NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE
A CASCADED CODING SCHEME FOR ERROR CONTROL

October 1, 1985

Technical Report

to

NASA
Goddard Space Flight Center
Greenbelt, Maryland

(NASA-CR-176276) A CASCADED CODING SCHEME
FOR ERROR CONTROL (Hawaii Univ., Honolulu.)
44 p HC A03/MF A01

Grant Number NAG 5-407
(Supplement No. 1)

Shu Lin
Principal Investigator
Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, Hawaii 96822
A CASCADED CODING SCHEME FOR ERROR CONTROL

Tadao Kasami
Osaka University
Toyonaka, Osaka, Japan

Shu Lin
University of Hawaii at Manoa
Honolulu, Hawaii 96822

ABSTRACT

In this report, we investigate a cascaded coding scheme for error control. The scheme employs a combination of hard and soft decisions in decoding. Error performance is analyzed. If the inner and outer codes are chosen properly, extremely high reliability can be attained even for a high channel bit-error-rate. Some example schemes are studied. They seem to be quite suitable for satellite down-link error control.
1. Introduction

In this paper we investigate a cascaded coding scheme for error control for a binary symmetric channel with bit-error rate \( \varepsilon < 1/2 \). In this scheme, two linear block codes, \( C_1 \) and \( C_2 \), are used. The inner code \( C_1 \) is a binary \((n_1,k_1)\) code with minimum distance \( d_1 \). The inner code is designed to correct \( t_1 \) or fewer errors and simultaneously detect \( \lambda_1 \) \((\lambda_1 > t_1)\) or fewer errors where \( t_1 + \lambda_1 + 1 \leq d_1 \) [1]. The outer code \( C_2 \) is an \((n_2,k_2)\) code with symbols from the Galois field \( GF(2^k) \) and minimum distance \( d_2 \). If each code symbol of the outer code is represented by a binary \( k \)-tuple based on certain basis of \( GF(2^k) \). Then the outer code becomes an \((n_2^k,k_2^k)\) linear binary code. For the proposed coding scheme, we assumed that the following conditions hold:

\[
\begin{align*}
  k_1 &= m_1^k , \\
  n_2 &= m_1 m_2 .
\end{align*}
\]

The encoding is performed in two steps as shown in Figure 1. First a message of \( k_2^k \) binary information digits is divided into \( k_2 \) bytes of \( k \) information bits each. Each \( k \)-bit byte (or binary \( k \)-tuple) is regarded as a symbol in \( GF(2^k) \). These \( k_2 \) bytes are encoded according to the outer code \( C_2 \) to form an \( n_2 \)-byte \((n_2k)\) bits codeword in \( C_2 \). At the second stage of encoding, the \( n_2 \)-byte codeword at the output of the outer code encoder is divided into \( m_2 \) segments of \( m_1 \) bytes (or \( m_1k \) bits) each. Each \( m_1 \)-byte segment is then encoded according to the inner code \( C_1 \) to form an \( n_1 \)-bit codeword. This \( n_1 \)-bit codeword in \( C_1 \) is called a frame. Thus, corresponding to a message of \( k_2^k \) bits at the input of the outer code encoder, the output of the inner code encoder is a sequence of \( m_2 \) frames of \( n_2 \) bits each. This sequence of \( m_2 \) frames is called a block. A block format is
depicted in Figure 2. We may view that the entire encoding operation is to cascade the two block codes, $C_1$ and $C_2$. The resultant cascaded code, denoted $C$, is a binary $(m_2,n_1,k_2)$ linear code. If $m_1 = 1$, the cascaded code $C$ is a concatenated code [2].

In the proposed scheme, the decoding also consists of two stages as shown in Figure 1. The first stage of decoding is the inner code decoding. Depending on the number of errors in a received frame, the inner code decoder performs one of the three following operations: error-correction, erasure and leave-it-alone (LIA) operations. When a frame in a block is received, its syndrome is computed based on the inner code $C_1$. If the syndrome corresponds to an error pattern $\hat{e}$ of $t_1$ or fewer errors, error correction is performed by adding $\hat{e}$ to the received frame. The $n_1-k_1$ parity bits are removed from the decoded frame, and the decoded $m_1$-byte segment is stored in a receiver buffer for the second stage of decoding. A successfully decoded segment is called a decoded segment with no mark. Note that the decoded segment is error-free, if the number of transmission errors in the received frame is $t_1$ or less. If the number of transmission errors in a received frame is more than $t_1$, the errors may result in a syndrome which corresponds to a correctable error pattern with $t_1$ or fewer errors. In this case, the decoding will be successful, but the decoded frame (or segment) contains undetected errors. If an uncorrectable error pattern is detected in a received frame, the inner code decoder will perform one of the following two operations based on a certain criterion [3]:

1. **Erase Operation** -- The erroneous segment is erased. We will call such a segment an erased segment.

2. **Leave-it-alone (LIA) Operation** -- The erroneous segment is stored in the receiver buffer with a mark. We call such segment a marked segment.
Thus, after $m_2$ frames of a received block have been processed, the receiver buffer may contain three types of segments: decoded segments without marks, erroneous segments with marks, and erased segments.

The above inner code decoding consists of three operations: error-correction, erasure and LIA operations. The decoding operation is described by the flowchart in Figure 3. An inner code decoding which performs only the error-correction and erasure operations is called an erasure-only decoding. On the other hand, an inner code decoding which performs only the error-correction and LIA operations is called a LIA-only decoding.

As soon as $m_2$ frames in a received block have been processed, the second stage of decoding begins and the outer code decoder starts to decode the $m_2$ segments stored in the buffer. Note that an erased segment creates $m_1$ symbol erasures (or $m_1$-bit byte erasures). Symbol errors are contained in the segments with or without marks. The outer code $C_2$ and its decoder are designed to correct the combinations of symbol erasures and symbol errors. Maximum-distance-separable codes with symbols from $GF(2^k)$ are most effective in correcting symbol erasures and errors.

Now we describe outer code decoding process. Let $i$ and $h$ be the numbers of erased segments and marked segments respectively. The outer code decoder declares an erasure (or raises a flag) for the entire block of $m_2$ segments if either of the following two events occurs:

(i) The number $i$ is greater than a certain threshold $T_{es}$ with $T_{es} \leq \lceil (d_2-1)/m_1 \rceil$.

(ii) The number $h$ is greater than a certain threshold $T_{ek}(i)$ with $T_{ek}(i) \leq [(d_2-1-m_1 i)/2]$, for a given $i$.

If none of the above two events occurs, the outer code decoder starts the error-correction operation on the $m_2$ decoded segments. The $m_1 i$ symbol
erasures and the symbol errors in the marked or unmarked segments are corrected based on the outer code \( C_2 \). Let \( t_2(i) \) be the error-correction threshold for a given \( i \) where

\[
t_2(i) \leq \lfloor (d_2 - l - m_1 i) / 2 \rfloor.
\]  
(3)

If the syndrome of the \( m_2 \) decoded segments in the buffer corresponds to an error pattern of \( m_1 i \) erasures and \( t_2(i) \) or fewer symbol errors, error-correction is performed. The values of the erased symbols, and the values and the locations of symbol errors are determined based on a certain algorithm.

If no error correction is made in a marked segment, or more than \( t_2(i) \) symbol errors are detected, then the outer code decoder again declares an erasure (or raises a flag) for the entire block of \( m_2 \) decoded segments. The entire outer code decoding operation is described by the flowchart shown in Figure 4.

In the rest of this paper, the error performance of the proposed cascaded coding scheme is analyzed. We show that, if proper inner and outer codes are chosen, the scheme provides extremely good reliability even for high bit-error-rate \( \varepsilon = 10^{-2} \). The scheme is particularly suitable for down link error control in satellite communications. We also consider interleaving the outer code. The minimum distance of the cascaded code is studied, and a lower bound is derived.

2. The Minimum Weight of a Cascaded Code

Consider the code \( C \) obtained by cascading the inner code \( C_1 \) and the outer code \( C_2 \) as described in Section 1. This cascaded code is an \((m_2 n_1, k_2, c)\) binary linear code. Let \( d \) be its minimum distance. For \( 0 \leq i \leq m_1 \), let \( d_{1,i} \) be the minimum weight of those codewords in \( C_1 \) which have exactly \( i \) nonzero symbols (a symbol is an \( \ell \)-bit byte) in the first \( m_1 \ell \)-bit bytes.

