Constructions of Generalized Concatenated Codes and Their Trellis-Based Decoding Complexity

Robert H. Morelos-Zaragoza  
*LSI LOGIC Corp.*, robert.morelos-zaragoza@sjsu.edu

Toru Fujiwara  
*Osaka University*

Tadao Kasami  
*Nara Institute of Science and Technology*

Shu Lin  
*University of Hawaii at Manoa*

Follow this and additional works at: [http://scholarworks.sjsu.edu/ee_pub](http://scholarworks.sjsu.edu/ee_pub)  
Part of the [Electrical and Computer Engineering Commons](http://scholarworks.sjsu.edu/ee_pub)

Recommended Citation  
Constructions of Generalized Concatenated Codes and Their Trellis-Based Decoding Complexity

Robert Morelos-Zaragoza, Senior Member, IEEE,
Toru Fujiwara, Member, IEEE, Tadao Kasami, Life Fellow, IEEE,
and Shu Lin, Fellow, IEEE

Abstract—In this correspondence, constructions of generalized concatenated (GC) codes with good rates and distances are presented. Some of the proposed GC codes have simpler trellis complexity than Euclidean geometry (EG), Reed–Muller (RM), or Bose–Chaudhuri–Hocquenghem (BCH) codes of approximately the same rates and minimum distances, and in addition can be decoded with trellis-based multistage decoding up to their minimum distances. Several codes of the same length, dimension, and minimum distance as the best linear codes known are constructed.

Index Terms—Generalized concatenated codes, multistage decoding, trellis complexity.

I. INTRODUCTION

The trellis structure of linear block codes was first introduced in [1] and later studied in [2]. In [2] it is shown that every binary linear \((n, k)\) code has an \(n\)-section trellis diagram with at most \(2^k\) states. Later on, the trellis structure of Reed–Muller (RM) codes was analyzed in [3], where a minimal trellis construction for linear block codes was presented. Since then, there has been a considerable amount of research effort devoted to the study and applications of the trellis structure of linear block codes.

A trellis diagram (or a trellis) for a linear block code with the minimum number of states is said to be minimal. A minimal trellis is unique up to graph isomorphism [3]–[5]. It has been shown [3]–[6] that the state complexity of a minimal trellis for a linear block code depends on the order of its code symbol positions. However, symbol ordering does not affect the trellis state complexity of maximum-distance-separable (MDS) codes. (This result will be particularly useful in this correspondence, as many of the outer codes used in the proposed concatenated constructions are MDS codes.)

Generalized concatenated (GC) codes were introduced by Zinoviev [7] and by Blokh and Zyablov [8] in 1976, and form a powerful family of error-correcting codes that can correct both random errors and random bursts of errors. In addition, GC codes are a class of multilevel codes that are amenable to multistage decoding, which provides a good tradeoff between error performance and decoding complexity.

Manuscript received April 11, 1996; revised September 22, 1996. This work was supported in part by NSF under Grant NCR-9415374, NASA under Grant NAG 5-931 and the Ministry of Education, Japan, under Grant (C) 06650416. The material in this correspondence was presented in part at the 1993 Symposium on Information Theory and its Applications, Kanazawa, Japan, October 1993, and at the 1994 IEEE International Symposium on Information Theory, Trondheim, Norway, June 1994. R. Morelos-Zaragoza is with LSI LOGIC Corp., Milpitas, CA 95035 USA. T. Fujiwara is with the Faculty of Engineering Science, Osaka University, Toyonaka, Osaka 560-8531 Japan. T. Kasami was with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara 630-0101 Japan. He is now with the Faculty of Information Science, Hiroshima City University, Hiroshima 731-3194 USA. S. Lin is with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA. Communicated by A. Vardy, Associate Editor for Coding Theory.

Publisher Item Identifier S 0018-9448/99$10.00 © 1999 IEEE
In this correspondence, good GC codes are constructed. These codes are good in the sense that they have lower trellis-based decoding complexities compared with permuted Bose-Chaudhuri-Hocquenghem (BCH), RM, and Euclidean geometry (EG) codes of the same lengths, the same (or approximately the same) rates and minimum distances, and that they can be decoded with trellis-based multistage decoding up to their minimum distances. The decoding complexity of a GC code $C$ is measured both by the maximum number of states in an $n$-section trellis for $C$ and by the number of addition-equivalent operations required in a Viterbi decoder using a minimal trellis for $C$.

