Character-Correcting Convolutional Self-Orthogonal Codes

S. Y. Tong

Bell Telephone Laboratories, Inc., Holmdel, New Jersey 07733

A class of convolutional, character-error-correcting codes with limited error propagation is presented. This class of codes is derived from binary convolutional self-orthogonal codes (BCSOC). By character-error-correcting, we mean that the code is character oriented, where each character can be thought of as a string of binary or higher base symbols of fixed length or as a single nonbinary symbol of correspondingly higher base. It is shown that, given a $t$-error-correcting BCSOC of rate $b - 1/b$, a character-error correcting convolutional self-orthogonal code (CCSOC) of rate $k(b - 1)/(k(b - 1) + 1)$ can be constructed for any integer $k$, the rate expansion factor. The CCSVCC so constructed corrects $t$ character errors, and also possesses large simultaneous burst-error-correcting capabilities. Lower bounds on the burst-error-correcting capability for both BCSOC and CCSVCC are found.

Decoding consists of a mixture of majority logic decoding and algebraic computation. The decoding algorithm seems practical if either the rate expansion factor $k$ or the number of errors corrected $t$ are not large. Such codes are most suitable for channels with both random and burst noise, and also effect a compromise between the cost of terminal equipment and the efficient use of channels.

INTRODUCTION

Random error correcting convolutional codes defined over a nonbinary finite field were first studied by Ebert and Tong (Ebert–Tong, 1969). These codes are highly efficient and, in one alternate form, optimal. However, they are generally difficult to decode.

In this report, a different construction of random-error-correcting convolutional codes over a nonbinary finite field is proposed. These codes combine the number-theoretic construction of the Robinson–Bernstein codes (Robinson–Bernstein, 1967) with the algebraic properties of the Bose–Chaudhuri–Hocquenghen codes. In Sections II and III the construction procedures and justifications are given. In Section IV decoding algorithms are discussed. In Section V the burst-error-correcting capabilities of codes in this class are found.
I. Definitions and Preliminaries

Linear convolutional codes can be described by a semiinfinite parity check matrix \( A \). The code words are semiinfinite sequences \( X = (x_1, x_2, \ldots) \) such that

\[
A \cdot X = 0. \tag{1}
\]

We consider such codes over a finite field \( GF(p^l) \). The elements of \( GF(p^l) \) are called characters; the "character size" of \( GF(p^l) \) is \( l \). The matrix \( A \) is given schematically in Fig. 1, where \( b \) is the smallest integer such that an \( N \times b \) matrix \( B_0 \) can generate the matrix \( A \); as shown in the figure, \( b \) is called the basic block length of the code and \( m \) is the number of check characters per basic block. The code so defined has rate \( (b - m)/b \) and its actual constraint length is given by \( n = (b/m) \cdot N \); we assume that \( m \) divides \( N \) evenly.

![Fig. 1. Schematic diagram of the semiinfinite parity check matrix \( A \). The nonzero elements are in the shaded areas.](image)

Since the decoding of a convolutional code proceeds sequentially by blocks of \( b \) characters the error-correcting properties of the codes are determined by the decoding of the first block. Thus it is sufficient to examine only \( A_N \), the first \( N \) rows of \( A \).

The minimum distance \( d \) of the code is defined as the smallest number of nonzero characters in the first \( n \) positions of \( X \) such that \( A_N \cdot X = 0 \), provided there is at least one nonzero character in the first basic block. That
is, $d$ is the smallest number of columns of $A_n$, including at least one column from the first block, whose linear combination over $GF(p^t)$ is zero. Clearly, every pattern of $t$ or fewer errors in $n$ consecutive bits can be corrected if and only if $d \geq 2t + 1$. Generally, $t$ will be used to denote the error-correcting capability.

II. Construction Procedure

For the sake of simplicity we shall limit our discussion to codes with $m = 1$.

Given a $B_0$ matrix of a binary convolutional self-orthogonal code (BCSOC) of rate $(b - 1)/b$, the $B_0$ matrix of a character-correcting convolutional self-orthogonal code (CCSOC) of rate $k(b - 1)/(k(b - 1) + 1)$ is constructed by the following procedure:

1. Given $B_{0b}$, the $B_0$ matrix of a BCSOC, for the $i$-th nonzero element in the $j$-th column of $B_{0b}$, replace it by a row vector $(1, \alpha^{i-1}, \ldots, \alpha^{k(i-1)})$, where $\alpha$ is a primitive element of $GF(p^t)$.

2. Replace all zero entries by zero row vectors of dimension $k$.

3. The last column (check column) of $B_{0b}$ remains unchanged.

The parameter $k$ is called the rate expansion rate factor.

