trellis module. It is also shown by means of examples that this class contains codes with better distance spectrum than the best known punctured convolutional codes with the same code rate and trellis complexity. Good GPCCs obtained with the aid of a computer search are presented.**

I. Introduction

A convolutional code can be represented by a semi-infinite trellis consisting, after a short transient, of concatenated copies of a topological structure called *trellis module*. Specifically, a trellis module M for a rate $R = k/n$ (i.e., a (n,k)) convolutional code C consists of n' trellis sections (from depth 0 to depth n'), 2^{ν_t} states at depth t, 2^{b_t} branches emanating from each state at depth t (for $0 \leq t \leq n'-1$), and l_t bits labeling each edge from depth t to depth $t+1$ (for $0 \leq t \leq n'-1$). McEliece and Lin [1] 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 [1] it is defined as:

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

Bartolomeu F. Uchôa-Filho is with GPqCom, EEL, CTC, UFSC, Florianópolis - SC. e-mail: uchoa@eel.ufsc.br

Richard Demo Souza is with CPGEI, CEFET/PR, Curitiba - PR. e-mail: richard@cpgei.cefetpr.br

Cecilio Pimentel is with CODEC, DES, UFPE, Recife - PE. e-mail: cecilio@ufpe.br

Mao-Chao Lin is with the Department of EE, NTU, Taiwan, R.O.C. e-mail: mclin@cc.ee.ntu.edu.tw

symbols per bit. There can be many trellis modules describing the same code. The *conventional* trellis module for the (n,k) convolutional code C, denoted by M_{conv}, consists of a single trellis section (i.e., $n' = 1$), 2^ν initial states and 2^ν final states; each initial state is connected by 2^k directed branches to final states, and each edge is labeled with n bits. The trellis complexity of M_{conv} is then given by $TC(M_{conv}) = (n/k)\, 2^{\nu+k}$ symbols per bit.

While every convolutional code can be decoded by using the conventional trellis module, punctured convolutional codes (PCCs) [2]–[4] form a special class of (n,k) convolutional codes that can be described by an alternative, low-complexity trellis module, namely, the PCC trellis module (M_{PCC}). For rate $R > 1/2$, PCCs can be obtained by puncturing a rate 1/2 periodically time-varying convolutional code (PTVCC) [5] called *mother* code. The trellis module M_{PCC} then consists of $n' = k$ trellis sections, 2^ν states and two branches emanating from each state at depth t (i.e., $b_t = 1$), $l_t = 1$ bit labeling each edge in $2k-n$ trellis sections, and $l_t = 2$ bits labeling each edge in $n-k$ trellis sections. The corresponding trellis complexity of M_{PCC} is $TC(M_{PCC}) = (n/k)\, 2^{\nu+1}$ symbols per bit. Some of the best known PCCs were tabulated in [4].

A theory of minimal trellis for convolutional codes has been developed by Sidorenko and Zyablov [6] and McEliece and Lin [1]. Unique (up to isomorphism), the minimal trellis module, $\widetilde{M}$, for the convolutional code C minimizes, among various complexity measures, the number of states at each depth and the total number of branches (see [7] for instance). For this minimal structure, 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 minimal trellis module for the (n,k) convolutional code C consists of $n' = n$ trellis sections, k of which has $\widetilde{b}_t = 1$ and the remaining $(n-k)$ trellis sections are informationless, i.e., a single edge leaves each state ($\widetilde{b}_t = 0$). There are $2^{\widetilde{\nu}_t}$ states at depth t, and $l_t = 1$ for all t. Since a low-complexity Viterbi decoder is desirable, we adopt henceforth the trellis complexity of the minimal trellis module, $TC(\widetilde{M})$, as the *trellis complexity of the convolutional code* C.