II. MINIMAL TRELLISES AND STATE COMPLEXITIES OF DECOMPOSABLE CODES

In this section, the connection between symbol orderings of linear codes and reduced upper bounds on the state complexity of their trellis diagrams is pointed out. A sufficient condition on the minimality of the product of trellises is also presented. In a later section, it is shown that if for each component code of a decomposable code there is an optimal ordering, then the ordering is also optimal for the overall code.

Throughout the correspondence, $(n, k, d)$ is used to denote the parameters of a linear block code of length $n$, dimension $k$, and minimum distance $d$. Let $C$ be an $(n, k, d)$ linear code over GF$(q)$. Suppose that $C$ is a decomposable code, defined, in terms of its linear $(n, k_i, d_i)$ subcodes $C_i$ with $1 \leq i \leq M$, by the following conditions:

(S) $C \leftarrow C_1 + C_2 + \cdots + C_M \triangleq \{ u_1 + u_2 + \cdots + u_M : u_i \in C_i \text{ with } 1 \leq i \leq M \}$.

(D) For $u_i \in C_i$, with $1 \leq i \leq M$, $u_1 + u_2 + \cdots + u_M = 0$ (the zero codeword) if and only if $u_1 = u_2 = \cdots = u_M = 0$.

Then it follows from (S) and (D) that

$$k = k_1 + k_2 + \cdots + k_M. \quad (1)$$

Let $T$ and $T_i$ with $1 \leq i \leq M$ denote the minimal trellis diagrams for $C$ and $C_i$, respectively. As defined in [10] and [11], the direct product of $M$ trellis diagrams, $T_1 \otimes T_2 \otimes \cdots \otimes T_M$, denoted $T_1 \otimes T_2 \otimes \cdots \otimes T_M$, is a trellis diagram defined as follows: The states in the product are $M$-tuples $(s_1, s_2, \ldots, s_M)$, where $s_i$ is a state of trellis $T_i$, $1 \leq i \leq M$, and there exists a branch of label $\ell = \ell_1 + \ell_2 + \cdots + \ell_M$, with initial state $(s_1, s_2, \ldots, s_M)$ and final state $(s_1', s_2', \ldots, s_M')$ if, and only if, there is a branch of label $\ell_i$ from state $s_i$ to state $s_i'$ in trellis $T_i$, $i = 1, 2, \ldots, M$.

Given a code symbol ordering for a code $C$, there is a unique trellis diagram (or trellis) with the minimum number of states for $C$, called the minimal trellis diagram (or minimal trellis) for $C$. For $1 \leq j \leq n$ and $1 \leq i \leq M$, let $s_j(T)$ and $s_j(T_i)$ denote the logarithms base $q$ of the numbers of states of $T$ and $T_i$ just after the $j$th code symbol. These numbers are known as the state complexities of $T$ and $T_i$. Since the product of $T_i$ with $1 \leq i \leq M$ is a trellis for $C$, we have that

$$s_j(T) = \sum_{i=1}^{M} s_j(T_i), \quad 1 \leq j \leq n. \quad (2)$$

For a linear code $C$ of length $n$, let $s_{\text{max}}(C)$ denote the maximum number of states at any bit position of a minimal $n$-section trellis $T$ for $C$. The quantity $s_{\text{max}}(T)$ will be referred to as the state complexity of code $C$. It follows from (2) and Wolf’s bound [2] that

$$s_{\text{max}}(T) \triangleq \max_{0 \leq j \leq n} s_j(T) \leq \sum_{i=1}^{M} \min(k_i, n-k_i). \quad (3)$$

For several classes of codes such as RM codes or their subcodes and repetition codes, and their dual codes, there are known code symbol orderings [6], [9] which result in reduced upper bounds on the state complexity of their trellis diagrams compared with Wolf’s bound. If there is such a code among $C_i$, with $1 \leq i \leq M$, then we can adopt the corresponding symbol ordering and evaluate the state complexity of trellis diagram for $C$ by applying upper bounds which are independent of any symbol ordering of each remaining component code.

Suppose that

$$C_i^{-} \leftarrow C_i^L - C_i^P = C_i^{P+} \leftarrow C_i^P + C_i^R = C_i^{-} \leftarrow C_i^{P-} \leftarrow C_i^P + C_i^R, \quad (4)$$

and

$$C_i^{\pm} \leftarrow C_i^{-} + \cdots + C_i^{\pm} \leftarrow C_i^{-} + \cdots + C_i^{\pm}, \quad (5)$$

where

$$C_i^{\pm} \triangleq \{ u \in C : u_{\ell} = 0, 1 \leq \ell \leq j \}$$

are the past and future subcodes of $C_i, 1 \leq i \leq M$, respectively. Then it follows from (4), (5), and property (D) that

$$k(C_i^{-}) = k(C_i^{P-}) + \cdots + k(C_i^{P+}), \quad (6)$$

$$k(C_i^R) = k(C_i^{P+}) + \cdots + k(C_i^{P-}). \quad (7)$$

Equations (6) and (7) imply that, under the assumptions (4) and (5)

$$s_j(T) = \sum_{i=1}^{M} s_j(T_i), \quad (8)$$

for any $j$ with $1 \leq j \leq n$.