Example 1. From a rate $2/3$ double-error-correcting ($t = 2$) BCSOC a CCSOC of rate $6/7$ is constructed as shown below, where $k = 3$.
To get a CCSOC of rate slightly lower, say, 5/6, one may delete any one of the columns of the rate 6/7 except the last, which is the check column.

It is seen that $B_{60}$ matrix can be thought of as the concatenation of the matrices $D_1, D_2, \ldots, D_{b-1}$ and check column $C$; i.e.,

$$B_{60} = D_1 : D_2 : D_3 : \cdots : D_{b-1} : C,$$

where $D_i$ is the expansion matrix derived from the $i$-th column of the $B_{60}$.

In the previous example,

$$D_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha^4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & \alpha^3 & \alpha^6 \\ 0 & 0 & 0 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha^4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & \alpha^3 & \alpha^6 \end{bmatrix}.$$

**DEFINITION.** Matrix $D_i'$ is defined as the matrix consisting of all nonzero rows of $D_i$.

In the example,

$$D_1' = D_2' = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha^4 \\ 1 & \alpha^3 & \alpha^6 \end{bmatrix}.$$

Observe that the $D_i'$ are all the same. Therefore, we shall drop the subscript in the sequel.

**Theorem 1.** A CCSOC that is generated by a $t$-error correcting BCSOC of rate $(b - 1)/b$ corrects $t$-character errors if and only if all the $\lambda$-th order minors of $D'$ are nonsingular:

$$\lambda = 1, 2, \ldots, \min(k, 2t).$$
Proof. Consider any linear combination of $\lambda$ columns of $D_i$, and let $S$ be the resultant column vector. We have

$$S = D_i \cdot I,$$

where $I$ is a column vector whose entries are the coefficients of the linear combination. Let $W(\ast)$ be the number of nonzero entries of $\ast$. Clearly, the weight of $I$ is $W(I) = \lambda$. Recall that $D_i$ is generated by the $i$-th column of the BCSOC and all columns of BCSOC are self-orthogonal. It follows that every column other than those in $D_i$ can cancel at most one nonzero entry of $S$. Thus at least $W(S)$ additional columns must be combined with $S$ to produce a codeword. This codeword has weight $W(I) + W(S) = \lambda + W(S)$. Hence the minimum distance $d$, of the code is $d = \min \{ \lambda + W(S) \}$. If one can assure $W(S) \geq 2t - \lambda + t$, then $d = \min \{ \lambda + W(S) \} \geq \min(2t + 1) = 2t + 1$ and the code can correct $t$ errors.

To assure this, observe that (2) is equivalent to

$$S' = D' \cdot I$$

by deleting all the zero rows of $D_i$. Obviously, $W(S') = W(S)$ since only zero entries of $S$ are deleted. To assure $W(S') \geq 2t - \lambda + 1$, for any set of $\lambda$ equations among the $2t$ equations of (3), it is necessary that there be at most $\lambda - 1$ zeroes in $S'$. But that is equivalent to requiring that any matrices $E_{\lambda}$, formed by first picking any $\lambda$ columns of $D'$ then picking any $\lambda$ rows of the $\lambda$ chosen columns, must be nonsingular. But $E_{\lambda}$ is precisely a $\lambda$-th order minor of $D'$. Thus $W(S') = W(S) \geq 2t - \lambda + 1$, if all $E_{\lambda}, \lambda = 1, 2, ..., \min(k, 2t)$, are nonsingular.

Conversely, if a particular $E_{\lambda}$ is singular, then there exists an $I$ such that $E_{\lambda} \cdot I = 0$.

Hence there is one combination of $\lambda$ columns that makes at least $\lambda$ components of $S'$ zero.

i.e.,

$$W(S') \leq 2t - \lambda.$$

One can construct a codeword by using this particular combination of $\lambda$ columns, and since only $W(S')$ nonzero columns remain, one can cancel each of these nonzero entries of $S'$ by choosing the parity check column corresponding to that nonzero entry. There are a total of $W(S')$ such check columns so that the code has minimum weight $= W(S') + \lambda \leq 2t$, therefore $d \leq 2t$. Hence the code does not correct all $t$-error patterns. Q.E.D.
Observe that the \( \mu \)-th order of the minors of \( D' \) are of the form

\[
m\mu(\alpha) = \begin{vmatrix}
\alpha^{i_1 j_1} & \cdots & \alpha^{i_1 j_\mu} \\
\vdots & & \vdots \\
\alpha^{i_\mu j_1} & \cdots & \alpha^{i_\mu j_\mu}
\end{vmatrix},
\]

where

\[
0 \leq i's \leq k - 1,
\]

\[
0 \leq j's \leq 2t - 1,
\]

which can be rewritten as

\[
m\mu(x) = \begin{vmatrix}
x_1^{i_1} & \cdots & x_\mu^{i_1} \\
x_1^{i_2} & \cdots & x_\mu^{i_2} \\
\vdots & & \vdots \\
x_1^{i_\mu} & x_\mu^{i_\mu}
\end{vmatrix}.
\]

