Geometrically Uniform TCM Codes over Groups Based on Multidimensional 4-PSK and 8-PSK Signals

S. Benedetto  R. Garelo  M. Mondin  G. Montorsi  
Dipartimento di Elettronica, Politecnico di Torino  
C.so Duca degli Abruzzi 24, 10129 Torino, Italy  
Ph. +39-11-5644031 - FAX +39-11-5644099

Abstract

The theory of geometrically uniform (GU) signal constellations and linear codes over groups is applied to the case of multidimensional PSK signals. The tables of GU partitions of multidimensional PSK constellations recently found in [1] are used to construct good GU codes over nonbinary abelian groups. We consider Lx4PSK and Lx8PSK constellations used to transmit information rates of 1 and 2 bits per signaling interval, respectively, and present a systematic constructive procedure to search for good codes over generating groups (Z_4)^L and (Z_8)^L. Rotational invariance conditions for the codes are easily incorporated into the search algorithm. In some cases, the codes found improve over the codes known so far. Moreover, the geometrical uniformity of codes allows a very easy performance evaluation, so that we also present a set of curves of error event probability for some selected codes.

1 GU PARTITIONS OF LxMPSK

A constellation S is Geometrically Uniform (GU) if its symmetry group G(S) of S is a subgroup of G(S), which is minimally sufficient to generate S starting from an arbitrary initial point of it, i.e. which is sharply transitive. A constellation S=MPSK is GU; denoting by R_M the cyclic group of rotations by multiples of 360/M degrees and by V the group consisting of the identity and the reflection about the line between any point or midpoint of S and the origin, the symmetry group of S is the semidirect product R_M • V, isomorphic to the nonabelian dihedral group D_M. Possible generating groups of S=MPSK, for M even, are the abelian group R_M isomorphic to Z_M and the nonabelian group R_M/2 • V, isomorphic to D_M/2. GU constellations have the important property that the Voronoi regions are congruent, so that the error probability is independent of which signal was transmitted. In [2] this property was shown to hold for signal sequences, through a suitable extension of the concept of geometrical uniformity. A normal subgroup G' of the generating group G(S) induces a GU partition S/S' of the constellation S, in which each subset of the partition is GU and has G' as a common generating group. A one-to-one mapping is induced between the quotient group G/G' and the subsets of the partition S/S'.

Given a GU partition, we can choose a label group A for the subsets of signals, isomorphic to the quotient group G/G'. By combining the label isomorphism with the one-to-one mapping G/G' → S/S', one obtains a one-to-one mapping m : A → S/S' called an isometric labeling. Partitions S/S' that admit isometric labelings by n-bit binary label alphabets A = (Z_2)^n, i.e. binary isometric labelings, are particularly important for applications. As an example, there exist binary isometric labelings for four-way partitions of the 2^2PSK constellation, but binary isometric labelings do not exist for partitions into more than four subsets [2]. It is now possible to extend the label group to a label sequence space A^n, by taking the n-fold Cartesian product of A with itself, and possibly letting n go to infinity. A label code C over the label group A is any subgroup of A^n. As an example, a linear rate k/n binary convolutional code may be used as a label code if the label group is A = (Z_2)^n. A generalized coset code, or GU code, is defined by a GU partition S/S', an isometric labeling m : A → S/S' and a label code. The basis for a GU TCM code with good properties in terms of minimum Euclidean distance among code sequences is a GU partition with a minimum squared Euclidean distance within constellations at a given partition level as large as possible. Given the 2D S=MPSK constellation, an LxMPSK constellation is obtained from L Cartesian products of S by itself. In [1] we presented efficient algorithms to find generating groups of such constellations and good GU partitions of LxMPSK, with L≤2,3,4 and M=4,8,16. These partitions have been used in [1] to find good GU TCM codes having the constraint of binary isometric labeling, i.e. with label group A isomorphic to (Z_2)^n. Although in some cases the codes found improve over previously known codes, the binary isomorphism constraint poses for some data rates serious limitations to the achievable coding gain. On the other hand, the tables of GU partitions is not limited to partitions admitting binary isomorphism. In particular, they can be used to construct TCM schemes over nonbinary groups. The concept of linear, or convolutional, codes over groups was recently introduced independently by [3],[4]. In [3] and [4] two structures for minimal, canonical encoders for codes over groups are introduced. In this paper we will use the structure introduced in [3], as it seems more directly suited.
to the tables of GU partitions derived in [1]. So far, to the authors knowledge and to the words in [4] “no interesting new codes are presented...It is hoped that future, more concrete work can be based on these foundations”, the remarkable theory of linear codes over groups has received no confirm of its “practical” usefulness through the presentation of new codes which improve, in some sense, over binary-based codes. In this paper, we present some results which aim at answering the following questions:

- In the field of GU TCM codes, can one expect any improvement when passing from codes isomorphic to binary labels to more general group codes?
- In general, apart from the important simplifications in the code search and performance evaluations, can GU codes over groups improve over the best non GU TCM codes based on binary convolutional codes known so far?

We will not present general and comprehensive results. However, for multidimensional 4 and 8PSK constellations, our results are of some interest. We prove that the answers to both questions are indeed positive, by presenting new GU codes based on nonbinary abelian groups which seem to incentivate the research in the field, and, in themselves, are a substantial improvement over known codes. One of the aspects which is given particular consideration is the rotational invariance, for which we present necessary and sufficient conditions which are easily incorporated within the algorithm searching for new codes. A linear code over the group G is a set of sequences e of elements c of drawn from the group G on a discrete index set I which form a group under the componentwise group operation. The general structure of the minimal canonical feedback-free encoder of a linear group code, drawn from [3], is represented in Fig. 1. The meaning of the symbols is as follows: A represents the output group, defined as the set of symbols c that occur in some code sequence e. In general, A is a subgroup of G and when A=G the code is called trim. F represents the input group, i.e. the set of symbols associated to the branches of the trellis leaving the zero state. The input group F must be a subgroup of the output group A. The labels of the output of the block “COSET DECOMPOSITION” are groups called granules: there exists a chain coset decomposition of F such that there is a one-to-one correspondence $F = \Gamma_0 \times \Gamma_1 \times \ldots \times \Gamma_{\nu}$, so that we have $|F| = |\Gamma_0| |\Gamma_1| \ldots |\Gamma_{\nu}|$. The granules are tied to the sequences of the code in this way: if $C_{[0,j]}$ is the subgroup formed by all the sequences which are zero outside the interval [0, i+1], then $\Gamma_0 = C_{[0,0]}$ and, for $j > 0$, $\Gamma_j = C_{[0,j]} / C_{[0,j-1]} C_{[1,j]}$, i.e. the quotient group of $C_{[0,j]}$ with the product of $C_{[0,j-1]}$ and $C_{[1,j]}$. Every element of the input is decomposed according to the input chain in $\nu$ elements $\gamma_j \in \Gamma_j$, and the contribution of $\gamma_j$ to the output is $\gamma_j$, a coset representative of $\gamma_j$ belonging to a set of coset representatives $|\Gamma_j|$ of $\Gamma_j$. All these contributions are summed in the block “COMBINATION”. We derive now a more practical encoder structure, which makes explicit use of the coset identifiers $c_j \in |\Gamma_j|$, which will be called generators of the code. Since we only use abelian groups $G \cong (Z_M)^J$, with $M$ a power of two, it is always possible, using the normal base decomposition of the granules, to find a binary chain coset decomposition for $\Gamma_j$ and further decompose every $\Gamma_j$ through a one-to-one correspondence:

$$\Gamma_j \rightarrow \Gamma_j^1 \times \ldots \times \Gamma_j^\nu \times \ldots \times \Gamma_j^{N_j} \quad \text{0} \leq j \leq \nu,$$