A sufficient condition for (4) (or (5)) to hold is given by

(P) For $u_i \in C_i^{P-}$ (or $C_i^{P+}$) with $1 \leq i \leq M$, $u_1 + u_2 + \cdots + u_M = 0$, if and only if $u_i = 0$ for $1 \leq i \leq M$.

Example 1: Suppose that for $1 \leq M \leq M$, the supports [12] of $C_1, C_2, \ldots, C_M$ are mutually disjoint (this property is called “DS structure” in [13]). Let $m(C_i)$ denote the effective length of $C_i$, i.e., the size of the support of $C_i, 1 \leq i \leq M$. If we use a symbol ordering such that for $1 \leq i \leq M$, any codeword of $C_i$ has nonzero components only from the $(\sum_{i=1}^{M} m(C_i)/k_i + 1)$th to $(\sum_{i=1}^{M} m(C_i)/k_i + 1)$th symbol positions, then the logarithm of the number of states at any symbol position of the minimal trellis $T$ for $C = C_1 + C_2 + \cdots + C_M$ is upper-bounded by

$$s_{\text{max}}(T) = \max_{1 \leq i \leq M} \min(k_i, m(C_i) - k_i).$$

Note that $C$ satisfies the definition of direct-sum in [14].

III. CONSTRUCTIONS OF GENERALIZED CONCATENATED CODES

Suppose that $n$ is the product of two integers $n_I$ and $n_G$ greater than one. For $1 \leq i \leq M$, let $C_{iI}$ be an $(n_I, k_{iI})$ linear code over GF$(q)$ such that

(DI) for $u_i \in C_{iI}$ with $1 \leq i \leq M$, $u_1 + u_2 + \cdots + u_{M} = 0$, if and only if $u_i = 0$ for $1 \leq i \leq M$.

Let $d_i$ be the minimum Hamming distance of $C_{iI} + C_{i+1I} + \cdots + C_{iMI}$. Let $C_{iO}$ be an $(n_O, k_{iO}, d_{iO})$ linear code over GF$(q^{d_{iO}})$ and let $C_{i}^{\ast}$ denote the concatenated code over GF$(q)$ with $C_{iI}$ as the inner code and $C_{iO}$ as the outer code. The $n$ code symbol positions in $C_{i}^{\ast}$ are divided into $n_O$ consecutive sections of length $n_I$ in such a way that each section of a codeword in $C_{i}^{\ast}$ is a codeword in $C_{iI}$.

Let the generalized concatenated code $C$ be defined as

$$C \triangleq C_1^{\ast} + C_2^{\ast} + \cdots + C_M^{\ast}. \quad (9)$$
Then, the condition (D) on \( C \) follows from the condition (D_1). The minimum Hamming distance \( d \) of \( C \) is lower-bounded [15] as
\[
d \geq \min_{1 \leq i \leq M} \delta_i d_{O_i} \tag{10}
\]
and a multistage decoding [15] up to the distance given by the right-hand side of (10) is possible. Let \( T^{(n_0)} \) and \( T^{(n_i)} \), with \( 1 \leq i \leq M \), denote the minimal \( n_0 \)-section trellis diagrams for \( C \) and \( C_i \), respectively, for which each section has length \( n_i \). Then the assumption (D_1) guarantees that if \( j \) is a multiple of \( n_i \) then property (P) holds. Therefore, at each end of a section,
\[
s_j(T^{(n_0)}) = \sum_{i=1}^{M} s_j(T^{(n_i)}). \tag{11}
\]

The above result means that, if there is a common optimal ordering of sections that gives the smallest state complexity of an \( n_0 \)-section trellis diagram for each component code \( C_i \), then the ordering of sections is optimal for the whole code \( C \). See also [16, Theorem 8.1].

