Generalized Punctured Convolutional Codes

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

Abstract—This letter introduces the class of generalized punctured convolutional codes (GPCCs), which is broader than and encompasses the class of the standard punctured convolutional codes (PCCs). 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 PCC is isomorphic to the minimal trellis module. A list containing GPCCs with better distance spectrum than the best known PCCs with same code rate and trellis complexity is 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. A trellis module \( M \) for a rate \( R = k/n \) (i.e., a \((n,k)\)) convolutional code \( C \) consists of \( n' \leq n \) trellis sections (from depth 0 to depth \( n' \)), \( 2^{v_1} \) states at depth \( t \), \( 2^{v_2} \) edges emanating from each state 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 \( M \), \( TC(M) \), as the trellis complexity of the convolutional code \( C \).

In this letter, we search for good (in a distance spectrum sense) \((n,k)\) convolutional codes with fixed \( TC(M) \). It appears that a convolutional code search taking this measure of complexity has only been considered in the literature by Tang and Lin [4]. The convolutional codes they found, all of which of rate \((n,n-1)\), had better weight spectrum or decoding complexity than the PCCs in [2]. 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, 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 \((\hat{M}_{GPCC})\) — that shares all of the topological characteristics of the minimal trellis, except possibly the minimality property. It will be shown that many of the \((n,n-1)\) codes found in [4] are GPCCs. Moreover, it is possible to define a template for the scalar generator matrix \( G_{scalar} \) of the GPCC which yields naturally to the minimal-span form [1], allowing the predetermination of the value of \( TC(M) \) for an ensemble of GPCCs.

II. GPCCS

A \((n,k)\) GPCC is a time-varying convolutional code of period \( n \) defined by the binary generator scalars \( g_{i,t} \), where \( t = 0, 1, \ldots, n-1 \) is a phase index and, for each fixed \( t \), \( i = 0, 1, \ldots, \hat{v}_t \), where \( \hat{v}_t \) is the memory order at phase \( t \), with the following 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 \( I = \{0, 1, \ldots, n-1\} \);
- \( \hat{v}_{t+1} \leq \hat{v}_t + 1 \), for \( t = 0, 1, 2, \ldots, n-2 \), and \( \hat{v}_0 \leq \hat{v}_{n-1} + \hat{v}_{n-1} \), where \( \hat{b}_t = 0 \) if \( t \in J \), and \( \hat{b}_t = 1 \) if \( t \in I \setminus J \).

Arranged in the “matrix module” (a vertical slice in \( G_{scalar} \) corresponding to one trellis module [1, eq. (2.4)]), the generator scalars are seen as (only the non-zero rows are shown):

\[
\begin{bmatrix}
  g_{0,0} & \cdots & g_{0,p-1} \\
  g_{1,0} & \cdots & g_{1,p} & g_{1,p+1} \\
  g_{2,0} & \cdots & g_{2,p} & g_{2,p+1} & \cdots \\
  \vdots & \ddots & \vdots & \vdots & \ddots \\
  g_{n-1,0} & \cdots & g_{n-1,p} & g_{n-1,p+1} & \cdots & g_{n-1,n-1} \\
  g_{0,n-1} & g_{1,n-1} & \cdots & g_{2,n-1} & \cdots & g_{n-1,n-1}
\end{bmatrix}
\]
where we have considered that $p \in J$. The lack of information ($\tilde{b}_p = 0$) at phase $p$ causes a “leap” in (1), and $g_0^{p+1}$ is placed to the right of $g_0^p$ (and not following the diagonal, as usual).

The GPCC trellis module $(M_{\text{GPCC}})$ for a rate $(n,k)$ GPCC has state complexity profile (obtained by examining the columns of (1)) $\tilde{\nu} = \{\tilde{\nu}_0, \tilde{\nu}_1, \ldots, \tilde{\nu}_{n-1}\}$, branch complexity profile $\tilde{b} = \{\tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_{n-1}\}$, and a single bit (i.e., $t_i = 1$) labeling each edge for all $t \in I$. The bit value at depth $t$ is obtained from (1) in the usual way. The restrictions defined previously were imposed to avoid unreachable states in the GPCC trellis module. The trellis complexity of $M_{\text{GPCC}}$ is given by:

$$TC(M_{\text{GPCC}}) = \frac{1}{T} \sum_{t=0}^{n-1} 2^{\tilde{\nu}_t + \tilde{b}_t}$$

symbols per bit.