where $|\Gamma_j| = 2^{N_j}$ and for every k we have $\Gamma_j^k \cong Z_2$. Then $|\Gamma_j^k|$ has the form $|\Gamma_j^k| = \{0, c_j^k\}$. As a consequence, we can further decompose every $\gamma_j$ as $\gamma_j = \gamma_j^1 \ldots \gamma_j^\nu \gamma_j^{N_j}$ and the correspondence between $\gamma_j^k$ and the coset representative $c_j^k$ can be implemented through a multiplication of $c_j^k$ by the corresponding value of a binary input line. Turning now to the possible applications, we introduce into the picture the information rate of the code $R$, defined as the number of information bits associated to each output symbol of the code. In the following, we will use also the information rate per signaling interval $R_{eff}$ defined in [5], which is useful for MD constellations like LxMPSK; it is related to $R$ by $R_{eff} = R/L$. It is evident that the order $|F|$ of the input group will be $|F| = 2^R$. Moreover, the number of states $N_s$ of the codes described by the minimal encoder of Fig.1 is:

$$N_s = |\Gamma_1|^1 \cdot |\Gamma_2|^2 \ldots \cdot |\Gamma_\nu|^\nu.$$  \hspace{1cm} (1)

For a given number of states, there are, as a consequence of (1), a limited number of configurations. In the tables of codes presented later, each code will be identified by a list of generators $c_j^k$; a generator is a sequence of $(j + 1)$ symbols belonging to $G$, separated by a comma; generators with different $k$ and/or $j$ are separated by a #. Every generator corresponds to a binary input line, appearing in the decomposition of $F$, with $j$ delay elements. At each signaling interval, the generator symbols whose binary input value is 1 are added together, and the result is the output element at that instant.

Example 1

As an example, with $S = 2 \times 4$PSK and $G(S) = (Z_4)^2$, for $N_s = 4$, the code identified by the generators (22, 02) # (11, 21) has the encoder structure reported in Fig.2.

2 CODES CONSTRUCTION

Given an encoder configuration, a linear code over a group $G$ can be obtained by building subgroups $C_{[0,j]}$ sequences of length $(j + 1)$, with $0 \leq j \leq \nu$, and then computing from them the quotient groups $\Gamma_j^k$ and the corresponding generators $c_j^k \in |\Gamma_j^k|$. This method allows an important simplification in the search. In fact, if we want to obtain a code with free Euclidean distance larger than $d^2$, we can discard, when passing through the subgroups $C_{[0,j]}$, all those subgroups generating subcodes with free distance lower than $d^2$. To do this, a crucial role is played by the tables of partition in [1], as they suggest the choice of the paralel transition subgroup, which is the starting step of the search algorithm. The algorithm we used to find good codes over groups requires the choice of:

- the constellation $S = L \times M$PSK with generating group $(Z_M)^J$. 

1410
• the rate \( R_{\text{eff}} \),
• the number of states \( N_o \),
• the minimum value \( d^2 \) of the code Euclidean free distance which we want to obtain.

Moreover, the encoder configuration and the parallel transition subgroup (if any) must be chosen, as follows. Given \( R_{\text{eff}} \), and consequently \( R = R_{\text{eff}} \cdot L \), the order \( |F| = 2^p \) of the input group is determined. For a given number of states, only a limited number of admissible encoder configurations from (1) can be used. Given \( S \), we can use the tables of [1], or the algorithms there described, to choose a parallel transition subgroup of \(|\Gamma_0|\) signals which guarantees an Euclidean distance larger than (or equal to) \( d^2 \). The chosen parallel transition subgroup \( C_{[0,0]} = \Gamma_0 \) is stored in the array \( D \). From the chosen encoder configuration we get the list of the values \( |\Gamma_j| \), with \( 1 \leq j \leq \nu \). After this initialization phase, we can apply the search algorithm, which is composed of the following steps:

**Description of the algorithm**
1. Set \( p = 1 \).
2. while \( |\Gamma_p| = 1 \) set \( p = p + 1 \).
3. Set \( \nu_p = \log_2 |\Gamma_p| \).
4. for every code belonging to \( D \) build the corresponding group \( C_{[0,0]} \cdot C_{[0,0]} \) and store it in the array \( B \).
5. Set \( D = B \).
6. Set \( q = 1 \).
7. Take a subgroup \( C \) belonging to \( D \); let \( N \) be its order.
8. Using the Procedure 1 (described later) build all the acceptable subgroups of \((G)^{p+1}\) of order \( 2N \) containing \( C \) and store them in \( B \).
9. if all the subgroups belonging to \( D \) have not been considered go to 7.
10. Set \( D = B \) and \( q = q + 1 \).
11. if \( q \leq N_o \) go to 7, else set \( p = p + 1 \).
12. if \( p \leq \nu \) go to 2, else the search algorithm is concluded.