A particular class of binary GC codes over GF(\( q \)) can be constructed as follows. An \( (n_0, n_1, n_2) \) code \( C_1 \), over GF(\( q \)), is partitioned into a chain of \( M \) \( (n_1, k_i, d_i) \) subcodes \( C_i \), \( i = 2, 3, \ldots, M+1 \), such that
\[
C_1 \supseteq C_2 \supseteq \cdots \supseteq C_{M+1}
\]
where, for convenience, we define \( C_{M+1} \triangleq \{0\} \), and \( d_{M+1} \triangleq \infty \). Let \( C_{i+1} = [C_i/C_{i+1}] \) denote an \( (n_1, k_i, d_i) \) subcode of \( C_i \), a set of coset representatives of \( C_{i+1} \) in \( C_i \), of dimension \( k_i = k_i - k_{i+1} \), and minimum Hamming distance \( \delta_i \geq d_i \). Then \( C_1 \) has the following coset decomposition [3]:
\[
C_1 = C_{i_1} + C_{i_2} + \cdots + C_{i_M}. \tag{12}
\]

Let \( C_{O_i} \) denote an \( (n_0, k_{O_i}, d_{O_i}) \) code \( C_{O_i} \), over GF(\( q^{h_i} \)), where
\[
k_{i} = \dim(C_i/C_{i+1}) = k_i - k_{i+1}, \quad i = 1, 2, \ldots, M.
\]

A GC code \( C \) is constructed from (9) as a direct sum of concatenated codes
\[
C = C_{O_1} \ast C_{i_1} + C_{O_2} \ast C_{i_2} + \cdots + C_{O_M} \ast C_{i_M}
\]
where \( C_{O_i} \ast C_{i_i} \) denotes a concatenated code with \( C_{O_i} \) as outer code and \( C_{i_i} \) as inner code, \( 1 \leq i \leq M \). It was shown in [8] that \( C \) is an \( (n_{O \cap i}, k, d) \) linear block code of dimension and minimum Hamming distance
\[
k = \sum_{i=1}^{M} k_{i} k_{O_i} \quad \text{and} \quad d \geq \min_{1 \leq i \leq M} \{\delta_i d_{O_i}\} \tag{13}
\]
respectively. Note that equality holds in (13) when \( C_{i_i} \), \( 1 \leq i \leq M \), contains the all-zero codeword, which is the case for all the codes presented in this correspondence.

**Example 2:** In this example, the trellis structure of a simple binary GC code is illustrated. Let \( n_1 = n_0 = 4 \), and consider the binary code \( C_1 = RM_2.2 \). Then \( M = 3 \) and \( C_1 = C_{i_1} + C_{i_2} + C_{i_3} \), where \( C_{i_1} = RM_2.2/RM_2.1 \) with generator matrix \( G_{i_1} = (001) \), \( C_{i_2} = RM_2.1/RM_2.0 \) with \( G_{i_2} = (101) \), and \( C_{i_3} = RM_2.0 \) with \( G_{i_3} = (111) \). Let \( C_{O_1} \) be a binary (4, 1, 4) repetition code, \( C_{O_2} \) be a nonbinary (4, 3, 2) over GF(\( 2^2 \)), and \( C_{O_3} \) be a binary (4, 4, 1) universal code. Then \( C = C_{O_1} \ast C_{i_1} + C_{O_2} \ast C_{i_2} + C_{O_3} \ast C_{i_3} \) is a binary GC (16, 11, 4) code. The four-section trellis diagrams of codes \( C_{O_j} \ast C_{i_j} \), \( j = 1, 2, 3 \), are shown in Fig. 1 where \( GF(2^2) = \{0, 1, \alpha, \alpha^2\} \), with \( \alpha^2 = 1 + \alpha \). In the trellis for Fig. 1(b), each element in the binary vector space \( \{0, 1\}^2 \), isometric to GF(\( 2^2 \)), is mapped onto a codeword in \( C_{i_2} \). As a result, the following four-bit vector representation of GF(\( 2^2 \)) is obtained: \( 0 = 0000 \), \( 1 = 0011 \), \( \alpha = 0101 \), and \( \alpha^2 = 0110 \).

The corresponding four-section diagram of the binary GC (16, 11, 4) code is shown in Fig. 2. It consists of two parallel and identical, up to branch labeling, subtrellises. Each subtrellis is in turn isometric, up to parallel branches, to the subtrellis of code.
$C_{02} + C_{T2}$. Also, each set of parallel branches has labels
\[
\{(ba_1b_2b_3), (ba_1b_2b_3) + (1111)\} = \{(ba_1b_2b_3), (ba_1b_2b_3)\}.
\]

The two parallel subtrellises have branch labels that differ by (0001), i.e., if $(ba_1b_2b_3)$ is the label of a branch in the upper subtrellis, the $(ba_1b_2b_3) + (0001)$ is the label of a branch in the lower subtrellis. \hfill $\square$