Such a determinant is called an alternant in the variables

\[(x_1 \cdots x_\mu),\]

where \( x_s = \alpha^{j_s}, s = 1, 2, \ldots, \mu. \)

To assure that the minors of \( D' \) are nonsingular, one must find out the zeroes of \( m\mu(\alpha) \) and select \( \alpha \) so that \( m\mu(\alpha) \) is nonzero for all \( \mu \). In the following section, the zeroes of \( m\mu(\alpha) \) are evaluated, and from this, constraints on \( \alpha \) are determined.

III. Evaluation of the Zeros

Consider all the minors of \( D' \), a \( k \times 2t \) matrix. Let \( \tau = \min(k, 2t) \). Then the minors to be considered are \( m\mu(\alpha) \), or simply \( m_\mu, 1 \leq \mu \leq \tau. \)

1. Every \( 1 \times 1 \) minor of \( D' \), \( m_1 \), is nonsingular since all entries of \( D' \) are powers of \( \alpha \), which is nonzero.

2. Every \( 2 \times 2 \) minor, \( m_2 \), is nonsingular if \( \alpha \) is a primitive element of \( GF(p^l) \), where

\[
l \geq \log_p[(k - 1)(2t - 1) + 2].
\]
Proof. Without loss of generality, assume $i_1 < i_2, j_1 < j_2$. Then

\[ m_2 = \begin{vmatrix} \alpha^{i_1j_1} & \alpha^{i_1j_2} \\ \alpha^{i_2j_1} & \alpha^{i_2j_2} \end{vmatrix} \]

where

\[ 0 \leq i_1 < i_2 \leq k - 1, \]
\[ 0 \leq j_1 < j_2 \leq 2t - 1. \]

It follows that $m_2 \neq 0$ if and only if

\[ \alpha^{(i_2-i_1)(j_2-j_1)} \neq 1 \quad \text{for all} \quad i, j. \]

But $\max(i_2 - i_1, j_2 - j_1) = (k - 1)(2t - 1)$, as

\[ \alpha^{p^l-1} = 1 \quad \text{and} \quad \alpha^k \neq 1, \quad k < p^l - 1. \]

Therefore $m_2$ is nonsingular if

\[ p^l - 1 > (k - 1)(2t - 1) \]

or

\[ l \geq \log_p[(k - 1)(2t - 1) + 2] \]

(3) $m_r$ is a Van-der-monde determinant; therefore it is nonsingular if no pairs of columns (rows) are identical.

**Case 1.** $k \leq 2t$. Then $\tau = k$

\[ m_r = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha^{i_1} & \alpha^{i_2} & \cdots & \alpha^{(k-1)i_1} \\ 1 & \alpha^{i_2} & - & \cdots & - \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{i_{k-1}} & \cdots & \cdots & \alpha^{(k-1)i_{k-1}} \end{bmatrix} = 0, \]

if $\alpha^{i_r} = \alpha^{i_s}, 0 \leq i_r < i_s \leq \tau - 1$. 
Case 2. \( 2t < k \). Then \( \tau = 2t \),

\[
m_\tau = \begin{bmatrix}
1 & 1 & \cdots & 1 & 1 \\
\alpha^{j_1} & \alpha^{j_2} & \cdots & \alpha^{j_{2t-1}} \\
1 & \alpha^{2j_1} & \cdots & \alpha^{2j_{2t-2}} \\
\vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{(2t-1)j_1} & \cdots & \alpha^{(2t-1)j_{2t-1}}
\end{bmatrix}
\]

if \( \alpha^{j_p} = \alpha^{j_q}, \ 0 \leq j_p < j_q < \tau - 1 \). In particular, \( m_\tau \) will be nonsingular if

\[
p^t - 1 > (k - 1)(2t - 1),
\]

\[
l \geq \log_p [(k - 1)(2t - 1) + 2].
\]

From the above three results we see that for \( \tau \leq 3 \), the CCSOC defined by \( B_{\nu_0} \) is \( t \) error correcting, if \( \alpha \) is a primitive element of \( GF(p^t) \), where

\[
l \geq \log_p [(k - 1)(2t - 1) + 2].
\]

**Example 2.** The rate 6/7 CCSOC of Example 1 is double-character-correcting since

\[
\tau = \min(k, 2t) = \min(3, 4) = 3,
\]

provided that \( \alpha \) is a primitive element of \( GF(2^t) \), where

\[
l \geq \log_2 [(3 - 1)(4 - 1) + 2] = 3.
\]