Then we have that
\[ d \geq \min_{0 \leq i_1, i_2, \ldots, i_{m_2} \leq m_1} \left( \sum_{j=1}^{m_2} d_{i_1, i_2, \ldots, i_{m_2}} \right) \quad (4) \]

\[ \sum_{j=1}^{m_2} i_j \geq d_2 \]

It is readily seen that

\[ d \geq \begin{cases} 
   d_1 \left\lceil \frac{d_2}{m_1} \right\rceil, & \text{for } m_1 < d_1 \\
   d_2, & \text{for } m_1 \geq d_1.
\end{cases} \quad (5) \]

\[ d \geq \begin{cases} 
   (d_1/n_1) \left( \left\lceil \frac{n_2 - k_2 + 1}{m_1} \right\rceil / m_2 \right), & \text{for } m_1 < d_1 \\
   (R_1 / \ell)(1 - R/R_1) + 1/n_2, & \text{for } m_1 \geq d_1.
\end{cases} \quad (6) \]

Suppose that the outer code \( C_2 \) is a maximum-distance-separable code over \( \text{GF}(2^\ell) \) [4-8]. Then

\[ d_2 = n_2 - k_2 + 1. \quad (7) \]

Let \( R_1, R_2 \) and \( R \) be the rates of \( C_1, C_2 \) and \( C \) respectively. Then

\[ R = \frac{k_2 \ell}{n_1 m_2} = \frac{k_2 m_1 \ell}{n_1 m_1 m_2} = R_1 R_2. \quad (8) \]

Let \( \delta \) be the ratio of \( d \) to the length \( n_1 m_2 \) of \( C \). It follows from (5) to (7) that

\[ \delta \geq \begin{cases} 
   (d_1/n_1) \left( \left\lceil \frac{n_2 - k_2 + 1}{m_1} \right\rceil / m_2 \right), & \text{for } m_1 < d_1 \\
   (R_1 / \ell)(1 - R/R_1) + 1/n_2, & \text{for } m_1 \geq d_1.
\end{cases} \quad (9) \]

\[ \delta \geq \begin{cases} 
   (d_1/n_1) \left( \left\lceil \frac{n_2 - k_2 + 1}{m_1} \right\rceil / m_2 \right), & \text{for } m_1 < d_1 \\
   (R_1 / \ell)(1 - R/R_1) + 1/n_2, & \text{for } m_1 \geq d_1.
\end{cases} \quad (10) \]

For a nontrivial maximum-distance-separable code with symbols from \( \text{GF}(2^\ell) \), the code length is \( 2^\ell + 2 \) or less. Therefore, for a given \( \ell \), the length of the cascaded code is upper bounded by a constant. Since \( m_1/n_1 = R_1 / \ell \), we see that, if \( d_1/n_1 \) is lower bounded by a positive constant, then the condition

\[ m_1 < d_1 \]

holds for large \( n_2 \). Suppose that \( m_1 < d_1 \) and \( k_2 \) is divisible by \( m_1 \).

It follows from (2) and (9) that
\[ \delta \geq \frac{1}{d_1/n_1} (1-R/R_1 + 1/m_2) . \] (11)

If the inner code meets the Varshamov-Gilbert bound [5-7], then
\[ \delta \geq H^{-1}(1-R_1) \left( 1-R/R_1 + 1/m_2 \right) , \] (12)

where \( H^{-1}(x) \) is the inverse of the binary entropy function \( H(x) = -x \log_2 x - (1-x) \log_2 (1-x) \).

Equation (12) gives a lower bound on the ratio \( \delta \) of the minimum distance to the length of the cascaded code \( C \) with a maximum-distance-separable as the outer code \( C_2 \). This bound is a generalization of Zyablov's bound [9] for concatenated codes,
\[ \delta \geq H^{-1}(1-R_1) \left( 1-R/R_1 + 1/n_2 \right) . \] (13)

Since \( n_2 \geq m_2 \), the bound given by (12) is tighter than that of Zyablov's.

Blokh and Zyablov [10] showed that the general concatenated codes with varying binary linear block inner codes exist which asymptotically meet the Varshamov-Gilbert bound for all rates. Thommesen [11] showed that there exist concatenated codes with varying nonsystematic binary linear block inner codes and Reed-Solomon outer codes which asymptotically meet the Varshamov-Gilbert bound for all rates. A concatenated code with varying binary linear block inner code can be regarded as a cascaded code with \( n_2 = m_1 \) and \( m_2 = 1 \).

It is unknown whether there exist concatenated codes with \( n_2 \geq 2 \) and a single inner code or cascaded codes with \( m_2 \geq 2 \) which asymptotically meet the Varshamov-Gilbert bound.

3. Probabilities of Correct Decoding, Incorrect Decoding and Decoding Failure for a Frame

In this section, we analyze the inner code decoding. We assume that the channel is a binary symmetric channel with bit-error-rate \( \epsilon < 1/2 \). Let \( P_c^{(1)} \) be the probability that a decoded segment is error-free. A decoded segment is error-free if and only if the corresponding received frame contains \( t_1 \) or fewer errors. Thus
Let $P_c^{(1)}$ be the probability of incorrect decoding for a frame. This is actually the probability of an error pattern of $\lambda_1 + 1$ or more errors whose syndrome corresponds to a correctable error pattern of $t_1$ or fewer errors.

Let $P_{es}^{(1)}$ be the probability of a frame erasure, and let $P_{el}^{(1)}$ be the probability that a LIA operation is performed on a frame. Let $P_{er}^{(1)}$ be the probability that a decoded segment with or without a mark contains errors. Then

$$P_c^{(1)} + P_{ic}^{(1)} + P_{es}^{(1)} + P_{el}^{(1)} = 1,$$

and

$$P_{er}^{(1)} = P_{ic}^{(1)} + P_{el}^{(1)}.$$

Note that $P_c^{(1)} + P_{ic}^{(1)}$ is the probability that a received frame is decoded successfully, and $P_{es}^{(1)} + P_{el}^{(1)}$ represents the probability of a decoding failure.

Let $A_i^{(1)}$ and $B_i^{(1)}$ be the numbers of codewords of weight $i$ in the inner code $C_1$ and its dual code $C_1^\perp$ respectively. Let $W_{j,s}^{(i)}(n)$ denote the number binary $n$-tuples with weight $j$ which are at a Hamming distance $s$ from a given binary $n$-tuple with weight $i$. The generating function for $W_{j,s}^{(i)}(n)$ \cite{12} is

$$\sum_{j=0}^{n} \sum_{s=0}^{n} W_{j,s}^{(i)}(n) X^j Y^s = (1+XY)^{n-i}(X+Y)^i.$$  

(17)

It was proved by MacWilliams \cite{12} that

$$P_c^{(1)} + P_{ic}^{(1)} = \sum_{i=0}^{n_1} \sum_{j=0}^{t_1} \sum_{s=0}^{n_1-j} W_{j,s}^{(i)}(n_1) (1-c)^j (1-c)^{n_1-j},$$

(18)

and

$$= 2 \sum_{i=0}^{n_1} B_i^{(1)} (1-2c)^i \sum_{s=0}^{t_1} P_s(i,n_1),$$

(19)
where \( r_1 = n_1 - k_1 \) is the number of parity-check bits of the inner code, and

\[ P_s(\cdot, \cdot) \] is a Krawtchouk polynomial [7, p. 129] whose generating function is

\[
\sum_{s=0}^{n} P_s(i,n)y^s = (1+Y)^{n-i}(1-Y)^i. \tag{20}
\]

Equations (18) and (19) are useful for computing \( P_c^{(1)} + P_{ic}^{(1)} \) if a formula for \( A_i^{(1)} \) or \( B_i^{(1)} \) is known, or \( \min(k_1, r_1) \) is small enough (say less than 25) to be feasible to compute \( A_i^{(1)} \) or \( B_i^{(1)} \) by generating all the codewords in \( C_1 \) or \( C_1^* \).

In order to evaluate the probability \( P_{e_1}^{(1)} \), we need to specify the condition under which the LIA operation is performed. For the LIA-only decoding, the LIA-operation is performed whenever an incorrectable error pattern in the received frame is detected. In this case, the frame erasure probability \( P_{e_1}^{(1)} \) is "zero". For the erasure-only decoding, it is obvious that \( P_e^{(1)} = 0 \). Now we consider the following case. Let \( d_1 = 2t_1 + 2 \). Suppose that \( t_1 \) is odd (or even), and the LIA-operation is performed whenever an incorrectable error pattern with even (or odd) number of errors is detected.

Erasure-operation is performed otherwise. For odd \( t_1 \), we have

\[
P_{e_1}^{(1)} = \sum_{\text{even } j \leq n_1} c^j (1-c)^{n_1-j} \left[ \binom{n_1}{j} - \sum_{i=0}^{n_1} A_i^{(1)} \sum_{s=0}^{t_1} W_j(i) n_1 \right], \tag{21}
\]

\[
= 2^{-1} \{ 1 + (1-2c)^{n_1} \} - 2^{-1} \sum_{i=0}^{n_1} B_i^{(1)} \left[ (1-2c)^i + (1-2c)^{n_1-i} \right] \sum_{s=0}^{t_1} P_s(i, n_1). \tag{22}
\]

(See Appendix A for a derivation of (22).) For even \( t_1 \), we have

\[
P_{e_1}^{(1)} = \sum_{\text{odd } j \leq n_1} c^j (1-c)^{n_1-j} \left[ \binom{n_1}{j} - \sum_{i=0}^{n_1} A_i^{(1)} \sum_{s=0}^{t_1} W_j(i) n_1 \right], \tag{23}
\]

\[
= 2^{-1} \{ 1 - (1-2c)^{n_1} \} - 2^{-1} \sum_{i=0}^{n_1} B_i^{(1)} \left[ (1-2c)^i - (1-2c)^{n_1-i} \right] \sum_{s=0}^{t_1} P_s(i, n_1). \tag{24}
\]

(See Appendix A for a derivation of (24)).
If $P_{e1}$ (or $P_{es}$) is known, then $P_{es}$ (or $P_{e2}$) and $P_{er}$ can be computed from (14) to (16) and (18) (or (19)).