Let $s(T_{(nO)})$ denote the logarithm base $q$ of the maximum number of states of a minimal $n_O$-section trellis for the $i$th-level concatenated code $C_{O1} + C_{T1}$. 1 \leq i \leq M$, and let $s(T_{(nO)})$ denote the logarithm base $q$ of the maximum number of states of an $n_O$-section trellis for the overall GC code. Then it follows from (11) and Wolf’s bound [2] that
\[
s(T_{(nO)}) = \sum_{i=1}^{M} s(T_{(i\leftarrow nO)}) \leq \sum_{i=1}^{M} k_i \min \{k_{O1}, n_O - k_{O1}\}. \tag{14}
\]

In the following, several good binary GC codes are constructed that can be decoded using a multistage decoding up to their minimum distances. The GC construction is best explained by considering the following example.

**Example 3:** A binary $(q = 2)$ GC code of length 63 will be constructed. Let $n_1 = 7$ and $n_O = 9$. Consider the partition of a $(7, 7, 1)$ binary code $C_1$ into the following subcode chain:
\[
(7, 7, 1) \supset (7, 6, 2) \supset (7, 3, 4) \supset [\emptyset].
\]

Then $M = 3$ and $C_1$ can be expressed as $C_1 = C_{11} + C_{T2} + C_{T3}$, where $C_{11} = [(7, 7, 1) / (7, 6, 2)]$ is a $(7, 1, 1)$ code with codewords $\{0000000, 0000001\}$, $C_{T2} = [(7, 6, 2) / (7, 4, 3)]$ is a $(7, 3, 2)$ code and has a generator matrix
\[
G_{T2} = \begin{pmatrix}
0010001 \\
0001011 \\
0000111
\end{pmatrix}
\]

and $C_{T3} = [(7, 3, 4) / [\emptyset]]$ is equivalent to the dual of a Hamming code of length 7 with a generator matrix
\[
G_{T3} = \begin{pmatrix}
1010101 \\
0110011 \\
0001111
\end{pmatrix}
\]

Let $C_{O1}$ be a binary $(9, 2, 6)$ code, the product of a $(3, 1, 3)$ binary code and a $(3, 2, 2)$ binary code, and let $C_{O2}$ and $C_{O3}$ be $(9, 7, 3)$ and $(9, 8, 2)$ maximum-distance-separable (MDS) codes over GF$(2^3)$, respectively. Then it follows from (9) and (13) that $C$ is a binary $(63, 47, 6)$ GC code. \hfill $\square$

**TABLE I**

<table>
<thead>
<tr>
<th>$C$</th>
<th>$M$</th>
<th>$n$</th>
<th>$k_{O1}$</th>
<th>$k_{T2}$</th>
<th>$k_{T3}$</th>
<th>$s(T_{(nO)})$</th>
<th>$s(T_{(nO)})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(63, 47, 6)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>(63, 43, 8)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(63, 24, 16)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(72, 52, 8)</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(64, 45, 8)</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(64, 25, 16)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(64, 48, 6)</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(64, 37, 10)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(64, 34, 12)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(64, 28, 14)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

The last three codes in Table I of this correspondence were also presented (up to a possible permutation of positions) in [18]. All codes listed in Table I, except the last row, have the same parameters $n$, $k$, and $d$, as the best linear codes known [17]. The $(64, 45, 8)$ GC code has the same rate, minimum distance and decoding complexity (see next section) as a $(64, 45, 8)$ extended and permuted BCH code.

In this section, the decoding complexity of some of the proposed GC codes is analyzed and compared with that of RM, BCH, or EG codes listed in Table I, except the last row, have the same parameters $n$, $k$, and $d$, as the best linear codes known [17]. The $(64, 45, 8)$ GC code has the same rate, minimum distance and decoding complexity (see next section) as a $(64, 45, 8)$ extended and permuted BCH code. Apparently, these codes are equivalent, as are the $(64, 37, 10)$ GC code and a $(64, 37, 10)$ extended and permuted EG code according to [18]. Some of the GC codes in Table I have either smaller decoding complexities than BCH or EG codes of comparable rate, as it is shown in the next section, or more information bits for the same minimum distance.

Table II lists the parameters of other binary GC codes of longer lengths constructed based on other choices of the inner code $C_1$. The first three codes listed in Table II are constructed based on a $(5, 5, 1)$ code and its $(5, 4, 2)$ subcode, while the remaining codes are based on RM, $r = 4, 3, 2$, and its RM subcodes. It is important to note that, for binary GC codes in general, equality in (10) does not always hold. This is to say that the right-hand side (RHS) of (10) is in most cases significantly lower than the actual minimum distance $d$. For all the codes listed in Tables I and II, however, equality holds in (10) and a trellis-based multistage decoding up to their minimum distances can be employed.