**Special Codes**

Consider the case of \( t = 4 \). We have that \( m_1, m_2, \) and \( m_4 \) are nonsingular if

\[
l \geq \log_p [(k - 1)(2t - 1) + 2].
\]

To evaluate \( m_3 \), we have

\[
m_3 = \begin{bmatrix}
\alpha^{i_1j_1} & \alpha^{i_2j_1} & \alpha^{i_3j_1} \\
\alpha^{i_1j_2} & \alpha^{i_2j_2} & \alpha^{i_3j_2} \\
\alpha^{i_1j_3} & \alpha^{i_2j_3} & \alpha^{i_3j_3}
\end{bmatrix},
\]

where, without the loss of generality, by reordering \( i \) and \( j \) and by changing the designation of \( i \) and \( j \) we may assume

\[
0 \leq i_1 < i_2 < i_3 \leq t - 1 = 3
\]
and

\[ 0 \leq j_1 < j_2 < j_3 \leq \delta - 1, \]

where

\[ \delta = \max\{k, 2t\}. \]

Let

\[ x_1 = \alpha^{j_1}, \quad x_2 = \alpha^{j_2}, \quad x_3 = \alpha^{j_3}. \]

The determinant in (12) is called an alternant. Let us denote an alternant by its power indices, as shown in the Appendix. We have, by using (A4) in the Appendix,

\[ m_3 = | A(i_1, i_2, i_3) | = | A(0, 1, 2) | \cdot E. \quad (13) \]

There are four combinations of \( i_1, i_2, i_3 \), viz.,

\[ m_3 = | A(0, 1, 2) |, \quad (14) \]

or

\[ m_3 = | A(1, 2, 3) | = | A(0, 1, 2) | \cdot a_3, \quad (15) \]

or

\[ m_3 = | A(0, 2, 3) | = | A(0, 1, 2) | \cdot a_2, \quad (16) \]

or

\[ m_3 = | A(0, 1, 3) | = | A(0, 1, 2) | \cdot a_1, \quad (17) \]

where \( a_i \) is the \( i \)-th degree elementary symmetric function. Thus the zeroes of \( m_3 \) are the zeroes of \( | A(0, 1, 2) | \) plus the zeroes of \( a_1, a_2, a_3 \).

\[ | A(0, 1, 2) | \] is a Vandermonde determinant if

\[ l \geq \log_p ((k - 1)(2t - 1) + 2). \]

Therefore, \( A(0, 1, 2) \neq 0, \)

\[ a_3 = x_1 x_2 x_3 = \alpha^{j_1+j_2+j_3} \neq 0, \]

\[ a_2 = x_1 x_2 + x_2 x_3 + x_1 x_3 = \alpha^{j_1+j_2} + \alpha^{j_2+j_3} + \alpha^{j_1+j_3} \]

\[ = \alpha^{j_1+j_2}(1 + \alpha^{j_3-j_2} + \alpha^{j_2-j_1}), \]

\[ a_1 = x_1 + x_2 + x_3 = \alpha^{j_1} + \alpha^{j_2} + \alpha^{j_3} \]

\[ = \alpha^{j_1}(1 + \alpha^{j_2-j_1} + \alpha^{j_3-j_1}). \]
The zeroes of \( a_1 \) and \( a_2 \) are the zeroes of the polynomials

\[
f_1 = 1 + x^{j_2-j_1} + x^{j_3-j_1}
\]

and

\[
f_2 = 1 + x^{j_3-j_2} + x^{j_3-j_1},
\]

where

\[0 \leq j_1 < j_2 < j_3 \leq \delta - 1.\]

Thus, if \( \alpha \) is generated by a primitive polynomial \( G(x) \) over \( GF(p) \) of degree \( \delta \) or more, then \( F(\alpha) \neq 0 \) for all polynomial \( F \) of degree less than \( \delta \). It follows that \( a_1 \neq 0, a_2 \neq 0 \) if

\[l \geq \delta.\]

Thus a sufficient condition for \( m_3 \neq 0 \) is

\[l_m \geq \max\{\log_x[(k - 1)(2l - 1) + 2], \delta\},\]

where

\[\tau = \min(k, 2t) = 4.\]

Now

\[\delta = \max\{k, 2t\} \geq \log_x[(k - 1)(2l - 1) + 2]
\]

\[\min\{k, 2t\} = 4.\]

We have the simplified expression

\[l_m \geq \max\{k, 2t\},\]

if

\[\tau = \min\{k, 2t\} = 4.\]

The condition on \( l \) is an upper bound, sometimes it is possible to do better than this by choosing specific generator polynomials for \( \alpha \) such that \( a_1, a_2 \) are nonzero for all possible choices of \( j_1, j_2, j_3 \). Table I shows some codes obtained by exhaustive search.