4. **Detail Error Probabilities for a Decoded Segment with no Mark**

For $0 \leq w \leq m_1$, let $P_e^{(1)}$ be the probability that the number of symbol (or byte) errors in a decoded segment **without a mark** is $w$. It is clear that

$$P_e^{(1)} = P_{e,0}^{(1)}$$

and

$$P_{e,w}^{(1)} = \sum_{w=1}^{m_1} P_{e,w}^{(1)}.$$  \hspace{1cm} (25)

To obtain the probability of a correct block decoding, we need to know $P_{e,w}^{(1)}$ for $0 \leq w \leq m_1$. In this section we will derive a formula for $P_{e,w}^{(1)}$.

For a binary $n_1$-tuple $\bar{v}$, we divide the first $k_1 = m_1 l$ bits into $m_1 l$-bit bytes as shown in Figure 5. For $0 \leq h \leq m_1$, let $i_h$ be the weight of the $h$-th $l$-bit byte of $\bar{v}$. Let $i_{m_1+1}$ be the weight of the last $r_1 = n_1 - k_1$ bits. Then the $(m_1+1)$-tuple, $(i_1, i_2, \ldots, i_{m_1+1})$, is called the **weight structure** of $\bar{v}$.

Suppose that a frame $\bar{u}$ is transmitted and an error pattern $\bar{e}$ with weight structure $(j_1, j_2, \ldots, j_{m_1+1})$ occurs. The probability of occurrence of $\bar{e}$ is

$$P(\bar{e}) = (1-c)^{n_1} \prod_{h=1}^{m_1+1} \frac{c}{1-c}^{j_h}.$$  \hspace{1cm} (26)

Suppose that there is a codeword $\bar{v}$ in $C_1$ which is at a distance $t_1$ or less from $\bar{e}$. Since the minimum distance of $C_1$ is assumed to be greater than $2t_1$, such a code $\bar{v}$ in $C_1$ is uniquely determined. Then the inner code decoder assumes that the frame $\bar{u}+\bar{v}$ was sent, and the error pattern $\bar{e}+\bar{v}$ occurred. The decoded segment is the first $k_1$-bits of $\bar{u}+\bar{v}$. If $\bar{v}$ is a **nonzero** codeword, the decoding is incorrect, and the first $k_1$-bits of $\bar{v}$ represent the errors introduced by the inner code decoder. If there is no such codeword $\bar{v}$ in $C_1$, then the inner code decoder performs either the LIA-operation or the erasure-operation. Conversely, for a codeword $\bar{v}$ in $C$ whose weight structure is $(i_1, i_2, \ldots, i_{m_1+1})$, there are...
error patterns $\tilde{e}$'s with weight structure $(j_1,j_2,\ldots,j_{m_1+1})$ such that the weight structure of $v+\tilde{e}$ is $(s_1,s_2,\ldots,s_{m_1+1})$. Let $A_{i_1,i_2,\ldots,i_{m_1+1}}$ be the number of codewords in $C_1$ with weight structure $(i_1,i_2,\ldots,i_{m_1+1})$. For $0 \leq w \leq m_1$, let

$$I_w = \{(i_1,i_2,\ldots,i_{m_1+1}) : 0 \leq i_h \leq l, 0 \leq i_{m_1+1} \leq r_1, \text{and exactly } w \text{ components of } (i_1,i_2,\ldots,i_{m_1+1}) \text{ are nonzero}\}. \quad (28)$$

Then, $P_e,w^{(1)}$ is given below:

$$P_{e,w}^{(1)} = \sum_{(i_1,i_2,\ldots,i_{m_1+1}) \in I_w} A_{i_1,i_2,\ldots,i_{m_1+1}} \prod_{h=1}^{m_1+1} \left[ \begin{array}{c} m_1 \left( i_h \right) \\ \Pi_{j=1}^{h} j_h \end{array} \right] \left( r_1 \right) \left( 1-c \right) \prod_{h=1}^{m_1+1} \left[ \begin{array}{c} m_1+1 \left( j_h \right) \\ \Pi_{j=1}^{h} j_h \end{array} \right] \left( 1-c \right) \prod_{h=1}^{m_1+1} \left[ \begin{array}{c} s_{m_1+1} \left( j_h \right) \\ \Pi_{j=1}^{h} j_h \end{array} \right] \left( 1-c \right) \prod_{h=1}^{m_1+1} \left[ \begin{array}{c} s_{m_1+1} \left( 1-j_h \right) \\ \Pi_{j=1}^{h} j_h \end{array} \right] \left( 1-c \right)$$

where

$$S_{t_1} = \{(s_1,s_2,\ldots,s_{m_1+1}) : 0 \leq s_h \leq l, 0 \leq s_{m_1+1} \leq r_1 \}$$

and

$$s_{m_1+1} = l$$

The formula given by (29) is useful if either (1) the dimension of $C_1$, $k_1$, is small enough (say $k_1 < 25$) to be feasible to compute the detail weight distribution, $A_{i_1,i_2,\ldots,i_{m_1+1}}$, by generating all the codewords in $C_1$, or (2) the dimension of $C_1$, $r_1$, is small enough to be feasible to compute the detail weight distribution of $C_1$ and the number of elements in $I_w$, $I_w^{(1)}$, is small enough to be feasible to enumerate all the elements in $I_w$ and compute $\{A_{i_1,i_2,\ldots,i_{m_1+1}}\}$ by using the generalized MacWilliams' Identity [7].
Next we will express the probability $P_{e,W}$ in terms of the detail weight distribution of the dual code $C_1$ of $C_l$. Let $H$ be a subset of $\{1,2,\ldots,m\}$. Let $P_e^{(1)}(H)$ be the probability that for $h \in H$, the $h$-th $l$-bit byte of a decoded segment is error-free. Let $\bar{H}$ be the complement of $H$ in $\{1,2,\ldots,m\}$.

Define the following set:

$$I(H) = \{(i_1,i_2,\ldots,i_{m+1}) : i_h = 0 \text{ for } h \in H, 0 \leq i_h < l \text{ for } h \not\in H \text{ and } 0 \leq i_{m+1} \leq r_1\}.$$  \hspace{\textwidth}

Then, we have that

$$s^{(1)}_e(H) = \sum_{(i_1,i_2,\ldots,i_{m+1}) \in I(H)} P_{s^{(1)}}(i_1,i_2,\ldots,i_{m+1}) \cdot \prod_{h=1}^{m+1} \binom{m+1}{i_h} i_h^{i_h} (1-i_h)^{m+1-i_h}.$$  \hspace{\textwidth}

Let $B^{(1)}_{i_1,i_2,\ldots,i_{m+1}}$ be the number of codewords in $C_l$ with weight structure $(i_1,i_2,\ldots,i_{m+1})$. Then we have Lemma 1.

**Lemma 1:**

$$s^{(1)}_e(H) = \sum_{h \in H} i_h^{i_h} (1-i_h)^{m+1-i_h} \cdot \prod_{h \in H} \binom{m+1}{i_h} i_h^{i_h} (1-i_h)^{m+1-i_h}.$$  \hspace{\textwidth}

where $|H|$ denotes the number of elements in $H$. 

-11-
Proof: See Appendix B.

For \(0 < s < m_1\), let \(\tilde{U}_S\) be the sum of \(P_e^{(1)}(H)\) where \(H\) is taken over all the subsets of \([1,2,\ldots,m_1]\) with \(s\) elements. Define

\[
U_s(i_1,i_2,\ldots,i_{m_1+1};\epsilon) = \prod_{H \subseteq [1,2,\ldots,m_1]} \frac{\prod_{h \in H} (1-2\epsilon)^{i_h}}{H \in [1,2,\ldots,m_1]} \prod_{h \notin H} (1-\epsilon)^{i_h} \
\cdot \tilde{Q}_t \left( \sum_{h \in H} i_h \right) 
\]

(37)

In the sum \(\tilde{U}_S\), error patterns with \(m_1-s-1\) or less symbol (or byte) errors in a decoded segment are counted more than once. In fact,

\[
\tilde{U}_S = \sum_{s=0}^{m_1} \left( \frac{s+1}{1} \right) P_e^{(1)}(e, m_1-s-1) + \left( \frac{s+2}{2} \right) P_e^{(1)}(e, m_1-s-2) + \cdots + \left( \frac{m_1}{m_1-s} \right) P_e^{(1)}(e,0).
\]

(38)

Using the principle of inclusion and exclusion [13], we have that

\[
P_e^{(1)}(e) = \sum_{h=0}^{j} (-1)^{h} \binom{m_1-h}{h} \tilde{U}_{m_1-J+h}. 
\]

(39)

For \(0 < j < m_1\), define

\[
T_j(i_1,i_2,\ldots,i_{m_1+1};\epsilon) = \sum_{h=0}^{j} (-1)^{h} \binom{m_1-h}{h} \tilde{U}_{m_1-J+h}(i_1,i_2,\ldots,i_{m_1+1};\epsilon) 
\]