In this paper, we search for good (in a distance spectrum sense) (n, k) convolutional codes with fixed $TC(\widetilde{M})$. It appears that a convolutional code search taking this measure of complexity has only been considered in the literature by Tang and Lin [8]. The convolutional codes they found, all of which of rate $(n-1)/n$, had better distance spectrum than the PCCs in [4], with the same rate and trellis complexity. Herein, we aim at finding convolutional codes better than PCCs for other code rates as well. To achieve this goal, we introduce a sufficiently broad class of convolutional codes, namely, the *generalized punctured convolutional codes* (GPCCs), which encompasses the class of PCCs. A code in this class can be represented by a trellis module — the GPCC trellis module (M_{GPCC}) — that shares all of the topological characteristics of the minimal trellis, as listed above, except possibly the minimality property. The GPCC trellis module is guaranteed to be isomorphic to the minimal trellis module if the code it represents is a PCC (see details in Sec. II).

There are two reasons for considering the class of GPCCs in our code search. First, many of the good $(n, n-1)$ convolutional codes found by Tang and Lin [8] are in fact GPCCs — we can show this by performing row operations on the scalar generator matrix of the code and turning this matrix into a GPCC form (details will be given in Section III). This means that the class of GPCCs is likely to contain good codes for other code rates as well. Second, it is possible to define a template for the scalar generator matrix of the GPCC which yields naturally to the minimal-span form [1], and one can easily control the spanlength of each row, predetermining the value of $TC(\widetilde{M})$ for an ensemble of GPCCs. This property makes it possible to search for codes with fixed $TC(\widetilde{M})$. As a result, we have found some GPCCs better than PCCs in [4], with the same rate and trellis complexity.

The remainder of this paper is organized as follows. In Section II, we introduce the class of GPCCs. Then, in Section III, we describe the computer code search and present a table containing good GPCCs. Finally, in Section IV, we conclude the paper.

II. GPCCs

In this section we introduce the class of generalized punctured convolutional codes. We begin by considering the class of PCCs. For simplicity, let us consider a (3,2) PCC of memory size $\nu = 2$, obtained by puncturing a rate 1/2 PTVCC of period 2. The scalar generator matrix

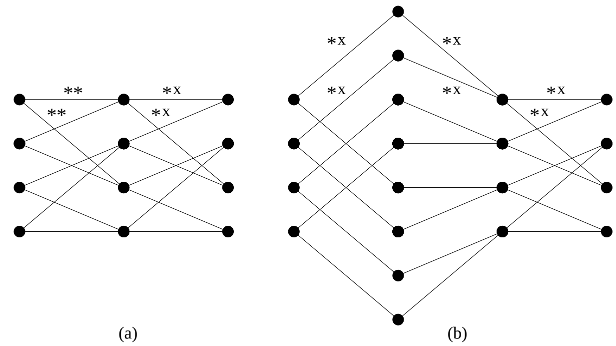

Fig. 1. (a) The PCC trellis module for a (3,2) PCC, and (b) the GPCC trellis module for the same code.

is then given by:

$$G_{scalar} = \begin{bmatrix} \ddots & & & & & \\ & G_2^1 & & & & \\ & G_1^1 & G_2^0 & & & \\ & G_0^1 & G_1^0 & G_2^1 & & \\ & & G_0^0 & G_1^1 & G_2^0 & \\ & & & G_0^1 & G_1^0 & \\ & & & & G_0^0 & \\ & & & & & \ddots \end{bmatrix}$$

where the generator submatrices G_i^t at time t, $t \in \{0, 1\}$, are of one of the following two forms: $G_i^0 = [* \ x]$, $G_i^1 = [* \ *]$ or $G_i^0 = [* \ *]$, $G_i^1 = [* \ x]$, where the nontrivial binary entries are marked by asterisk and x denotes a punctured position. The trellis module M_{PCC} for this code is shown in Fig. 1(a), where we have assumed without loss of generality that $G_i^0 = [* \ *]$, $G_i^1 = [* \ x]$. We can now regroup the entries of the same scalar generator matrix so that it can look like:

$$G_{scalar} = \begin{bmatrix} \ddots & & & & & \\ & \widehat{G}_2^2 & & & & \\ & \widehat{G}_1^2 & \widehat{G}_2^0 & \widehat{G}_3^1 & & \\ & \widehat{G}_0^2 & \widehat{G}_1^0 & \widehat{G}_2^1 & \widehat{G}_2^2 & \\ & & \widehat{G}_0^0 & \widehat{G}_1^1 & \widehat{G}_1^2 & \\ & & & [0 \ x] & \widehat{G}_0^2 & \\ & & & & & \ddots \end{bmatrix}$$