Similarly, one may evaluate the character size for \( \tau = \min[k, 2t] = 5. \)
It can be shown in this case that the upper bound on the size of \( \alpha \) is

\[l_m \geq 2\tau - 3 = 2[\max\{k, 2t\}] - 3. \] (20)
TABLE I

<table>
<thead>
<tr>
<th>$d$</th>
<th>$t$</th>
<th>$l$</th>
<th>$G(x)$</th>
<th>$l_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>4</td>
<td>Unrestricted</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>5</td>
<td>Unrestricted</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>5</td>
<td>75$^a$</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
<td>75</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>5</td>
<td>75</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>6</td>
<td>147</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>6</td>
<td>147</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>6</td>
<td>147</td>
<td>11</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>6</td>
<td>147</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$k$</th>
<th>$l$</th>
<th>$G(x)$</th>
<th>$l_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2t</td>
<td>4</td>
<td>Unrestricted</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Unrestricted</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>75</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>75</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>75</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>147</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>147</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>147</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>147</td>
<td>12</td>
</tr>
</tbody>
</table>

$^a$ Octal representation of $G(x)$, e.g., is $75 = x^4 + x^3 + x + 1 = 111101$.

IV. THE DECODING TECHNIQUES AND ERROR PROPAGATION

We first note that decoding of CCSOC involves only the correct decoding of the first block of the code.

Consider a $t$-error correcting CCSOC, and assume that the first error occurs in the $i$-th subblock of the first block, and there are $\lambda$ errors ($\lambda \leq t$) in that subblock. It follows that at least $2t - (\lambda - 1)$ check symbols out of the $2t$ check symbols that check the subblock are nonzero. By hypothesis, there can be at most $t - \lambda$ additional errors in the first constraint length which are checked by the same set of check symbols; hence, at least $2t - (\lambda - 1) - (t - \lambda) = t + 1$ check symbols are nonzero. Thus, $\lambda$ errors...
(λ ≤ t) in the i-th subblock are detected whenever a majority of check symbols that check the subblock are nonzero. On the other hand, if there is no error in the i-th subblock, with at most t errors outside of the block, at most t check symbols that check the i-th subblock can be nonzero; hence, the majority detection rule still gives the right answer.

Given that errors in the i-th subblock are detected, the next step is to correct them. There are at least \(2t - (t - λ) = t + λ\) check equations that involve only the errors in the i-th block and nothing else. The λ error values and locations can be found by using any \(2λ\) of the \(t + λ\) equations, and the result must check with the \(t + λ\) check equations. Hence, after the effect of the errors are removed from the parity checks at most \(2t - (t + λ) = t - λ\) parity checks can be nonzero. It follows that the weight of the syndrome vector can only decrease if a correction (right or wrong) is made. Thus, even if a decoding error is made, the number of nonzero elements in the syndrome register must go to zero eventually. Thus, the decoder will recover from any decoding error; hence, error propagation is limited. It can be shown that the length of the maximum error propagation \(L\) is bounded in the same way as the BCSOC from which the CCSOC is derived. Such bounds have been derived by Robinson and Bernstein (1967).

From the above discussion, we see that, in principle, the CCSOC can be decode by exhaustive search. The correct solutions are those which check with at least \(t + 1\) parity checks. Since there are only \(k\) possible error locations and at most \(\min(k, t)\) errors can occur in the i-th subblock, the search is not unduly complex, if either \(k\) or \(t\) are reasonably small.

V. Decoding Algorithms for Some Special Codes

We present a simplified decoding algorithm for those CCSOC which satisfy the conditions

\[
\min[k, t] \leq 2. \tag{21}
\]

First, note that if (21) is satisfied, for each subblock, one may have none, one, or two errors in that block.

Given that there are no more than \(t\) errors in a constraint length, we see that

1. If there are no errors in the subblock, then at least \(t\) of the \(2t\) parity checks of that subblock must be zero.

2. Conversely, if a majority of the \(2t\) parity checks are nonzero, either one or two errors must have occurred in the subblock.
A. Assume that one error occurs at position \( j (j = 1, 2, \ldots, k) \). The checks are

\[
S_i = \alpha^{(j-1)}e_1 \quad (i = 0, 1, \ldots, 2t - 1)
\]

(22)

with a majority of

\[
\alpha^{-i(j-1)}S_1 = e_1
\]

(23)

for some value of \( i \). If the majority of the \( S_i \) are equal, the first position is in error; if not, one multiplies \( S_i \) by \( \alpha^{-1} \) and compares again. Do this recursively for \( k \) times. A single error is located and corrected by this procedure.