(40)

Then it follows from (36) to (40) that we have

Theorem 1:

\[
P_e^{(1)} = \sum_{i_1=0}^{r_1} \sum_{i_2=0}^{r_1} \sum_{i_3=0}^{r_1} \sum_{i_{m_1+1}=0}^{r_1} B_{i_1,i_2,\ldots,i_{m_1+1}}^{(1)} \cdot T_j(i_1,i_2,\ldots,i_{m_1+1};\epsilon)
\]

(41)

It is feasible to obtain the detail weight distribution \(\{B_{i_1,i_2,\ldots,i_{m_1+1}}^{(1)}\}\) by generating all the codewords in \(C_1\) for relatively small \(r_1\), say less than 25. Note that the number of terms to be added in the right-hand side of (37) is \(\binom{m_1}{s}\), and therefore the number of terms to be added or subtracted in the right-hand side of (39) is at most \(2^{m_1}\). For small \(m_1\), \(T_j(i_1,i_2,\ldots,i_{m_1+1};\epsilon)\) can be easily computed and added for each codeword generated. If the
The dual code of $C_1$ of $C$ contains the all-one vector, then $p_{e,j}^{(1)}$ can be computed by generating every codeword in the even-weight subcode and using

$$T_j(i_1,i_2,...,i_{m+1};e) + T_j(l-i_1,l-i_2,...,l-i_{m+1},r_1-i_{m+1};e)$$

instead of $T_j(i_1,i_2,...,i_{m+1};e)$.

For $\lambda = 1$, the outer code is a binary code. In this case, the formula given by (41) is not easy to evaluate since $m_1$ is relatively large. For $\lambda = 1$, let $\lambda_1^{(1)}_{i_1,i_2}$ be the number of codewords in $C_1$ whose weight in the first $k_1$ bits is $i_1$ and weight in the last $r_1$ bits is $i_2$. Then

$$P^{(1)}_{e,i_1} = \sum_{i_2=0}^{r_1} \sum_{j_1=0}^{k_1} \sum_{j_2=0}^{r_1} W(i_1)(j_1)W_2(i_2)(j_2)(r_1)\epsilon^j_1 j_1+ j_2 = (1-\epsilon)^{n_1-j_1-j_2}$$

where

$$S_1' = \{(s_1,s_2) : 0 \leq s_1 \leq k_1, 0 \leq s_2 \leq r_1 \text{ and } 0 \leq s_1 + s_2 \leq t_1 \}.$$ (43)

Let $\bar{B}_{i_1,i_2}^{(1)}$ be the number of codewords in the dual code of $C_1$ whose weight in the first $k_1$ bits is $i_1$ and weight in the last $r_1$ bit is $i_2$. Define

$$Q'_s(i,n,h,m,\gamma) = \sum_{s=0}^{t} Q_s(i,n,h,m,\gamma)\epsilon^s$$ (44)

$$Q'_s(i,n,h,m,\gamma) = \sum_{s=0}^{t} Q'_s(i,n,h,m,\gamma)\epsilon^s$$ (45)

Note that $Q_s(i,n,m,\gamma) = Q'_s(i,n,0,m,\gamma)$. It follows from (17), (20) and (44) that

$$(1+\gamma)^{m-h}(1+Y)^h(1+Y)^{n-i-1}(1-Y)^i = \sum_{s=0}^{n+m} Q'_s(i,n,h,m,\gamma)\epsilon^s$$ (46)

Then we have Theorem 2.

**Theorem 2:** For $\lambda = 1$,

$$P^{(1)}_{e,i_1} = 2^{-r_1(1-\epsilon)} \sum_{h_1=0}^{r_1} \bar{B}_{h_1,h_2}^{(1)}(1-2\epsilon)^h_2 p_{i_1}(h_1,k_1)Q'_s(h_2,r_1,i_1,k_1,\epsilon/1-\epsilon).$$ (47)
Proof: See Appendix C.

For $k_1 > r_1$, it is more convenient to use (47) than (42) to evaluate $P_{e, i_1}^{(1)}$.

5. Detail Error Probability for a Marked Segment

In this section we will evaluate the probability of symbol errors in a marked segment. Let $P_{e, w}^{(1)}$ be the probability that the number of erroneous symbols in a marked segment is $w$. Then

$$P_{e, w}^{(1)} = \sum_{w=1}^{m_1} P_{e, w}^{(1)}$$

(48)

We first consider the LIA-only decoding. Define

$$J_w = \{(j_1, j_2, \ldots, j_{m_1+1}) : 0 < j_h < \ell \text{ for } 0 \leq h \leq m_1, 0 \leq j_{m_1+1} < r_1, \text{ and there are exactly } w \text{ nonzero components in } (j_1, j_2, \ldots, j_{m+1})\}$$

(49)

Then it follows from the definition of $P_{e, w}^{(1)}$ that

$$P_{e, w}^{(1)} = \binom{m_1}{w} [1 - (1 - \varepsilon)^{2\ell}]^w (1 - \varepsilon)^w - \sum_{i_1=0}^{\ell} \sum_{i_m=0}^{\ell} \sum_{i_{m+1}=0}^{r_1} A_i^{(1)} \sum_{j_1, j_2, \ldots, j_{m+1}} \left[ \sum_{j_h=1}^{m_1} w_j^{(1)} \sum_{s_h=1}^{s_h} \sum_{i_{m+1}=0}^{r_1} \sum_{j_{m+1}=0}^{j_{m+1}} S_{j_1, j_2, \ldots, j_{m+1}} \right]$$

(50)

where $S_{j_1, j_2, \ldots, j_{m+1}}$ is defined by (30). The first term of (50) represents the probability that there are exactly $w$ erroneous symbols in the first $m_1$ bytes of a received frame, and the second term is the probability that the syndrome of these symbol errors corresponds to an error pattern of $t_1$ or fewer errors.
Define

\[ R_w(i_1,i_2,...,i_m;\varepsilon) = \sum_{H \subseteq \{1,2,...,m\}} \prod_{h \in H} \left( (1-2\varepsilon)^{i_h} - (1-\varepsilon)^{i_h} \right), \quad (51) \]

where the summation is taken over all the subsets of \(\{1,2,...,m\}\) with exactly \(w\) elements. Then \(P_{e,k,w}^{(l)}\) can be expressed in terms of the detail weight distribution of the dual code of \(C_1\).

**Theorem 3:**

\[ P_{e,k,w}^{(l)} = (1-\varepsilon)^{-k_w} \left( \binom{m}{w} - (1-\varepsilon)^{i_{m+1}} \right) \sum_{i_1=0}^{r_1} \sum_{i_m=0}^{r_m} \sum_{i_{m+1}=0}^{r_{m+1}} B_{i_1,i_2,...,i_{m+1};\varepsilon}^{(l)} \cdot P_t \left( \sum_{h=1}^{m+1} i_h - (m+1-1)R_w(i_1,i_2,...,i_m;\varepsilon) \right). \]

**Proof:** See Appendix D. \(\square\)

For \(k=1\), \(R_w(i_1,i_2,...,i_m;\varepsilon)\) can be simplified as follows. Let \(i\) denote \(\sum_{h=1}^{m+1} i_h\). Since \(0 < i_h < 1\) for \(1 < h \leq m\),

\[ (1-2\varepsilon)^{i_h} - (1-\varepsilon)^{i_h} = (-1)^{i_h} \varepsilon. \]

Consequently, we have that

\[ R_w(i_1,i_2,...,i_m;\varepsilon) = \varepsilon^w \sum_{h=0}^{m} (-1)^h \binom{k-i}{h} \binom{w-h}{w-h}. \]

Using the definition of Krawtchouk polynomial [7, p. 151], we have that

\[ R_w(i_1,i_2,...,i_m;\varepsilon) = \varepsilon^w p_w(i,k). \]

Define

\[ I_j = \{(i_1,i_2,...,i_m) : 0 < i_h < 1 \text{ for } 1 \leq h \leq m_1 = k_1 \text{ and } \sum_{h=1}^{m_1} i_h = j \}. \]

\(\square\)
Then
\[
B_{j_1, j_2}^{(1)} = \sum_{i_{j_1, j_2}} B_{i_{j_1, j_2}}^{(1)}.
\] (56)

It follows from (52), (54) and (56) that we have Corollary 4 [see Appendix E].

**Corollary 4:** For \( \ell = 1 \),
\[
p^{(1)}_{e\ell, w} = \epsilon^w (1-\epsilon)^{-r_1} \left\{ \frac{k_1}{w} \right\} - 2 \sum_{i_1=0}^{r_1} \sum_{i_2=0}^{r_1} B_{i_1, i_2}^{(1)} \left(1-2\epsilon\right)^{i_2} i_2 \cdot P_t (i_1+i_2-1, n_1-1) P_w (i_1, k_1).
\] (57)

Now we consider the decoding in which both LIA and erasure operations are performed. Suppose that the LIA-operation is performed whenever an incorrectable error pattern with even (or odd) weight is detected. In a similar way to that for deriving (22), formula (54) and (57) can be modified. For \( \ell = 1 \),
\[
p^{(1)}_{e\ell, w} = \epsilon^w (1-\epsilon)^{-r_1} \left\{ \frac{k_1}{w} \right\} - 2 \sum_{i_1=0}^{r_1} \sum_{i_2=0}^{r_1} B_{i_1, i_2}^{(1)} \left(1-2\epsilon\right)^{i_2} i_2 \cdot P_t (i_1+i_2-1, n_1-1) P_w (i_1, k_1).
\] (58)

where + (or -) is taken for even \( w \), and - (or +) is taken for odd \( w \).