where now all shown generator submatrices are of the form $\widehat{G}_i^t = [* \ x]$. This is the GPCC form of the scalar generator matrix of this PCC. It should be noted that at time $t = 1$ $\widehat{G}_0^1 = [0 \ x]$, and at time $t = 2$ $\widehat{G}_0^2$ is placed to the right of $\widehat{G}_0^1$ (and not following the diagonal, as usual). This means that the same information bit would feed the time-varying encoder at times $t = 1$ and $t = 2$. In effect, there is only one edge leaving each state at depth $t = 1$ in the GPCC trellis module. The output branch label is the same as that produced by a zero information bit at time $t = 1$ in a PCC. At times $t = 0$ and $t = 2$, there are two branches leaving each state. Finally, note that at time $t = 1$ the generator submatrix $\widehat{G}_3^1$ is nontrivial. The trellis section with two-bit branch labels in Fig. 1(a) has been replaced by two trellis sections with one-bit branch label, and there are 8 states at depth $t = 1$ in the trellis module M_{GPCC} for this code, which is shown in Fig. 1(b).

In general, in the trellis module M_{PCC} for a (n, k) PCC with memory size ν taken from [4], the $n - k$ non-punctured trellis sections with state complexities $\nu_t = \nu_{t+1} = \nu$ is replaced by two punctured trellis sections in the trellis module M_{GPCC}, comprehending the times t, $t+1$, and $t+2$, where the corresponding state complexities are $\widehat{\nu}_t = \widehat{\nu}_{t+2} = \nu$ and $\widehat{\nu}_{t+1} = \nu + 1$. The $2k-n$ punctured trellis sections in M_{PCC} are replicated to the GPCC trellis module.

Let us now define the class of GPCCs. Consider the set $\mathcal{C}'$ containing all rate 1/2 PTVCCs of period k, where the memory size vary from one phase to another within a period and satisfies a topological constraint, namely, if $\boldsymbol{\nu}' = \{\nu_0, \nu_1, \ldots, \nu_{k-1}\}$ is the set of memory sizes of a PTVCC in $\mathcal{C}'$ then $\nu_{t+1} \leq \nu_t + 1$, for $0 \leq t < k-1$, and $\nu_0 \leq \nu_{k-1} + 1$. The class of (n, k) GPCCs is precisely the set of codes obtained by puncturing the codes in $\mathcal{C}'$ in exactly $2k - n$ positions within a period. As a particular case, if $\nu_t = \nu$ for all $t \in \{0, 1, \ldots, k-1\}$, then the GPCC is a PCC.

The GPCC trellis module for a (n, k) GPCC has state complexity profile $\widehat{\boldsymbol{\nu}} = \{\widehat{\nu}_0, \widehat{\nu}_1, \ldots, \widehat{\nu}_{n-1}\}$, which is related to $\boldsymbol{\nu}'$ as follows. Let u_t denote the number of non-punctured phases of the PTVCC (to give origin to the GPCC) occurring prior to phase t. By convention, $u_0 = 0$. Then, for $0 \leq t \leq k-1$, set $\widehat{\nu}_{t+u_t} = \nu_t$ if the t-th phase of the PTVCC has been punctured, and set $\widehat{\nu}_{t+u_t} = \nu_t$ and $\widehat{\nu}_{t+u_t+1} = \nu_t + 1$ if the t-th phase of the PTVCC has not been punctured. The branch complexity profile of M_{GPCC}, $\widehat{\boldsymbol{b}} = \{\widehat{b}_0, \widehat{b}_1, \ldots, \widehat{b}_{n-1}\}$, can be obtained as follows. For $0 \leq t \leq k-1$, set $\widehat{b}_{t+u_t} = 1$ if the t-th phase of the PTVCC has been punctured, and set $\widehat{b}_{t+u_t} = 1$ and $\widehat{b}_{t+u_t+1} = 0$ if the t-th phase of the PTVCC has not been punctured. The phases in which $\widehat{b}_t = 0$ correspond to the phases with $\widehat{G}_0^t = [0\ x]$ in the scalar generation matrix of the GPCC.