**IV. Decoding Complexity**

In this section, the decoding complexity of some of the proposed GC codes is analyzed and compared with that of RM, BCH, or EG codes of the same lengths and the same, or approximately the same, rates and minimum distances.
Consider a binary linear block code and its trellis-based soft-decision decoding using the Viterbi algorithm. To update the branch metrics, one addition operation is performed per branch, except for the first code symbol. On the other hand, the number of comparisons at each state, to determine the survivor branch sequence, equals the number of arriving branches minus one. It follows that the total number of addition-equivalent decoding operations is

$$\psi(T) = \sum_{j=1}^{n} 2 \times 2^{h_j(T)} - 2^{2^{j(T)}} - 2^{3^{j(T)}}$$  \hspace{1cm} (15)$$

where \( h_j(T) \) denotes logarithm base 2 of the number of branches in \( T \) for the \( j \)th code symbol position.

The number of addition-equivalent decoding operations can be reduced dramatically by using the recursive MLD algorithm proposed in [19]. In this correspondence, the number of addition-equivalent decoding operations by the recursive MLD algorithm, denoted \( \psi^{(0)}_{\text{MLD}} \) in [19], is also used as a complexity measure. In comparing the constructed GC codes with BCH and EG codes, the permutations presented in [19] are considered.

Table III lists the state complexity, \( s_{\text{min}}(T) \), \( (3) \), and the total number of addition-equivalent decoding operations for both Viterbi decoding, \( \psi(T) \), and recursive MLD using optimally sectionalized trellis diagrams, \( \psi^{(0)}_{\text{MLD}} \), of some of the GC codes constructed in the previous section, compared with those of permuted RM, EG, or BCH codes of approximately the same rates and minimum distances. The last two codes listed in the table, (64, 47, 6) and (64, 48, 6) GC codes, have a considerably reduced decoding complexity compared to either EG or BCH codes of the same rate and minimum distance. The last two codes listed in the table, (64, 34, 12) and (64, 28, 14) GC codes, both have reduced decoding complexity in comparison with BCH codes of the same minimum distance.

**V. TWO-STAGE SOFT-DECISION DECODING**

The multilevel structure of the GC codes constructed in this correspondence allows for the use of a suboptimal trellis-based multistage decoding. Consider the following trellis-based two-stage soft-decision decoding of a GC code \( C \). Let \( C \) be expressed as a
The minimum distance of $C_{G2}$ is 4. However, the codewords of $C_{G2} + C_{G2}$ in correspondence to different values of the first 16 information bits, are at a minimum distance 8.

In the second decoding stage, a codeword in $C_{G1}$, in correspondence to the $k_{1}$ information bits decoded in the first stage, $\mathbf{r} = (u_{1}, u_{2}, \cdots, u_{n})$, is used to obtain a modified received sequence $\hat{r} = (r'_{1}, r'_{2}, \cdots, r'_{n})$, where $r'_i = (-1)^{k_i}r_i$ (assuming binary-phase shift keying (BPSK) modulation over an additive white Gaussian noise (AWGN) channel). An MLD for code $C_{G2}$ is used. At the end of this final decoding stage, the $k_{2} = \sum_{i=1}^{M} k_{i_{j}}$ remaining information bits are decoded.

For $i = 1, 2$, let $\psi_{i}^{(0)}$ denote the number of addition-equivalent decoding operations in the $i$th decoding stage for the recursive MLD using optimally sectionalized trellis diagrams. Let $\psi_{T_{SD}}^{(0)}$ denote the total number of addition-equivalent decoding operations in the above two-stage soft-decision decoding procedure. Then it follows that

$$\psi_{T_{SD}}^{(0)} = \psi_{1}^{(0)} + \psi_{2}^{(0)}.$$  

It should be noted that this reduced decoding complexity comes at the expense of a moderate loss due to an increased number of nearest neighbors (NN).