B. If no majority agreement is made in procedure \( A \), there can be at most two errors in the subblock.

Case 1. \( t = 2 \). The error may be corrected by solving the simultaneous equations

\[
S_i = \alpha^{ri}e_1 + \alpha^{si}e_2, \quad i = 0, 1, 2, 3.
\]

(24)

The error locations \( r, s \leq 1, 2, \ldots, k \) may be found by using Peterson's (Peterson, 1961) decoding algorithm for BCH codes, i.e., we solve the quadratic equation

\[
\sigma_0 x^2 - \sigma_1 x + \sigma_2 = 0,
\]

(25)

where

\[
\sigma_0 = S_0S_2 - S_1^2,
\]

\[
\sigma_1 = S_2S_0 - S_1S_2,
\]

\[
\sigma_2 = S_1S_3 - S_2^2.
\]

If no solution is found, errors are detected; otherwise the values \( e_1, e_2 \) are found by substituting the error locations and solving for (24).

Case 2. \( k = 2 \). The error locations are known, viz., \( r = 0, s = 1 \) from (24); then

\[
S_i = e_1 + \alpha^i e_2, \quad i = 0, 1, \ldots, 2t - 1.
\]

(26)

Consider the set

\[
U_i = \alpha^{-i}(S_i + S_{i+1}) = (1 + \alpha) e_2, \quad i = 0, 1, \ldots, 2t - 2.
\]

(26)

One partitions the set \( U_i \) into equivalent classes by the relation "equal." We claim that those \( S_i \) whose \( U_i \) belong to the class of largest membership are not corrupted by noise.
This statement is true since it is a special case of Lemma 2 of Reed and Solomon (1960).

From any $U_i$ of this class, we have

$$e_2 = \frac{U_i}{1 + \alpha}$$

and

$$e_1 = S_i + \frac{\alpha^i U_i}{1 + \alpha} = \frac{\alpha S_i + S_{i+1}}{1 + \alpha}. \quad (29)$$

If there is a tie in finding the largest class, errors are detected. Note that modification of this technique for $k \geq 3$ is possible, although the complexity of computation grows quickly.

VI. Burst Error-Correcting Capacity of Self-Orthogonal Codes

Following Wyner and Ash (1963), we first defined two types of bursts which will be considered in the sequel.

**Definition.** A type $B1$ burst of length $l$ is an error pattern that is confined to $l$ consecutive symbols.

**Definition.** A type $B2$ burst of length $l$ is a type $B1$ burst of length $l$ with the additional restriction that the burst is also confined to $l/b$ consecutive blocks; each block has $b$ symbols, and we assume $b \parallel l$.

The burst error-correcting capacities of BSCOC and its derivative, the CCSOC, are closely related. We shall first give a bound for BCSOC, then generalize it to CCSOC.

To simplify the discussion, we confine ourselves to a subclass of such codes that appear to be interesting.

The class of BCSOC we consider is that of the form due to Robinson and Bernstein (1967). The properties are:

(P1) Codes are of rate $(b - 1)/b$ ($b = 2, 3, \ldots$).

(P2) The $B_0$ matrix of the code has its first row all "1." That is, the check bit of the first block checks all the $b - 1$ information bits of that
block. We also restrict the CCSOC to be considered to those having the following property:

(P3) All CCSOC are derived from the R–B (Robinson–Bernstein) type BCSOC.

Consider the decoding of an R–B type BCSOC. We observe that

Situation A. If the $i$-th bit of the first block is in error, then the error is correctable if and only if, at most $t - 1$ of the $2t$ parity checks, $C_1 \cdots C_{2t}$ that check the $i$-th bit are changed by other errors.

Situation B. If the $i$-th bit of the first block is not in error, then no decoding error of the $i$-th bit is possible if and only if no more than $t$ of the set $\{C_i\}$ are changed by other errors.

By (P2) any other error in the first block can be involved in only $C_1$. Thus in Situation A the $i$-th error is correctable if no more than $t - 2$ parity checks of $C_2 \cdots C_{2t}$ are changed by errors in the subsequent blocks.

By (P1) there is one check bit in each block. Consider a block. If its check bit, say, $\theta$, is a member of $\{C_i\}$ then all information bits of the $i$-th block can have only one common check with the $i$-th bit of the first block, viz., $\theta$. As the code is self-orthogonal, there can be no other common checks. Conversely, if $\theta$ is not a member of $\{C_i\}$ then each information bit of the block can have at most one check common to the $\{C_i\}$. So there can be at most $b - 1$ check bits common to the information bits of that block and the $\{C_i\}$. Consequently, for any errors in the $j$-th block, at most $b - 1$ checks of $\{C_i\}$ may be changed. It follows that the number of subsequent blocks allowed to have errors is at least $[(t - 2)/(b - 1)]$.