We now show that the PCCs form a subclass of the GPCCs. Consider for simplicity a $((n,k),\nu)$ PCC with memory $\nu$ where $k > n/2$. It is shown in [6] that this PCC can be obtained by puncturing a rate $1/n$ time invariant convolutional code of memory $\nu$ or, equivalently, a rate $1/2$ periodically time-varying convolutional code (PTVCC) of period $k$ and memory $\nu$. Assuming the latter case, then, for the punctured phases, the generator sub-matrices of the PCC are of the form $G_i = [G_i(1) \; x]$, where $x$ represents a punctured output, and, for the non-punctured phases, $G_i = [G_i(1) \; G_i(2)]$. The generator scalars of the GPCC are obtained as follows. Let $u_t$ denote the number of non-punctured phases in the PTVCC occurring prior to phase $t$. By convention, $u_0 = 0$. Then, for $0 \leq t \leq k - 1$, set $g_{t+u_t} = G_i(1)$ and, if the $t$-th phase is non-punctured, then also set $g_{t+u_t+1} = G_i(2)$. The state and branch complexity profiles of $M_{\text{GPCC}}$ for this PCC are given as follows. For $0 \leq t \leq k - 1$, set $\tilde{\nu}_{t+u_t} = \nu$, $\tilde{b}_{t+u_t} = 1$ and, if the $t$-th phase is non-punctured, then also set $\tilde{b}_{t+u_t+1} = 1$. From the construction of the minimal trellis module [1], we can see that the state complexities of $M_{\text{GPCC}}$ and $M$ coincide, i.e., $\tilde{\nu}_t = \nu_t$ for all $t$. According to a property of the minimal trellis for block codes (see, for instance, [5, Theorem 4.26]), which can be adapted to convolutional codes, the equality above for all $t$ implies that the two trellis modules are isomorphic. Moreover, for $\tilde{\nu}$ and $\tilde{\nu}$ as defined above, the summation in (2) becomes $n \cdot 2^{\nu + 1}$. Therefore, for any PCC, $M_{\text{GPCC}}$ is isomorphic to $\tilde{M}$, and $TC(\tilde{M}) = TC(M_{\text{GPCC}}) = TC(M_{\text{PCC}})$.

III. CODE SEARCH RESULTS

In order to find good GPCCs, we first calculated the value of $TC(\tilde{M})$ for the existing PCCs [2]. We then proposed templates for $G_{\text{scalar}}$ of GPCCs. By placing in this matrix 0’s and 1’s in specific positions, while the others were set free to assume any binary value, we could define ensembles of GPCCs with the same $TC(\tilde{M})$ as for the existing PCCs.

As an example, consider the best (5,3) PCC with $\nu = 4$, found by an exhaustive search in [2]. This code has $d_f = 6$ and distance spectrum $18,0,139,0,1210,\ldots$. The state complexity profile of the GPCC trellis module for this PCC is $\tilde{\nu} = (4,4,5,4,5)$, and the branch complexity profile $\tilde{b} = (1,1,0,1,0)$. Since this code is a PCC, by the property presented at the end of Section II we have that $\tilde{\nu} = \tilde{\nu}$ and $\tilde{b} = \tilde{b}$. So $TC(\tilde{M}) = 53.33$ symbols per bit. To exemplify our code search, consider the following template for the “matrix module” of a (5,3) GPCC with state and branch complexity profiles of the GPCC trellis module $\tilde{\nu} = (4,4,5,4,5)$ and $\tilde{b} = (1,1,0,1,0)$:

$$G_{\text{mod}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

We can perform row operations on (4). Let $[\kappa]$ denote the $\kappa$-th row of a matrix, where $\kappa = 1, 2, \ldots$. Performing $[3\kappa] = [3\kappa] + [3\kappa + 2]$, we turn $G_{\text{mod}}$ in (4) into a GPCC form. We thus have a GPCC whose GPCC trellis module has state complexity profile $(4,4,5,4,5)$ 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.
### Table I