We can summarize the topological restrictions on $\widehat{\boldsymbol{\nu}}$ and $\widehat{\boldsymbol{b}}$ of the GPCC trellis for a (n, k) GPCC as follows:

- $\widehat{\nu}_{t+1} \leq \widehat{\nu}_t + \widehat{b}_t$, for $t = 0, 1, 2, \ldots, n-2$, and $\widehat{\nu}_0 \leq \widehat{\nu}_{n-1} + \widehat{b}_{n-1}$.
- $\widehat{b}_t = 0$ for all $t \in J$, and $\widehat{b}_t = 1$ for all $t \in I \backslash J$, where J is some subset of size $n-k$ of the set $I = \{0, 1, \ldots, n-1\}$;

The trellis complexity of M_{GPCC} for the (n, k) GPCC is given by:

$$TC(M_{GPCC}) = \frac{1}{k} \sum_{t=0}^{n-1} 2^{\widehat{\nu}_t + \widehat{b}_t} \tag{2}$$

symbols per bit.

We now show that the trellis module M_{GPCC} for any (n, k) PCC with memory size ν is the minimal trellis module. We should note that the scalar generator matrix of any PCC in [4] looks like:

$$\begin{bmatrix}
\ddots & [\overline{1}\ x] & & & & & \\
 & [*\ x] & [*\ \overline{1}] & & & & \\
 & \vdots & [*\ *] & & & & \\
 & [\underline{1}\ x] & \vdots & \ddots & [\overline{1}\ x] & & \\
 & & [\underline{1}\ *] & & [*\ x] & [*\ \overline{1}] & \\
 & & & & \vdots & [*\ *] & \\
 & & & & [\underline{1}\ x] & \vdots & \ddots \\
 & & & & & [\underline{1}\ *] &
\end{bmatrix} \tag{3}$$

where every underlined entry (the leftmost nonzero entry in its row) and every overlined entry (the rightmost nonzero entry in its row) occupy positions in a way that satisfy the so called "LR" property [1]. Following the procedure developed in [1] for finding the minimal trellis for convolutional codes, one can find that the state complexities of $\widetilde{M}$ are $\ldots, \nu, \nu, \nu+1, \nu, \ldots, \nu, \nu, \nu+1, \nu, \ldots$. On the other hand, from the construction of the GPCC trellis module described above, we can see that the state complexities of M_{GPCC} and $\widetilde{M}$ coincide, i.e., $\widehat{\nu}_t = \widetilde{\nu}_t$ for all t. According to a property of the minimal trellis for block codes (see for instance [7] for details), which can be adapted to the case of convolutional code, the equality above for all t implies that the two trellis modules are isomorphic. So, for the PCCs in [4], M_{GPCC} is isomorphic to $\widetilde{M}$.

III. Code Search Results

As already mentioned, our goal is to find GPCCs with better distance spectrum than PCCs, with the same code rate and the same trellis complexity. We now describe the procedure we followed to find good GPCCs. The first step was to calculate the value of $TC(\widetilde{M})$ for the existing PCCs. We then proposed templates for the scalar generator matrix of GPCCs. By placing in this matrix the leading and trailing "ones" of each row (as illustrated in the matrix (3)) in specific positions, while others were set free to assume any binary value, we could define ensembles of GPCCs with a particular trellis complexity. By choosing this complexity to be equal to the same $TC(\widetilde{M})$ for the existing PCCs, we searched within the