The values of $\psi_{T_{SD}}^{(0)}$ for selected GC codes are shown in the last column of Table III. Note the dramatic reduction in decoding complexity using two-stage soft-decision decoding. As an example, note that for the (64, 45, 8) GC code, two-stage decoding ($\psi_{T_{SD}}^{(0)} = 34, 842$) is about one order of magnitude less computationally intensive than optimal trellis-based recursive maximum-likelihood decoding of the (64, 42, 8) RM code ($\psi_{min}^{(0)} = 326, 017$). It is also worthwhile to note that, although suboptimal, the above two-stage decoding of the GC codes in Table III is up to the minimum distance of the code. To illustrate the loss due to the increased NN, Fig. 3 shows simulation results on the error performance of the (64, 45, 8) GC code, with both MLD and TSD. The loss is only about 0.3 dB compared to optimum MLD.

### VI. CONCLUSIONS

In this correspondence, binary generalized concatenated (GC) codes with very low decoding complexity have been constructed. The decoding complexity was measured both by the maximum number of states of a minimal trellis diagram and by the number of addition-equivalent operations of a Viterbi decoder. Many of the GC codes presented have the same parameters as the best linear codes known.

In addition, some of the GC codes have significantly smaller trellis-based decoding complexity than that of BCH and EG codes of the same length and approximately the same rate and minimum distance. Moreover, a trellis-based two-stage soft-decision decoding up the minimum distance was presented. The procedure was shown to drastically reduce the decoding effort, compared to maximum-likelihood decoding. The GC codes presented in this correspondence offer an excellent tradeoff between decoding complexity and error performance.

### REFERENCES


### TABLE III

The Decoding Complexity of Selected GC Codes

<table>
<thead>
<tr>
<th>Code, C</th>
<th>Basin</th>
<th>$k_{1}$</th>
<th>$\psi_{T_{SD}}^{(0)}$</th>
<th>$\psi_{T_{MLD}}^{(0)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC (64, 45, 8)</td>
<td>4</td>
<td>12</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>EG (64, 45, 8)</td>
<td>5</td>
<td>13</td>
<td>25</td>
<td>48</td>
</tr>
<tr>
<td>GC (64, 45, 8)</td>
<td>6</td>
<td>14</td>
<td>50</td>
<td>92</td>
</tr>
<tr>
<td>RM (64, 42, 8)</td>
<td>7</td>
<td>15</td>
<td>73</td>
<td>135</td>
</tr>
<tr>
<td>BCH (31, 24, 15)</td>
<td>8</td>
<td>16</td>
<td>97</td>
<td>182</td>
</tr>
<tr>
<td>GC (72, 32, 8)</td>
<td>9</td>
<td>17</td>
<td>121</td>
<td>220</td>
</tr>
<tr>
<td>GC (72, 32, 8)</td>
<td>10</td>
<td>18</td>
<td>145</td>
<td>260</td>
</tr>
<tr>
<td>GC (64, 45, 8)</td>
<td>11</td>
<td>19</td>
<td>170</td>
<td>310</td>
</tr>
<tr>
<td>BCH (31, 24, 15)</td>
<td>12</td>
<td>20</td>
<td>194</td>
<td>352</td>
</tr>
<tr>
<td>GC (64, 45, 8)</td>
<td>13</td>
<td>21</td>
<td>218</td>
<td>396</td>
</tr>
<tr>
<td>GC (64, 45, 8)</td>
<td>14</td>
<td>22</td>
<td>242</td>
<td>432</td>
</tr>
<tr>
<td>RM (64, 42, 8)</td>
<td>15</td>
<td>23</td>
<td>266</td>
<td>492</td>
</tr>
<tr>
<td>BCH (31, 24, 15)</td>
<td>16</td>
<td>24</td>
<td>290</td>
<td>530</td>
</tr>
</tbody>
</table>

### direct sum $C = C_{G1} + C_{G2}$, where

$C_{G1} \triangleq C_{G1} \ast C_{G2} + C_{G1} \ast \cdots + C_{G1} \ast C_{G1}$  

$C_{G2} \triangleq C_{G2} \ast C_{G2} + C_{G2} \ast \cdots + C_{G2} \ast C_{G2}$

with $1 \leq L \leq M$.

**Example 4:** Let $C$ be the (64, 45, 8) GC code in Table III and let

$C_{G1} = (8, 1, 8) \ast RM_{3}, 3/3, RM_{3}, 2 + (8, 5, 4) \ast RM_{3}, 2/3, RM_{3}, 1$

$C_{G2} = (8, 7, 2) \ast RM_{3}, 1/3, RM_{3}, 0 + (8, 8, 1) \ast RM_{3}, 0$.  

Then $M = 4$ and $L = 2$.

As a general design rule of an $M$-level GC code, the component codes $C_{G1} \ast C_{G2}$ at the first $L$ partition levels ($i = 1, 2, \cdots, L, L \leq M$) should be selected so as to have a large minimum Hamming distance, and yet simply a trellis structure. This is in order to guarantee that decisions are correct in the first decoding stage with high probability, resulting in good error performance.