An important question is which provides better performance, "the LIA-only decoding," or "the erasure-only decoding." LIA-only operation may be reasonable only if
\[
\sum_{w=m_1/2+1}^{m_1} p^{(1)}_{e\ell, w} < p^{(1)}_{e\ell}.
\] (59)

If
\[
\sum_{w=m_1/2+1}^{m_1} p^{(1)}_{e\ell, w} < 1 - p^{(1)}_c + p^{(1)}_{ic}
\] (60)
where \( P_{e(1)(1)}^{(1)w} \) is computed under the assumption that the inner code decoding is a LIA-only decoding, then a LIA-only decoding provides better performance than the erasure-only decoding.

6. The Probability of a Correct Block Decoding

In this section, we will evaluate the probability that a block of \( m \) segments will be decoded correctly by the outer code decoder. Let \( P_e(j,i,h) \) denote the probability that there are \( h \) segments with marks and \( j \) symbol errors in a set consisting of \( i \) decoded segments without marks and \( h \) segments with marks. It follows from the definition of \( P_e(j,i,h) \) that

\[
P_e(j,0,1) = P_e(j,0,1) = 0, \quad \text{for } j > m_1,
\]

and

\[
P_e(i,j,h) = \sum_{w=0}^{\min(j,m_1)} P_e(j-w,i-1,h)P_e^{(1)}_{w} + P_e(j-w,i,h-1)P_e^{(1)}_{w} .
\]

From (61) to (64), \( P_e(j,i,h) \) can be computed readily.

The probability that, after the inner code decoding of a block of \( m_2 \) frames, there exist \( i \) erased segments, \( h \) marked segments, and \( j \) symbol errors in the marked and unmarked (or decoded) segments is

\[
C_{m_2} \left[ \sum_{i=0}^{C_{m_2}^{(1)}} P_e(j,m_2-i-h,h) \right] .
\]

Therefore, the probability of correct decoding of a block denoted \( P_c \) is given by

\[
P_c = \sum_{i=0}^{C_{m_2}^{(1)}} \left[ \sum_{h=0}^{C_{m_2}^{(1)}} P_e(j,m_2-i-h,h) \right] .
\]

Let \( P_{es} \) and \( P_{er} \) denote the probabilities of a block erasure and an incorrect decoding respectively. Then
It follows from definitions that the following equality and bounds hold:

\[
P_{es} + P_{es} + P_{er} = 1
\]  

(67)

\[
P_{es} + P_{er} = \sum_{i=0}^{t_{2}(i)} \left( \sum_{h=0}^{m_{2} - m_{1}^i} P_{e}(j, m_{2} - i - h, h) \right) + \sum_{i=0}^{t_{2}(i)+1} \left( \sum_{h=m_{2} - m_{1}^i}^{m_{2} - m_{1}^i + t_{2}(i) - 1} P_{e}(j, m_{2} - i - h, h) \right)
\]

(68)

\[
P_{er} < \sum_{i=0}^{T_{es}} \left( \sum_{h=0}^{m_{2} - m_{1}^i} P_{e}(j, m_{2} - i - h, h) \right) + \sum_{i=T_{es}+1}^{m_{2} - 1} \left( \sum_{h=m_{2} - m_{1}^i}^{m_{2} - m_{1}^i + t_{2}(i) - 1} P_{e}(j, m_{2} - i - h, h) \right)
\]

(69)

\[
P_{es} > \sum_{i=T_{es}+1}^{m_{2}} \left( \sum_{h=0}^{m_{2} - m_{1}^i} P_{e}(j, m_{2} - i - h, h) \right)
\]

(70)

where

\[
d_{2} - m_{1}^i - t_{2}(i) - 1 \leq \sum_{j=t_{2}(i)+1}^{d_{2} - m_{1} - t_{2}(i)} P_{e}(j, m_{2} - i - h, h) = 0
\]

if \(d_{2} - m_{1}^i - t_{2}(i) - 1 = 2t_{2}(i)\).

If every error pattern of symbol-weight equal to or greater than \(d_{2} - m_{1}^i - t_{2}(i)\) causes an incorrect block decoding, then the equality holds in (69). We consider the number of those error patterns of the smallest symbol-weight \(w = d_{2} - m_{1}^i - t_{2}(i)\) which lead to an incorrect decoding. Suppose that \(C_2\) is a maximum-distance-separable code over \(\mathbb{G}(2^k)\). Let \(L\) be a set of \(w\) symbol positions outside the erased segments such that every marked segment has a symbol
position in $L$. The number of codewords in $C_2$ of weight $j \geq d_2$ whose nonzero positions are specified is [6, p. 71]

$$\sum_{h=0}^{j-d_2} (-1)^h \binom{j}{h} (2^j - 1).$$

Let $E(L)$ be the set of vectors of symbol-weight $w$ which satisfies the following conditions: (1) $L$ is the set of nonzero symbol positions of each vector, and (2) there exists a codeword in $C_2$ which is at a distance (outside the erased segments) $t_2(i)$ or less from each vector. If such a codeword exists, then the codeword is unique, has weight $d_2$ and has a nonzero symbol at every symbol position in either $L$ or an erased segment. The number of such codewords in $C_2$ is

$$\binom{n_2 - m_1 - w}{t_2(i)} (2^j - 1).$$

Therefore the number of error patterns in $E(L)$ is

$$|E(L)| = \binom{n_2 - m_1 - w}{t_2(i)} (2^j - 1) < (2^j - 1)^{t_2(i)} / t_2(i)!$$

The ratio of $|E(L)|$ to the number of error patterns whose set of nonzero symbol positions is $L$ is

$$\binom{n_2 - m_1 - w}{t_2(i)} (2^j - 1)^{m_1} \frac{m_1 + 1 - d_2}{t_2(i)!} < (2^j - 1)^{2t_2(i)} / t_2(i)!$$

If any nonzero symbol error occurs with the same probability and $P_e(w, m_2 - i - h, h)$ is dominant in the summation of (69), then $P_{er}$ is nearly equal to

$$\frac{1}{2^j - 1} / t_2(i)! \times \text{of the right-hand side of (69)}. \text{ On the other hand, if a symbol error with a small bit-weight is more likely than the symbol errors with a larger bit-weight, then the right-hand side of (69) might be a tight bound.}$$
No feasible procedure for computing \(P_{es}\) or \(P_{er}\) has been devised except for small \(k_2\) or \((n_2-k_2)\lambda\). The following simple bounds on \(P_{es} + P_{er}\) and \(P_{es}\) are useful for small bit-error rate \(\epsilon\). We will consider an erasure-only decoding.

If there are symbol errors in a set of decoded segments, then there are at least \([s/m_1]\) segments containing error symbols. Hence

\[
\sum_{j=2}^{n_2-m_1} \frac{P_{e}(j, m_2-i, 0)}{(s/m_1)} < \left(\frac{m_2-i}{s/m_1}\right)
\]

It follows from (68), (69) and (74) that

\[
P_{es} + P_{er} \leq \sum_{i=0}^{T_{es}} \left(\begin{array}{c} m_2 \\ i \end{array}\right) [P_{es}]^{i} [P_{er}]^{1-i} f_0(i)
\]

\[
+ \sum_{i=T_{es}+1}^{m_2} \left(\begin{array}{c} m_2 \\ i \end{array}\right) [P_{es}]^{i} [1-P_{es}]^{1-i} f_1(i)
\]

\[
P_{er} \leq \sum_{i=0}^{T_{es}} \left(\begin{array}{c} m_2 \\ i \end{array}\right) [P_{es}]^{i} [P_{er}]^{1-i} f_0(i)
\]

where

\[
f_0(i) = \left\lceil \frac{(t_2(i)+1)/m_1} \right\rceil \quad \text{and} \quad f_1(i) = \left\lceil \frac{(d_2-m_1-t_2(i))/m_1} \right\rceil \]

Suppose that \(d_1 > 2t_1+1\). In the right-hand sides of (72) (73), the product,

\[
[P_{es}]^{i} [P_{er}]^{1-i} f_0(i)
\]

for \(\alpha = 0\) or \(1\), is upper bounded by

\[
max x \left(1-P(1) - p_2(x) \right) f_0(i)
\]

under the constraint,

\[
\frac{d_1-t_1-1}{i} \left(\begin{array}{c} n_1 \\ i \end{array}\right) \leq x \leq 1 - P(1)_c
\]

since

\[
P_{es} \geq \sum_{i=t_1+1}^{d_1-t_1-1} \left(\begin{array}{c} n_1 \\ i \end{array}\right) (1-P(1)_c)^{n_1-i}
\]

-20-
and \( P_{\text{es}}^{(1)} + P_{\text{er}}^{(1)} = 1 - P_{\text{c}}^{(1)} \). Let \( LH \) denote the left-hand side of (78). Then the maximum of (77) occurs at \( x = LH \) for \( i(1-P_{\text{c}}^{(1)})/(i+f_a(i)) \leq LH \), and

\[ x = i(1-P_{\text{c}}^{(1)})/(i+f_a(i)) \]

otherwise. Similarly, in the second summation of (72), \( P_{\text{es}}^{(1)} \) is upperbounded by \( 1-P_{\text{c}}^{(1)} \) if \( 1-P_{\text{c}}^{(1)} < i/m_2 \), otherwise \( P_{\text{es}}^{(1)} \) is upperbounded by \( i/m_2 \). The bounds derived from (75) and (76) in this way are weak for large \( \varepsilon \), however they are useful for a quick estimation of the system reliability because they do not depend on the detail weight structure of the inner and outer codes, \( C_1 \) and \( C_2 \).