For every subgroup in \( B \) we have a list of \( R \) generators: \( \log_2 |\Gamma_0| \) generators of the parallel transition subgroup and all the elements \( a \) which have been used in Step 3 to pass from a group of order \( N \) to a group of order \( 2N \).

For every code in \( B \) compute the free squared Euclidean distance \( d^2 \) and its multiplicity \( N_f \).

(Note that the computation of \( d^2 \) is simplified by the geometrically uniformity of the code and only the distance from the all-zeroes sequence have to be computed.)

**Procedure I**
Given a subgroup \( C \) of order \( N \) formed by sequences of length \( p + 1 \) we construct all the subgroups of \((G)^{p+1}\) of order \( 2N \) containing \( C \) as follows: select, among all the \((p+1)\)-sequences \( a \in (G)^{p+1} \), those satisfying the following conditions:
• \( a \not\in C \) and \( 2a \in C \);
• the first symbol of \( a \) is different from the first symbol of all the elements of \( C \) (in particular is different from \( 0 \)) and the last symbol of \( a \) is different from the last symbol of all the elements of \( C \) (in particular is different from \( 0 \));
• given the coset \( C' = a + C \), all the sequences belonging to \( C' \) (in particular \( a \)) have a distance from the zero sequence greater or equal to \( d^2 \).

To reduce the search complexity, we have incorporated in the algorithm the following shortcuts. In procedure 1, it is not allowed to assume all the values in \((G)^{p+1}\); it is simple, starting from the sequences belonging to the subgroup \( C \) under exam, to generate directly only the sequences \( a \) such that \( 2a \notin C \). To avoid the choice of nonminimal generators, which can be obtained as the sum of shortest generators and lead to catastrophic encoders, a necessary condition is that for every sequence belonging to \( C' \) (in particular for \( a \)), the sum of all the symbols of the sequence is different from \( 0 \) (see Section 3). In this way most of non-minimal generators are discarded; those which still pass the test yield codes with an effective number of states less than \( N_o \) and, being suboptimum, are discarded at step 13 of the algorithm.

**Example 2**
We want to build a code with \( S = 2 \times 4\)PSK, \( G(S) = (Z_4)^2 \), \( R_{\text{eff}} = 1\)bit/T and number of states \( N_o = 2 \), with \( d^2 = 8 \). For \( N_o = 2 \) there is only one admissible encoder configuration from (1): \( |\Gamma| = 2 \) and \( |\Gamma_0| = 1 \), \( V > 1 \). Since \( R_{\text{eff}} = 1 \) bit/T we have \( R = 2 \) and \( |F| = 4 \). As a consequence we must have \( |\Gamma_1| = 2 \), i.e. a pair of parallel transitions. From the tables of [1] we can choose as parallel transition subgroup the group: \( G = \{00, 22\} \), which satisfies the distance constraint. We put \( G \) into \( D \) and set \( p = 1 \); then we have \( N_o = 1 \). Given \( G \) we build the group \( C_{[0,0]} \cdot C_{[1,0]} \cdot C_{[1,1]} \cdot C_{[0,1]} \cdot C_{[0,2]} \cdot C_{[2,0]} \cdot C_{[2,2]} \). The resulting group \( G(S^L) \) is composed of \( 2^L \) elements.