The first decoding stage is MLD for the super code $C_{G1} + C_{G2}$, where $C_{G2} \subseteq C_{G2}$. Code $C_{G2}$ is chosen such that it has smaller decoding complexity than $C_{G2}$. For all the codes presented in Table III, code $C_{G2}$ has the same inner codes as $C_{G2}$, and a single $(n_{G2}, n_{G2} - 1, 2)$ or $(n_{G2}, n_{G2} - 1, 2)$ code as outer code. After the most likely codeword is determined in this stage, the first $k_{1} = \sum_{i=1}^{M} k_{i_{j}}$ information bits are decoded.

Note that the minimum distance of code $C_{G1} + C_{G2}$ is smaller than or equal to the minimum distance $d$ of $C$. However, for all the GC codes in Table III, the codewords of $C_{G1} \ast C_{G2}$ in correspondence to different values of the first $k_{1}$ information bits, are at a distance at least $d$. In other words, in the first stage, the information bits are decoded up to the minimum distance of the code.

**Example 4 (Cont.):** For the (64, 45, 8) GC code

$C_{G2} = (8, 8, 1) \ast RM_{3}, 1/3, RM_{3}, 0 + 8, 8, 1) \ast RM_{3}, 0$

$= (8, 8, 1) \ast RM_{3}, 1$.
A Low-Weight Trellis-Based Iterative Soft-Decision Decoding Algorithm for Binary Linear Block Codes

Takuya Koumoto, Toyoo Takata, Tadao Kasami, Life Fellow, IEEE, and Shu Lin, Fellow, IEEE

Abstract—This paper presents a new low-weight trellis-based soft-decision iterative decoding algorithm for binary linear block codes. The algorithm is devised based on a set of optimality conditions and the generation of a sequence of candidate codewords for an optimality test. The initial candidate codeword is generated by a simple decoding method. The subsequent candidate codewords, if needed, are generated by a chain of low-weight trellis searches, one at a time. Each search is conducted through a low-weight trellis diagram centered around the latest candidate codeword and results in an improvement over the previous candidate codewords that have been already tested. This improvement is then used as the next candidate codeword for a test of optimality. The decoding iteration stops whenever a candidate codeword is found to satisfy a sufficient condition on optimality or the latest low-weight trellis search results in a repetition of a previously generated candidate codeword. A divide-and-conquer technique is also presented for codes that are not spanned by their minimum-weight codewords. The proposed decoding algorithm has been applied to some well-known codes of lengths 48, 64, and 128. Simulation results show that the proposed algorithm achieves either practically optimal error performance for the example codes of length 48 and 64 or near optimal error performance for the (128, 29, 32) RM code with a significant reduction in computational decoding complexity.

Index Terms—Iterative decoding, low-weight subtrellis, optimality.

I. INTRODUCTION

The application of trellis-based maximum likelihood decoding (MLD) algorithms is limited due to the prohibitively large trellises for codes of long block lengths. To overcome the state and branch complexity problems of large trellises for long block codes, several new approaches have been proposed [1]–[8]. Most recently, Moorthy et al. have shown that the minimum-weight subtrellis of a code is sparsely connected and has much simpler state and branch complexities than the full-code trellis [9]. Based on this fact, they proposed a minimum-weight subtrellis-based iterative decoding algorithm for linear block codes to achieve suboptimum error performance with a drastic reduction in decoding complexity compared with a trellis-based MLD algorithm, using a full-code trellis. The Moorthy–Lin–Kasami (MLK) algorithm is devised based on the following: 1) generation of a sequence of candidate codewords based on a set of test error patterns using the Chase Algorithm II [10] and an algebraic decoder; 2) two test conditions: one to test the optimality of a candidate codeword and the other to test whether the most likely (ML) codeword is at a distance no greater than the minimum distance from the tested candidate codeword; and 3) a minimum-weight

Manuscript received August 5, 1997; revised August 10, 1998. This work was supported under NSF under Grant NCR-9415374, NASA under Grant NAG5-931, and the Ministry of Education, Japan, Grant 06650416. T. Koumoto is with the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. T. Takata is with the Faculty of Software and Information Science, Iwate Prefectural University, Iwate, Japan. T. Kasami is with the Faculty of Information Science, Department of Computer Science, Hiroshima City University, Hiroshima, Japan. S. Lin is with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA. Communicated by A. Vardy, Associate Editor for Coding Theory. Publisher Item Identifier S 0018-9448(99)01416-9.