7. Interleaving

In this section, we investigate how interleaving affects the error performance of the cascaded scheme. Suppose that the outer code is interleaved in such a way that each symbol (or \( l \)-bit byte) in a segment is from a different outer code codeword as shown in Figure 6. Thus, the interleaving depth (or degree) is \( m_1 \). The code array consists of \( n_2 \) frames and is transmitted column by column. As for the decoding, after \( n_2 \) received frames have been decoded, the \( n_2 \) decoded segments are arranged into an array as shown in Figure 7. Then each row is decoded based on the outer code \( C_2 \). Note that buffers are needed to store code arrays at both transmitter and receiver.

For \( 1 \leq u < m_1 \), let \( P_{\text{e}}(u) \) be the probability that the \( u \)-th symbol of a decoded segment with no mark is erroneous. If the inner code \( C_1 \) is quasi-cyclic by every \( s \)-bit shift where \( s \) divides \( l \), then \( P_{\text{e}}(u) \) is independent of \( u \).

It follows from the definition that

\[ P_{\text{e}}(u) = P_{\text{c}}^{(1)} + P_{\text{iC}}^{(1)} - P_{\text{e}}^{(1)}(\{u\}) \qquad (79) \]

where \( P_{\text{e}}^{(1)}(\{u\}) \) is given by (31) or (35). Hence \( P_{\text{e}}(u) \) can be computed from either (18) and (31) or (19) and (35).

Let \( P_{\text{eL}}(u) \) be the probability that the \( u \)-th symbol of a marked segment is erroneous. For simplicity, the LIA-only decoding is considered. Define
J(u) = \{(j_1,j_2,\ldots,j_{m+1}) : 0 \leq j_h \leq \ell \text{ for } 1 \leq h \leq m_1, j_u \neq 0 \text{ and } 0 \leq j_{m+1} \leq r_1\}.

Modifying the derivation of (50) or (52), we have that

\[
P_{e_k}(u) = 1 - (1-c)^\ell - \sum_{i_1=0}^{r_1} \sum_{i=0}^{m_1} \sum_{i_{m+1}=0}^{\ell-j} A_{i_1,i_2,\ldots,i_{m+1}} J(u) S_{t_1}^{(1)} \prod_{h=1}^{m_1} W_{e_k}^{(1)} \left(\frac{j_{m+1}}{j_{m+1} + 1} (1-c)^{j_{m+1} + 1 - j} \right) \cdot \prod_{h=1}^{m_1} (1-c)^{j_{m+1}} \left(1 - (1-c)^{j_{m+1}}\right) P_{e_k}^{(1)} \left(\frac{j_{m+1}}{j_{m+1} + 1} k - 1, n - 1\right).
\]

and

\[
P_{e_k}(u) = 1 - (1-c)^\ell - 2 \sum_{i_1=0}^{r_1} \sum_{i=0}^{m_1} \sum_{i_{m+1}=0}^{\ell-j} B_{i_1,i_2,\ldots,i_{m+1}} J(u) S_{t_1}^{(1)} \prod_{h=1}^{m_1} W_{e_k}^{(1)} \left(\frac{j_{m+1}}{j_{m+1} + 1} (1-c)^{j_{m+1} + 1 - j} \right) \cdot \prod_{h=1}^{m_1} (1-c)^{j_{m+1}} \left(1 - (1-c)^{j_{m+1}}\right) P_{e_k}^{(1)} \left(\frac{j_{m+1}}{j_{m+1} + 1} k - 1, n - 1\right).
\]

[See Appendix F for the derivation of (81)].

Since the outer code is interleaved by a depth of \(m_1\), the \(u\)-th symbol of every segment is from the \(u\)-th outer code codeword for \(1 \leq u \leq m_1\). Let \(P_{c}(u)\), \(P_{es}(u)\) and \(P_{er}(u)\) denote the probabilities of a correct decoding, an erasure and an incorrect decoding for the \(u\)-th outer code codeword respectively. Then formulas or bounds for \(P_{c}(u)\), \(P_{es}(u)\) and \(P_{er}(u)\) can be derived from those for \(P_{c}\), \(P_{es}\) or \(P_{er}\) by the following replacements: \(m_1 \rightarrow i, m_2 \rightarrow n_2\) and

\[
\sum_{h} \sum_{j} P_{e}(j,m_2-i-h,h) + \sum_{h} \left(\sum_{j} P_{e}(j,m_2-i-h,h)\right) \sum_{j} \left(\sum_{s=0}^{j} P_{e}(j,m_2-i-h,h)\right) [P_{e_k}^{(1)}(u)]^s
\]

\[
\cdot \left[1-P_{es}(u) - P_{e_k}(u) - P_{e_k}(u)\right]^{n_2-i-h-s} [P_{e_k}^{(1)}(u)]^{j-s} [P_{e_k}^{(1)}(u)]^{h-j+s}.
\]

The restrictions on thresholds, \(T_{es}, T_{e_k}(i)\) and \(t_2(i)\) can be relaxed as follows:

\[T_{es} \leq d_2 - 1, \quad T_{e_k}(i) \leq (d_2-1-i)/2, \quad t_2(i) \leq (d_2-1-i).\]
8. Example Schemes

In the following we consider two example schemes using cascaded coding for error control. In the first example scheme, the inner code is a triple-error-correcting and quadruple-error detecting (59,40) code which is obtained by deleting 4 information bits from the distance-8 \((63,44)\) BCH code. The generator polynomial of this code is

\[
g_1(X) = (1+X)(1+X+X^6)(1+X+X^2+X^4+X^6)(1+X^2+X^5+X^6)\]

Since the code contains only even-weight codewords, it is capable of detecting all the error patterns of weight 4 and all the error patterns of odd weight greater than 4. Moreover, the code is majority-logic decodable in two steps [1], and hence the decoder can be easily implemented. The outer code is the \((255,223)\) Reed-Solomon (RS) code with symbols from \( \text{GF}(2^8) \) and minimum distance \( d_2 = 33 \). This outer code is capable of correcting any combination of \( i \) symbol erasures and \( t_2(i) \) symbol errors with \( i + 2t_2(i) < 33 \). For the first example scheme, the important parameters are: \( n_2 = 255 \), \( k_2 = 223 \), \( n_1 = 59 \), \( k_1 = 40 \), \( \lambda = 8 \), \( m_1 = 5 \), \( m_2 = 51 \), \( t_1 = 3 \) and \( d_2 = 33 \). Suppose that the erasure-only decoding is adopted. Then, \( T_{es} = 6 \) and \( t_2(i) = \left[\frac{(32-5i)}{2}\right] \). The error performance of this example scheme for bit-error-rate \( \epsilon = 10^{-2} \) and \( 10^{-3} \) is given in Table 1. The bounds on \( P_{es+Per} \) and \( P_{er} \) are computed based on the weak bounds given by Eq. (75) and Eq. (76). Even from these weak bounds, we see that this scheme provides extremely high reliability. Tighter bounds on error performance based on (68) and (69) are being computed for inner code decoding with all three operations. Computation results will be tabulated in our next report. We believe that high reliability can be achieved by using a less powerful RS code of length 255 as the outer code. We are also computing the error performance of the scheme using interleaving.
For the second example scheme, the inner code is a double-error-correcting and triple-error-detecting (53,40) code which is obtained from the distance-6 (63,50) BCH code by deleting 10 information bits. Besides detecting all triple errors, the code is also capable of detecting all error patterns of odd weight greater than 3. The generator of the code is \([1], g(X) = (1+X)(1+X+X^6)(1+X+X^2+X^4+X^6)\).

The outer code is the same as the one used in the first example scheme. The error performance of this second example scheme is still being evaluated. However, if we use erasure-only decoding with \(T_{es} = 3\), \(t_2(0) = t_2(1) = t_2(2) = t_2(3) = 0\), then for bit-error-rate \(\epsilon = 10^{-2}\), the block error probability \(P_{er}\) is upper bounded by \(2.13 \times 10^{-12}\).