Thus an error in the $i$-th position is correctable if the error pattern consists of $m$ blocks of errors, where

$$m \leq \left\lceil \frac{t - 2}{b - 1} \right\rceil + 1.$$ 

Since each block consists of $p$ bits, so, in particular, a single type $B2$ burst of length

$$b \left\lceil \frac{t - 2}{b - 1} + 1 \right\rceil = b \left\lceil \frac{t + b - 3}{b - 1} \right\rceil$$

is correctable.

$^1 \left[x\right] = \text{greatest integer} \leq x.$
Consider the Situation (B). There are two cases to consider:

(i) There are errors in the first block. By P2, only the first check bit of the \( i \)-th bit can be corrupted by errors in the first block. The number of error blocks allowed is

\[
    u_1 = \left\lceil \frac{t - 1}{b - 1} \right\rceil + 1.
\]

(ii) All errors are in subsequent blocks. Then the number of error blocks allowed is

\[
    u_2 = \left\lfloor \frac{t}{b - 1} \right\rfloor \leq \left\lceil \frac{t - 1}{b - 1} \right\rceil + 1 = u_1.
\]

So we may ignore (i). Thus if the \( i \)-th bit of the first block is not in error, there can be no decoding error if the error pattern consists of \( m \) blocks of errors, where

\[
    m \leq \left\lceil \frac{t}{b - 1} \right\rceil.
\]

To ensure correct decoding, both situations (i) and (ii) must be satisfied. Hence, we have

\[
    m \leq \min \left\{ \left\lceil \frac{t - 2}{b - 1} \right\rceil + 1, \left\lceil \frac{t}{b - 2} \right\rceil \right\}.
\]

Observe that

\[
    \left\lceil \frac{t - 2}{b - 1} \right\rceil + 1 \leq \left\lceil \frac{t}{b - 1} \right\rceil
\]

for \( b = 2, 3 \). Hence we have

**Theorem 2.** Every \( t \)-error correcting R–B type BCSOC of rate \((b - 1)/b\) corrects all error patterns that are confined to \((t - 1)\) blocks, not necessarily consecutive if \( b = 2 \), and \([t/b - 1]\) blocks if \( b \geq 3 \).

**Corollary.** Any \( t \)-error correcting R–B type BCSOC corrects type B2 bursts of length at least \( 2(t - 1) \) bits if \( b = 2 \), and \( b[t/(b - 1)] \) bits if \( b \geq 3 \).

We now consider the burst-error-correcting capacity of a CCSOC derived from R–B type BCSOC. Consider the Situation A. As each subblock may have at most \( k \) errors which may produce a maximum of \( k - 1 \) zero syndromes. Any other errors in the first block would interfere with the first check
symbol. Consequently, a maximum of \((t - 1) - (k - 1) - 1 = t - k - 1\) check symbol changes from errors in the subsequent blocks can be tolerated. Hence, for Case 1, one may have

\[
\left\lfloor \frac{t - k - 1}{b - 1} \right\rfloor
\]

subsequent blocks in error without affecting the correction of errors in the first block.

Next consider the Situation B, with the approach similar to that employed in the BCSOC case. One needs to consider only \((B)\). Here, with no error in the first block, a decoding error is avoided if not more than \(t\) check symbols are nonzero. This implies that one may have \(\lfloor t/(b - 1) \rfloor\) blocks in error. Therefore, to avoid a decoding error, it is sufficient that all the errors are restricted to \(m\) blocks where

\[
m = \min \left\{ \left\lfloor \frac{t - k - 1}{b - 1} \right\rfloor + 1, \left\lfloor \frac{t}{b - 1} \right\rfloor \right\}
\]

\[
= \left\lfloor \frac{t - k - 1}{b - 1} + 1 \right\rfloor \quad \text{if } k + 2 > b
\]

\[
= \left\lfloor \frac{t}{b - 1} \right\rfloor \quad \text{if } k + 2 \leq b.
\]

We state this result formally.

**Theorem 3.** Every \(t\)-character correcting CCSVOC of rate

\[
\frac{k(b - 1)}{k(b - 1) + 1}
\]

derived from a R-B type BCSVOC of rate \(b - 1/b\), corrects all error patterns that are confined to \(\lfloor t/(b - 1) \rfloor\) blocks if \(k + 2 \leq b\) and corrects all error patterns confined to

\[
\left\lfloor \frac{t - k - 1}{b - 1} + 1 \right\rfloor
\]

blocks if \(k + 2 > b\).