TABLE I
SOME GOOD GENERALIZED PUNCTURED CONVOLUTIONAL CODES

R	ν	$G(D)$	$TC(\widetilde{M})$	d_f	Spectrum
3/4	3^c	[3 1 1 0; 0 3 1 1; 2 2 3 3]	21.33	4	29,0,532,0,10059
3/4	4^m	[1 1 2 3; 2 1 1 1; 4 6 4 1]	42.67	4	3,44,160,638,3558
3/5	3^b	[3 1 0 0 1; 0 3 3 1 0; 2 0 0 3 3]	26.67	4	1,5,14,40,102
3/5	4^n	[3 3 2 2 3; 2 3 3 1 0; 6 0 2 3 1]	53.33	6	15,0,136,0,1208
4/5	3^m	[1 1 1 1 0; 2 2 1 1 1; 0 2 2 1 1; 2 2 0 2 3]	20.00	3	5,36,200,1065,5893
4/5	4^b	[3 1 1 1 0; 2 3 0 1 1; 2 2 3 1 0; 0 0 2 3 2]	40.00	4	30,126,815,4822,29046
4/7	3^n	[1 1 0 1 1 1 1; 2 1 1 1 0 1 0; 0 2 2 3 3 1 0; 2 2 2 0 0 1 1]	28.00	5	4,13,33,81,203
4/7	4^n	[3 1 0 1 1 0 1; 2 3 3 3 2 0 1; 2 2 2 1 1 1 0; 0 2 0 2 0 3 3]	56.00	6	8,27,46,143,380

^{c} Code also found by Cain et al. [2]
^{m} Code also found by Tang and Lin [8], but with different $G(D)$
^{b} Code also found by Bocharova and Kudryashov [4]
^{n} New code found in this study by a random search

corresponding ensemble and found GPCCs with better distance spectrum.

The free Hamming distance of the convolutional code will be denoted by d_f, and the distance spectrum is defined as a sequence of numbers indicating multiplicities; the first number in this sequence corresponds to the number of codewords of Hamming weight d_f, the second number corresponds to the number of codewords of Hamming weight d_f+1, and so on. The best code is the one with maximum d_f. In the case that more than one code with the same trellis complexity and the same d_f exists, the best code is the one having the smallest distance spectrum in a lexicographical sense.

As an example, consider the best (5,3) PCC with memory size $\nu = 4$, found by an exhaustive search performed by Bocharova and Kudryashov in [4]. This code has $d_f = 6$ and distance spectrum 18,0,139,0,1210,... Viewing this PCC as a GPCC, the state complexity profile of the GPCC trellis module is $\widehat{\boldsymbol{\nu}} = (4,4,5,4,5)$, and the branch complexity profile $\widehat{\boldsymbol{b}} = (1,1,0,1,0)$. Since this code is a PCC, by the property presented at the end of Section II we have that $\widetilde{\boldsymbol{\nu}} = \widehat{\boldsymbol{\nu}}$ and $\widetilde{\boldsymbol{b}} = \widehat{\boldsymbol{b}}$. So $TC(\widetilde{M}) = 53.33$ symbols per bit. After performing a random code search, we found a (5,3) GPCC with $d_f = 6$ and distance spectrum 15,0,136,0,1208,... The state complexity profile of the GPCC trellis module for this GPCC is $\widehat{\boldsymbol{\nu}} = (4,4,5,5,6)$, and the branch complexity profile $\widehat{\boldsymbol{b}} = (1,1,0,1,0)$. Although the GPCC trellis module for this code is more complex than the GPCC trellis module for the PCC, the minimal trellis modules for the two codes have exactly the same state and branch complexity profiles. Therefore, the two codes have the same trellis complexity: $TC(\widetilde{M}) = 53.33$ symbols per bit, but the GPCC we found has a better distance spectrum.

As a second example, consider the best (4,3) PCC with memory size $\nu = 4$, found by Lee [9] (also listed in [4]). This code has $d_f = 4$ and distance spectrum 5,42,134,662,3643,... Tang and Lin [8] have found a (4,3) convolutional code with the same free Hamming distance and the same trellis complexity, but with a better distance spectrum given by 3,44,160,638,3558,... We show next that this code is in fact a GPCC. First, we write part of the scalar generator matrix of the code found in [8]:

$$
\begin{array}{cccccccccccccccc}
1&1&0&1&0&0&1&1&0&0&0&0&&&&\\
0&1&1&1&1&0&0&0&0&0&0&0&&&&\\
0&0&0&1&0&0&1&1&0&1&1&0&&&&\\
&&&&1&1&0&1&0&0&1&1&0&0&0&0\\
&&&&0&1&1&1&1&0&0&0&0&0&0&0\\
&&&&0&0&0&1&0&0&1&1&0&1&1&0\\
&&&&&&&&1&1&0&1&0&0&1&1\\
&&&&&&&&0&1&1&1&1&0&0&0\\
&&&&&&&&0&0&0&1&0&0&1&1\\
&&&&&&&&&&&&\ddots&&&
\end{array}
$$