<table>
<thead>
<tr>
<th></th>
<th>(\epsilon = 10^{-2})</th>
<th>(\epsilon = 10^{-3})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p(e)_{es})</td>
<td>0.289 \times 10^{-2}</td>
<td>0.4348 \times 10^{-6}</td>
</tr>
<tr>
<td>(p(e)_{er})</td>
<td>0.4491 \times 10^{-4}</td>
<td>0.7246 \times 10^{-9}</td>
</tr>
<tr>
<td>(P_{es})</td>
<td>\leq 0.265 \times 10^{-8}</td>
<td>\leq 0.1664 \times 10^{-27}</td>
</tr>
<tr>
<td>(P_{er})</td>
<td>\leq 0.2183 \times 10^{-8}</td>
<td>\leq 0.1664 \times 10^{-27}</td>
</tr>
</tbody>
</table>

9. Conclusion

In this report, we investigated a cascaded coding scheme for error control. The scheme employs a combination of hard and soft decisions in decoding. Error performance is analyzed. If the inner and outer codes are chosen properly, extremely high reliability can be achieved even for a high channel bit-error-rate. Two example schemes are being studied. Both use shortened BCH codes as the inner codes. One code has a rate of 2/3, and is...
majority-logic decodable. Hence the decoding can be implemented easily. The other code has a rate of about 4/5; and since it has only 13 parity-check bits, it can be decoded with a table-look-up decoding. Based on our preliminary computation results, both schemes provide high reliability even for a high bit-error-rate, say $\epsilon = 10^{-2}$. They seem to be quite suitable for satellite down-link error control. Since the inner codes have rates greater than 1/2, the two example schemes definitely have advantage in bandwidth over the usual concatenated coding scheme using a rate 1/2 convolutional code as the inner code and a RS code as the outer code. Further evaluation of these two example schemes will be reported in our next technical report to NASA.
REFERENCES


APPENDIX A

Derivation of Expression (22) and (24)

It follows from (17) and MacWilliams' identity [11] that

\[
\sum_{i=0}^{n_1} A_1^{(1)} \sum_{j=0}^{n_1} \sum_{s=0}^{n_1} W(i)_j s (n_1)_j Y_s j = \sum_{i=0}^{n_1} A_1^{(1)} (1+XY)^i (X+Y)^i
\]

\[
= 2^{-r_1} \sum_{i=0}^{n_1} B_1(i) (1+X)^{n_1-i} (X+Y)^{n_1-i} \cdot \frac{n_1}{i+1}.
\]

Therefore, we have that

\[
\sum_{i=0}^{n_1 \text{ even}} A_1^{(1)} \sum_{j=0}^{n_1 \text{ even/s odd}} W(i)_j s (n_1)_j X_j Y_s
\]

\[
= 2^{-r_1} \sum_{i=0}^{n_1} B_1(i) (1+X)^{n_1-i} (1-X)^{n_1-i} (1+Y)^{n_1-i} (1-Y)^{n_1-i}.
\]

(A-1)

where the "+" and "-" signs of the second term in the bracket for even and odd \( j \) respectively. It follows from (20) and (A-2) that

\[
\sum_{i=0}^{n_1} A_1^{(1)} \sum_{j=0}^{t_1} W(i)_j s X_j Y_s
\]

\[
= 2^{-r_1} n_1 t_1 \sum_{i=0}^{n_1} B_1(i) (1+X)^{n_1-i} (1-X)^{n_1-i} (1+Y)^{n_1-i} (1-Y)^{n_1-i}.
\]

(A-2)

Substituting \( \frac{c}{1-c} \) for \( X \) and \( 1 \) for \( Y \) and multiplying both sides of (A-3) by \( (1-\varepsilon)^{n_1} \), we obtain the second term of (22) for even \( j \) and the second term of (24) for odd \( j \).
APPENDIX B

Proof of Lemma 1

Let $|H| = u$. It follows from (17) that

$$\sum_{(i_1,i_2,\ldots,i_{m_1+1}) \in I(H)} A^{(1)}(i_1,i_2,\ldots,i_{m_1+1}) \prod_{h=1}^{m_1+1} w_h \prod_{h=0}^{j_h} s_{h,h} = \sum_{i=0}^{n_1-u-i} B_i^{(1)}(H) \prod_{h=1}^{m_1+1} x_h \prod_{h=0}^{j_h} s_{h,h}$$

The set of codewords in $C_1$ whose weight in the $h$-th $k$-bit byte is zero for every $h$ in $H$ is a linear $(n_1,k_1-ku)$ subcode of $C_1$. Let $C_1(H)$ denote the linear $(n_1-ku,k_1-ku)$ code obtained from the above subcode by deleting the $u$ zero $k$-bit bytes for the $u$ positions in $H$. Let $A_i^{(1)}(H)$ denote the number of codewords of weight $i$ in $C_1(H)$. Then

$$A_i^{(1)}(H) = \sum_{(i_1,i_2,\ldots,i_{m_1+1}) \in I(H)} A^{(1)}(i_1,i_2,\ldots,i_{m_1+1}) \prod_{h=1}^{m_1+1} w_h \prod_{h=0}^{j_h} s_{h,h}$$

The right-hand side of (B-1) can be rewritten as

$$B_i^{(1)}(H) = \sum_{i=0}^{n_1-ku} B_i^{(1)}(H) \prod_{h=1}^{m_1+1} x_h \prod_{h=0}^{j_h} s_{h,h}$$

Let $B_i^{(1)}(H)$ be the number of codewords of weight $i$ in the dual code of $C_1(H)$.

Then, by MacWilliams' identity [7], (B-3) can be written as

$$B_i^{(1)}(H) = \sum_{i=0}^{n_1-ku} A_i^{(1)}(H) \prod_{h=1}^{m_1+1} x_h \prod_{h=0}^{j_h} s_{h,h}$$
It follows from (35), (B-1) and (B-4) that

\[ \sum_{(i_1, i_2, \ldots, i_{m+1}) \in I(H)} \left[ \prod_{h=1}^{m_1} \sum_{j_h=0}^{r_1} \sum_{s_h=0}^{r_1} n_{j_h s_h} \right] \left( \prod_{h=1}^{m_1} \sum_{j_h=0}^{r_1} \sum_{s_h=0}^{r_1} n_{j_h s_h} \right) \]

\[ = \frac{r_1}{2} \left( \prod_{i=0}^{m_1} B_i^{(1)}(H)(1+X) \right)^{n_1 - \lambda u} \frac{1}{(1-X)^{i}(1+Y)^{i}} \frac{1}{(1-Y)^{i}}. \] 

(B-4)

Taking the terms on both sides of (B-5) for which the degree of Y is \( t_1 \) or less and substituting "1" for Y, we have that

\[ \sum_{(i_1, i_2, \ldots, i_{m+1}) \in I(H)} \left[ \prod_{h=1}^{m_1} \sum_{j_h=0}^{r_1} \sum_{s_h=0}^{r_1} n_{j_h s_h} \right] \left( \prod_{h=1}^{m_1} \sum_{j_h=0}^{r_1} \sum_{s_h=0}^{r_1} n_{j_h s_h} \right) \]

\[ = 2 \frac{r_1}{2} \left( \prod_{i=0}^{m_1} B_i^{(1)}(H)(1+X) \right)^{n_1 - \lambda u} \frac{1}{(1-X)^{i}(1+Y)^{i}} \frac{1}{(1-Y)^{i}}. \] 

(B-5)

Substituting \( \varepsilon/(1-\varepsilon) \) for X and multiplying the left-hand side of (B-6) by \( n_1 \), we obtain the right-hand side of (32). Therefore we have that

\[ P_e^{(1)}(H) = 2 \frac{r_1}{2} \left( \prod_{i=0}^{m_1} B_i^{(1)}(H)(1+X) \right)^{n_1 - \lambda u} \frac{1}{(1-X)^{i}(1+Y)^{i}} \frac{1}{(1-Y)^{i}}. \] 

(B-7)

Since a generator matrix of the dual code of \( C_1(H) \) can be obtained from a parity-check matrix of \( C_1 \) by deleting all columns corresponding to the h-th \( \lambda \)-bit positions for \( h \in H \), the following relation holds.

\[ B_i^{(1)}(H) = \prod_{i=0}^{m_1} B_i^{(1)}(H)(1+X)^i(1+Y)^i(1-X)^i. \] 

(B-8)
where

\[ I_1(H) = \{(i_1, i_2, \ldots, i_{m+1}) : 0 \leq i_h \leq l \text{ for } 1 \leq h \leq m, \ 0 \leq i_{m+1} \leq r_1, \ \text{ and } \sum_{h \in H} i_h = i\}. \]

Then, expression (36) of Lemma 1 follows from (B-7) and (B-8).
APPENDIX C

Proof of Theorem 2

It follows from (17) that

\[
\sum_{i_1, i_2} \left[ \sum_{j_1} W_{j_1} (k_1 X Y)^{j_1 s_1} \right] \sum_{j_2} W_{j_2} (r_1 X Y)^{j_2 s_2} = \left( 1 + XY \right)^{r_1 - 1} \left( 1 + X + Y \right) = \left( 1 + X \right)^{r_1} \left( 1 - X \right)^{r_2} \left( 1 + Y \right)^{r_1} \left( 1 - Y \right)^{r_2}.
\]  

(C-1)