**Corollary.** Any \(t\)-character correcting CCSVOC of rate

\[
\frac{k(b - 1)}{k(b - 1) + 1}
\]
derived from an R-B type BCSOC of rate $b - 1/b$ corrects type B2 bursts of length at least

$$(k(b - 1) + 1) \left\lceil \frac{t}{b - 1} \right\rceil$$

if $k + 2 \leq b$ and corrects B2 bursts of length at least

$$(k(b - 1) + 1) \left\lceil \frac{t - k - 1}{b - 1} + 1 \right\rceil$$

if $k + 2 > b$.

**Example.** The best-known rate 1/2 six-error correcting BCSOC (Robinson-Bernstein, 1967) has constraint length of 256 bits and guaranteed burst (B2) correcting capacity of

$$2(t - 1) = 10 \text{ bits.}$$

The corresponding CCSOC with $k = 2$ is a code of rate 2/3, with constraint length 384 characters and corrects bursts (B2) of

$$\left\lceil \frac{6 - 2 - 1}{1} + 1 \right\rceil (2 + 1) = 12 \text{ characters.}$$

This compares well with the best-known (Robinson–Bernstein, 1967) rate 2/3 BCSOC of constraint length 867 bits which also corrects six random errors and with a burst (B2) correcting capability of nine bits.

**VII. Conclusions**

We have presented a new class of character-error correcting convolutional codes.

This class of codes is seen to be a generalization of self-orthogonal convolutional codes and shares many advantages of the self-orthogonal codes. It has the additional advantages of being more efficient and has larger burst-error-correcting ability although decoding is more difficult.

Furthermore, this class of codes is generally easier to decode than the previously known character-error correcting convolutional codes of Ebert and Tong (1969). However, these codes are not as efficient as the Ebert–Tong codes.

Thus, this class of codes seems to be a compromise between the two extremes, viz., the easy-to-decode but inefficient self-orthogonal codes and the hard-to-decode but efficient Ebert–Tong codes.
APPENDIX

Properties of Alternants

Let us denote an alternant by its power indices (leaving out the arguments) as follows:

\[ |A(0, k_1, k_2 \ldots k_{n-1})| = \begin{vmatrix} 1 & X_1^{k_1} & X_1^{k_2} & \cdots & X_1^{k_{n-1}} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & X_n^{k_1} & X_n^{k_2} & \cdots & X_n^{k_{n-1}} \end{vmatrix}. \quad (A1) \]

There are two useful identities (Aitkin, 1956):

\[ |A(0, k_1, k_2, \ldots, k_{n-1})| = |A(0, 1, \ldots, n-1)| \cdot H, \quad (A2) \]

where

\[ H = \begin{vmatrix} h_0 & h_{k_1} & \cdots & h_{k_{n-1}} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & h_{k_1-n+1} & \cdots & h_{k_{n-1}-n+1} \end{vmatrix}. \quad (A3) \]

The \(|A(0, 1, \ldots, n-1)|\) is identified as a \(n \times n\) Vandermonde determinant and \(H\) is called a bialternant. Each \(h_i\) denotes the complete homogeneous symmetric function of degree \(i\) which is the sum of the \(\binom{n+i-1}{i}\) products of the \(x_1 \cdots x_n\) taken \(i\) at a time and with unrestricted repetition of any \(x_i\) in a product. We note that \(h_0 = 1\) and \(h_i = 0, i < 0\).

By the aid of Jacobi's theorem (Aitkin, 1956) on minors of the adjoint, the second identity can be written as

\[ \frac{|A(0, k_1, \ldots, k_{n-1})|}{|A(0, 1, \ldots, n-1)|} = H = \begin{vmatrix} a_{s_1} & a_{s_2} & \cdots & a_{s_r} \\ a_{s_1-1} & \vdots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{s_{r-1}} & \cdots & \cdots & a_{s_{r-1}} \end{vmatrix} = E, \quad (A4) \]

where \(a_i\) is the elementary symmetric function of degree \(i\), which is the sum of the \(\binom{n}{i}\) products of \(x_1, x_2, \ldots, x_n\) taken \(i\) at a time without repetition. We note \(a_i = 0\) if \(i > n\) or \(i < 0\) and \(a_0 = 1\). The set \(\{s_i\}\) is bicomplementary with respect to \(k_{n-1}\) to the set \(\{k_i\}\). That is, the collection defined by the members of the two sets \(\{k_i\} + \{k_{n-1} - s_i\}\) consists of exactly the \(k_{n-1} + 1\) integers: 0, 1, ..., \(k_{n-1}\).

RECEIVED: March 16, 1970; REVISED: August 15, 1970
REFERENCES