**Some Good Generalized Punctured Convolutional Codes**

<table>
<thead>
<tr>
<th>$R$</th>
<th>$\nu$</th>
<th>$G(D)$</th>
<th>$TC(\tilde{M})$</th>
<th>$d_f$</th>
<th>Spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/3</td>
<td>3&lt;sup&gt;t&lt;/sup&gt;</td>
<td>[1 2 3; 6 1 3]</td>
<td>24.00</td>
<td>4</td>
<td>2,11,34,109,366</td>
</tr>
<tr>
<td>2/3</td>
<td>4&lt;sup&gt;t&lt;/sup&gt;</td>
<td>[1 3 8; 10 12 7]</td>
<td>48.00</td>
<td>5</td>
<td>4,17,54,192,681</td>
</tr>
<tr>
<td>3/4</td>
<td>3&lt;sup&gt;b&lt;/sup&gt;</td>
<td>[1 1 0; 0 3 1; 2 2 3 3]</td>
<td>21.33</td>
<td>4</td>
<td>29,0,532,0,10059</td>
</tr>
<tr>
<td>3/4</td>
<td>4&lt;sup&gt;t&lt;/sup&gt;</td>
<td>[1 1 2 3; 2 1 1 1; 4 6 4 1]</td>
<td>42.67</td>
<td>4</td>
<td>3,44,160,638,3558</td>
</tr>
<tr>
<td>3/5</td>
<td>3&lt;sup&gt;n&lt;/sup&gt;</td>
<td>[3 1 0 1 0; 0 3 3 2; 2 2 2 1 1]</td>
<td>26.67</td>
<td>4</td>
<td>1,5,11,38,112</td>
</tr>
<tr>
<td>3/5</td>
<td>4&lt;sup&gt;n&lt;/sup&gt;</td>
<td>[3 3 2 2 3; 2 3 3 1 0; 6 0 2 3 1]</td>
<td>53.33</td>
<td>6</td>
<td>15,0,136,0,1208</td>
</tr>
<tr>
<td>4/5</td>
<td>3&lt;sup&gt;t&lt;/sup&gt;</td>
<td>[1 1 1 1 0; 2 2 1 1 1; 2 2 0 2 3]</td>
<td>20.00</td>
<td>3</td>
<td>5,36,200,1065,5893</td>
</tr>
<tr>
<td>4/5</td>
<td>4&lt;sup&gt;b&lt;/sup&gt;</td>
<td>[3 1 1 1 0; 2 3 0 1 1; 2 2 3 1 0; 0 0 2 3 2]</td>
<td>40.00</td>
<td>4</td>
<td>30,126,815,4822,29046</td>
</tr>
<tr>
<td>4/7</td>
<td>3&lt;sup&gt;n&lt;/sup&gt;</td>
<td>[1 1 0 1 0 1 1 1; 2 1 1 1 0 1 0; 0 2 2 3 3 1 0; 2 2 2 0 0 1 1]</td>
<td>28.00</td>
<td>5</td>
<td>4,13,33,81,203</td>
</tr>
<tr>
<td>4/7</td>
<td>4&lt;sup&gt;n&lt;/sup&gt;</td>
<td>[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]</td>
<td>56.00</td>
<td>6</td>
<td>8,27,46,143,380</td>
</tr>
<tr>
<td>5/6</td>
<td>3&lt;sup&gt;t&lt;/sup&gt;</td>
<td>[1 1 0 1 0 0; 0 1 1 0 1 0; 2 0 0 1 0 1; 2 2 2 0 1 1; 0 0 0 2 3]</td>
<td>19.20</td>
<td>3</td>
<td>15,96,601,3903,25325</td>
</tr>
<tr>
<td>5/6</td>
<td>4&lt;sup&gt;b&lt;/sup&gt;</td>
<td>[1 0 0 1 1 1; 2 1 1 1 0 0; 0 2 1 1 0 1; 2 2 2 1 1 1; 0 2 0 2 1 1]</td>
<td>38.40</td>
<td>4</td>
<td>111,0,5628,0,293724</td>
</tr>
<tr>
<td>5/7</td>
<td>3&lt;sup&gt;n&lt;/sup&gt;</td>
<td>[1 1 0 0 1 1 1; 0 1 1 1 1 0 0; 2 0 0 1 1 0 0; 2 2 2 0 1 0 0; 0 0 2 2 0 1 1]</td>
<td>22.40</td>
<td>4</td>
<td>15,55,174,798,3214</td>
</tr>
<tr>
<td>5/7</td>
<td>4&lt;sup&gt;b&lt;/sup&gt;</td>
<td>[1 1 0 0 1 1 1; 2 1 1 1 0 1; 2 2 2 1 0 0 1; 0 2 0 2 1 1 0; 2 0 0 0 2 1 1]</td>
<td>44.80</td>
<td>4</td>
<td>2,27,109,445,1955</td>
</tr>
</tbody>
</table>

<sup>t</sup> Code found in [4], but with different $G(D)$; <sup>b</sup> Code listed in [2]; <sup>n</sup> New code found in this study by a random search.

Some good GPCCs are tabulated in Table I for different code rates $R$, and memory order $\nu$ of the corresponding PCC (which has the same $TC(\tilde{M})$ of the listed GPCC). The polynomial generator matrices $G(D)$ 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 listed in [4]. For other code rates, the new GPCCs shown in Table I have better distance spectrum than the corresponding PCCs [2] with the same trellis complexity.

### IV. Conclusions

The trellis complexities considered in this paper were restricted to those of the PCCs in [2]. The codes listed in Table I only improved the distance multiplicity, however a whole bunch of more complex convolutional codes with greater $d_f$ exist. The study of GPCCs with other code rates and various trellis complexities is a research topic currently being investigated by the authors. For example, by placing in (3) the leading and trailing 1’s in specific positions we found a (5,3) GPCC with $d_f = 5$ and distance spectrum $5,13,38,113,303,\ldots$, but with $TC(\tilde{M}) = 37.33$ symbols per bit.

### References