At a first look, it may seem from the positions of the leading and trailing "ones" in each column of this matrix that this code is a GPCC whose GPCC trellis module has state complexity profile (2,5,6,5). Recalling the topological restrictions for GPCCs presented in Section II, such a profile is not allowed — 4 states at depth zero and 32 states at depth one; there are not enough branches to reach the 32 states. However, we are allowed to perform row operations on this matrix without changing the code. Let $[\kappa]$ denote the κ-th row of a matrix, where $\kappa = 1,2,3,\ldots$. With the following operation:

$$[3\kappa] \leftarrow [3\kappa] + [3\kappa + 2]$$

we turn the matrix above in the following matrix:

$$
\begin{array}{cccccccccccccccc}
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & & & & \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & & & & \\
 & & & & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
 & & & & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
 & & & & & & & & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
 & & & & & & & & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
 & & & & & & & & & & & & \ddots & & &
\end{array}
$$

We now have a GPCC whose GPCC trellis module has state complexity profile (4,5,6,5) and branch complexity profile (1,1,0,1), satisfying the topological restrictions. Note that this code is not a PCC, and has distance spectrum better than that of the best PCC of the same rate and trellis complexity. Therefore, the class of GPCCs seems to contain interesting convolutional codes.

Some good GPCCs found after a computer search are tabulated in Table I. The generator matrices are shown in octal form with the highest power in D in the most significant bit of the representation (*e.g.* $6 \equiv D + D^2$). Table I shows several $(n-1, n)$ GPCCs with the same distance spectrum of the best $(n-1, n)$ codes with the same code rate and trellis complexity of the $(n-1, n)$ PCCs. For other code rates, the GPCCs have better distance spectrum than the corresponding PCCs with the same trellis complexity.

IV. Conclusions

In this paper, we have considered convolutional codes with low trellis complexity (i.e., low Viterbi decoding complexity) and good distance spectrum (i.e., good error performance). The trellis complexity of the convolutional code was defined as the total number of edge symbols per information bit in the minimal trellis module representing the code. A new class of convolutional codes, called the generalized punctured convolutional codes (GPCCs), was introduced. A code in this class was shown to be represented by a trellis module, the GPCC trellis module, whose topology resembles that of the minimal trellis module. The class of GPCCs was shown to include all standard punctured convolutional codes. The GPCC trellis module for a punctured convolutional code was shown to be isomorphic to the minimal trellis module. We showed, by means of examples, that the class of GPCCs contains many interesting convolutional codes. We searched for GPCCs with better distance spectrum than the standard punctured convolutional codes, with the same code rate and trellis complexity. A table containing good GPCCS has been provided.

References

[1] 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.

[2] J. B. Cain, G. C. Clark, Jr., and J. M. Geist, "Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 97-100, Jan. 1979.

[3] Y. Yasuda, K. Kashiki, and Y. Hirata, "High-rate punctured convolutional codes for soft-decision Viterbi algorithm," *IEEE Trans. Commun.*, vol. COM-32, no. 3, pp. 315-319, Mar. 1984.

[4] 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.

[5] 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.

[6] 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.

[7] 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).

[8] H.-H. Tang and M.-C. Lin, "On $(n, n-1)$ convolutional codes with low trellis complexity," *IEEE Trans. Commun.*, vol. 50, no. 01, pp. 37-47, Jan. 2002.

[9] P. J. Lee, "Constructions of rate $(n-1)/n$ punctured convolutional codes with minimal required SNR criterion," *IEEE Trans. Commun.*, vol. 36, no. 10, pp. 1171-1173, Oct. 1988.