3 ROTATIONALLY INVARIANT CODES

Let us first recall some definitions and properties about the rotational invariance of a code (see [1] and the references therein). Let \( r_k \) denote the rotation by \( k \) elements. Given a 2D constellation \( S = \text{MPSK} \), we introduce the pure rotation group \( R/(\text{MPSK}) = \{1, r_1, r_2, \ldots, r_{M-1}\} = \langle r_k \rangle \geq Z_M \). For a multidimensional constellation \( S^L \) obtained by \( L \) Cartesian products of \( S \) by itself, we can introduce the Rotationally Invariant subGroup \( RIG(S^L) = \langle r_k \rangle \geq Z_M \). For a multidimensional constellation \( S^L \) obtained by \( L \) Cartesian products of \( S \) by itself, we can introduce the Rotationally Invariant subGroup \( RIG(S^L) = \langle r_k \rangle \geq Z_M \). For a multidimensional constellation \( S^L \) obtained by \( L \) Cartesian products of \( S \) by itself, we can introduce the Rotationally Invariant subGroup \( RIG(S^L) = \langle r_k \rangle \geq Z_M \).
symbols can only be associated to state sequences replicating the same state, i.e. states presenting a self-transition in the trellis. As a consequence, a necessary and sufficient condition for a group code \( C \) over \( \mathbb{Z}_L^k \) to be invariant with respect to \( R \) is that there exists in the trellis a self-transition labeled by the symbol \( k, \ldots, k \). We will show which are the implications of this on the choice of the code generators. A generator \( g^j \) consists of a sequence of \((j+1)\) symbols of \( G \). Let us define as \( s^j \) the sum in \( G \) of these \((j+1)\) symbols. Owing to the encoder structure, a self-transition occurs when the contents of the delay elements for every binary input line are all ones or all zeros. The symbols associated with a self-transition are all the possible binary combinations of \( s^j \), \( \forall j, \forall k \), which form a group denoted by \( \Sigma \). A necessary and sufficient condition for a group code \( C \) over \( \mathbb{Z}_L^k \) to be invariant with respect to \( R \) is that the symbol \( k, \ldots, k \) belongs to \( \Sigma \). As \( RIG(s^j) \) is a cyclic group, the code \( C \) will be invariant with respect to it if the symbol \( 1, \ldots, 1 \) belongs to \( \Sigma \). The necessary and sufficient conditions for the rotational invariance of a code have also been incorporated within the algorithm as an option; this allows us to find either the best code in terms of \( d^j_2 \) or the best rotationally invariant code.

4 CODES OVER \((Z_4)^L\) USING L×4PSK

Linear TCM codes over \((Z_4)^L\) using \( L \times 4\)PSK constellations for \( L = 2, 3, 4 \) and \( R_{eff} = 1 \) have been obtained using the previously described algorithm and the tables of GU partitions derived from [1]. For \( L = 1 \), minimal encoders with more than one state cannot be constructed. In fact, in this case \( R_{eff} = 1 \) bit/T, so that \( |F| = 2 \) and, being \( G = Z_4 \), we have \(|A| \leq 2 \). The best codes found for \( L = 2 \) are reported in Table 1. For each number of states, when the best code in terms of \( d^2_2 \) is not rotationally invariant, we also report the best rotationally invariant code. In the tables, we list the parameters \( d^2_2, N_f \). The last column gives the asymptotic gain in dB with respect to the uncoded 2-PSK. Remarkable codes are the rotationally invariant 16-state code with \( d^2_2 = 16 \), to be compared with the 180 degrees invariant code of [5] having \( d^2_2 = 12 \), and the rotationally invariant codes with 32, 64, 128 and 256 states (not existing in [5]). The performance of the codes in terms of upper bounds to the error event probability are shown in Fig. 5, together with the curve referring to the uncoded 2-PSK. The curves are obtained through the standard transfer function technique, owing to the geometrical uniformity of the codes. The bounds are quite tight below \( 10^{-4} \). In the figure, for each number of states, we report the best codes in terms of \( d^2_2 \) and \( N_f \), together with (dashed curves), the best rotationally invariant codes for 4, 16, 32 and 64 states. These curves permit to evaluate the penalty in coding gain which is paid to obtain the rotational invariance.

5 CODES OVER \((Z_8)^L\) USING L×8PSK

Linear TCM codes over \((Z_8)^L\) using \( L \times 8\)PSK constellations for \( L = 2, 3, 4 \) and \( R_{eff} = 2 \) have been obtained using the previously described algorithm and the tables of GU partitions derived from [1]. Also in this case, for \( L = 1 \), we have \(|F| = 4 \) and, being \( G = Z_8 \), \(|A| \leq 4 \). Thus, no minimal encoders with more than one state can be constructed. The best codes found for \( L = 3 \) are reported in Table 2. We remember here that in [1], we proved that GU TCM schemes employing MD 8PSK constellations and based on binary convolutional codes had an upper bound of 4 in the achievable \( d^2_2 \); this upper bound is no more valid here, as it can be seen from the table. The symbols in the table are evident from the previous explanation. The asymptotic gain is evaluated with respect to uncoded 4-PSK. The table shows significant improvement with respect to the codes previously found. Remarkable examples are the rotationally invariant 8-state code, and, in particular, the two 16-state codes. The first is 90 degrees invariant with a \( d^2_2 = 5.757 \), to be compared with 4 in [5]; the second is rotationally invariant, with \( d^2_2 = 5.172 \). Also in this case, we present a complete set of curves of error event probability in Fig. 6.

Acknowledgments

We are grateful to G. David Forney, Jr. and M. D. Trott for providing a preprint of their paper [3], which has been used extensively throughout this work. We also had several helpful discussions with them, and with H. Loeliger and T. Mittelholzer.

References

Table 1: $S=2\times4$PSK, $R_{eff}=1$ bit/T

<table>
<thead>
<tr>
<th>$N_g$</th>
<th>generators</th>
<th>$d_f^2$</th>
<th>$N_f$</th>
<th>$\gamma$(dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>02 # 03,21</td>
<td>90</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>22.02 # 11,21</td>
<td>360</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>20.02 # 12,21</td>
<td>90</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>22.02 # 02,11,21</td>
<td>360</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>20,22,20 # 12,13,12</td>
<td>90</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>32</td>
<td>02,22,22 # 21,11,11,04</td>
<td>360</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>52</td>
<td>22,02,02 # 02,13,01,21</td>
<td>90</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>64</td>
<td>22,02,22,22 # 11,23,33,13</td>
<td>360</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>64</td>
<td>22,22,20,02 # 15,11,32,21</td>
<td>90</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>128</td>
<td>22,20,22,02 # 02,11,10,11,21</td>
<td>90</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>256</td>
<td>22,02,20,20,02 # 13,23,30,12,21</td>
<td>90</td>
<td>24</td>
<td>25</td>
</tr>
</tbody>
</table>

Figure 4: Two-state encoder of example 2

Table 2: $S=3\times8$PSK, $R_{eff}=2$ bit/T

<table>
<thead>
<tr>
<th>$N_g$</th>
<th>generators</th>
<th>$d_f^2$</th>
<th>$N_f$</th>
<th>$\gamma$(dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>044 # 422 # 215 # 222 # 004 # 013,002</td>
<td>90</td>
<td>3.172</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>004 # 042 # 402 # 222 # 130 # 021,002</td>
<td>45</td>
<td>2.929</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>004 # 042 # 402 # 222 # 023,002 # 111,021</td>
<td>45</td>
<td>4.000</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>444 # 266 # 173 # 040,004 # 084,042 # 036,061</td>
<td>45</td>
<td>4.586</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>444 # 222 # 040,004 # 155,024,042 # 127,061</td>
<td>90</td>
<td>5.757</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>444 # 266 # 040,004 # 137,040 # 153,002 # 105,021</td>
<td>45</td>
<td>5.172</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td>444 # 266 # 040,004 # 177,173 # 113,042 # 145,031,061</td>
<td>45</td>
<td>6.000</td>
<td>2</td>
</tr>
<tr>
<td>64</td>
<td>044 # 422,004 # 004,022 # 215,402 # 002,411 # 085,313,201</td>
<td>45</td>
<td>6.101</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 6: Upper bounds to the error event probability for $3\times8$PSK TCM codes (Table 2)