By the generalized MacWilliams' identity [7, p. 147], we have

\[
\sum_{i_1, i_2} A_{i_1, i_2}^{(l)} = 2^{r_1} \sum_{h_1=0}^{r_1} \sum_{h_2=0}^{r_1} B_{h_1, h_2}^{(l)} P_1 (h_1, k_1) P_i (h_2, r_1).
\]  

(C-2)

It follows from (20) that

\[
\sum_{i_2} p_{i_2} (h_2, r_1) \left( 1 + XY \right) = (1 + X)^{r_1 - h_2} (1 - X)^{h_2} (1 + Y)^{r_1} (1 - Y)^{r_2}.
\]  

(C-3)

It follows from (C-1) to (C-3) and (46) that

\[
\sum_{i_2} \left[ \sum_{j_1} W_{j_1} (k_1 X Y)^{j_1 s_1} \right] \sum_{j_2} W_{j_2} (r_1 X Y)^{j_2 s_2} = \left( 1 + XY \right)^{r_1 - i_2} \left( 1 + X + Y \right) = \left( 1 + X \right)^{r_1 - h_2} \left( 1 - X \right)^{h_2} \left( 1 + Y \right)^{r_1} \left( 1 - Y \right)^{r_2}.
\]  

(C-4)

Taking the terms on both sides of (C-4) for which the degree of Y is \( t_1 \) or less, substituting \( \epsilon/(1-\epsilon) \) for X and 1 for Y and multiplying the both sides by \( n_1 \), we obtain Eq. (47) from (42).
APPENDIX D

Proof of Theorem 3

Let \( F(X_1, X_2, \ldots, X_{m+1}, Y) \) be defined as follows

\[
F(X_1, X_2, \ldots, X_{m+1}, Y) = \sum_{i_1=0}^{m_1} \sum_{i_2=0}^{m_2} \ldots \sum_{i_{m+1}=0}^{m_{m+1}} A_{i_1, i_2, \ldots, i_{m+1}} \cdot (1+X_{m+1})^{i_{m+1}}
\]

\[
\left( \prod_{j=0}^{m_1} \sum_{j_h=0}^{j_{m_1+1}} w_{j_h, s_{j_h}} (x_h)^{i_h} \right) \left( \prod_{j=0}^{m_1} \sum_{j_{m+1}=0}^{j_{m+1+1}} w_{j_{m+1}, s_{j_{m+1}}} (r_1)^{i_{m+1}} \right). \tag{D-1}
\]

It follows from (17) and generalized MacWilliams identity [7, p. 147] that

\[
F(X_1, X_2, \ldots, X_{m+1}, Y) = \sum_{i_1=0}^{m_1} \sum_{i_2=0}^{m_2} \ldots \sum_{i_{m+1}=0}^{m_{m+1}} A^{(1)} \prod_{h=1}^{m_1+1} (1+X_h)^{i_h} \cdot (X_h + Y) \cdot (1+X_{m+1})^{i_{m+1}}
\]

\[
= 2^{r_1} \sum_{i_1=0}^{m_1} \sum_{i_2=0}^{m_2} \ldots \sum_{i_{m+1}=0}^{m_{m+1}} B^{(1)} \prod_{h=1}^{m_1} (1+X_h)^{i_h} \cdot (X_h + Y) \cdot (1+X_{m+1})^{i_{m+1}} \cdot \prod_{h=1}^{m_1+1} (1+Y)^{i_h} \cdot (1+X_{m+1})^{i_{m+1}} \cdot (1+Y)^{i_{m+1}}. \tag{D-2}
\]

Let \( H \) be a subset of \{1, 2, 3, \ldots, m_1\} and \( F_{H, t_1}(X_1, X_2, \ldots, X_{m+1}, Y) \) be the sum of the terms of \( F(X_1, X_2, \ldots, X_{m+1}, Y) \) for which the degree of \( X_h \) is nonzero for \( h \in H \) and is zero for \( h \in \{1, 2, \ldots, m_1\} - H \), and the degree of \( Y \) is \( t_1 \) or less.

Using (20), and (D-2), we have that

\[
F_{H, t_1}(X_1, X_2, \ldots, X_{m+1}, Y) = 2^{r_1} \sum_{i_1=0}^{m_1} \sum_{i_2=0}^{m_2} \ldots \sum_{i_{m+1}=0}^{m_{m+1}} B^{(1)} \prod_{h=1}^{m_1} (1+X_h)^{i_h} \cdot (X_h + Y) \cdot (1+X_{m+1})^{i_{m+1}} \cdot \prod_{h=1}^{m_1+1} (1+Y)^{i_h} \cdot (1+X_{m+1})^{i_{m+1}} \cdot (1+Y)^{i_{m+1}}. \tag{D-3}
\]
Let \( F_{w,t_1}(X_1,X_2,\ldots,X_{m+1},Y) \) be defined as the sum of \( F_{H,t_1}(X_1,X_2,\ldots,X_{m+1},Y) \) over all the subsets, \( H \)'s, of \( \{1,2,\ldots,m \} \) with exactly \( w \) elements. Then the second term of (50) is equal to

\[
-(1-\varepsilon)^{n_1} \sum_{w_1} F_{w,t_1}(\varepsilon/(1-\varepsilon), \varepsilon/(1-\varepsilon), \ldots, \varepsilon/(1-\varepsilon), 1) \tag{D-4}
\]

Using (D-3), the definition of \( R_w \) given by (51) and the following identity [7, p. 153]:

\[
\sum_{s=0}^{t} P_s(i,n) = P_t(i-1,n-1) . \tag{D-5}
\]

Then (D-4) is equal to

\[
-(1-\varepsilon)^{n_1} \sum_{w_1} F_{w,t_1}(\varepsilon/(1-\varepsilon), \varepsilon/(1-\varepsilon), \ldots, \varepsilon/(1-\varepsilon), 1) \tag{D-5}
\]

\[
= \sum_{i_1=0}^{i_{m+1}} \sum_{i_{m+1}+1}^{i_{m+1}+1} \sum_{i_1=0}^{i_{m+1}} B_{i_1,i_2,\ldots,i_{m+1}} \tag{1-2\varepsilon}
\]

Then (D-4) is equal to

\[
P_{t_1}(\sum_{h=1}^{m+1} i_{h-1,n_i-1}) R_w(i_1,i_2,\ldots,i_{m+1}) .
\]
APPENDIX E

Derivation of (57)

Let

\[ F(X_1, X_2, Y) = \sum_{i_1=0}^{r_1} \sum_{i_2=0}^{r_1} A(i, j) X_1^{i_1} Y^{i_2} \]

where

\[ A(i, j) = \sum_{j_1=0}^{r_1} \sum_{s_1=0}^{r_1} W_{j_1, s_1} (k_1) X_1^{j_1} Y^{s_1} \]

It follows from (17), (20) and the generalized MacWilliams' identity [7, p. 147] that

\[ F(X_1, X_2, Y) = \sum_{i_1=0}^{r_1} \sum_{i_2=0}^{r_1} A(i, j) (1+X_1 Y)^{i_1} (1+X_2 Y)^{i_2} \]

Then, it follows from (E-2) that

\[ F_{j_1, t_1}(X_1, X_2, Y) = 2^{t_1} \sum_{i_1=0}^{r_1} \sum_{i_2=0}^{r_1} B(i, j) P_{j_1, k_1}(i_1) X_1^{j_1} (1+X_2)^{i_1} (1-X_2)^{i_2} \]

By (56), we have that

\[ P_{e^i, j_1}^{(1)} = \left( \frac{k_1}{j_1} \right) (1-\epsilon)^{j_1} \left( \frac{k_1-1}{j_1} \right) \left( \frac{n_1}{j_1-1} \right) F_{j_1, t_1}(e^{(1)} \epsilon/(1-\epsilon), \epsilon/(1-\epsilon), 1) \]

Thus (57) follows from (E-3) and (E-4).
APPENDIX F

Derivation of (81)

Let $F_u(X_1, X_2, \ldots, X_{m_1+1}, Y)$ be the sum of terms of $F(X_1, X_2, \ldots, X_{m_1+1}, Y)$ defined in Appendix D for which the degree of $X_u$ is nonzero and the degree of $Y$ is $t_1$ or less. Using (20) and (D-2), we have that

$$F_u(X, X, \ldots, X, Y) = 2^{-r_1} \sum_{i_1=0}^{r_1} \sum_{i_{m_1+1}=0}^{r_1} \sum_{i_1, i_2, \ldots, i_{m_1+1}} B(i_1, i_2, \ldots, i_{m_1+1})$$

$$= \left[ \sum_{s=0}^{t_1} \sum_{h=1}^{m_1} \prod_{1 \leq h \leq m_1 \atop h \neq u} (1+X)^{l-i_h} (1-X)^{l-i_h} (1+X)^{h} (1-X)^{u-1} \right] \left[ \sum_{s=0}^{t_1} \sum_{h=1}^{m_1} \prod_{1 \leq h \leq m_1 \atop h \neq u} (1+X)^{l-i_h} (1-X)^{l-i_h} (1+X)^{h} (1-X)^{u-1} \right]$$

The second term of (80) is equal to

$$(1-\varepsilon)^{n_1} F_u(\varepsilon/(1-\varepsilon), \varepsilon/(1-\varepsilon), \ldots, \varepsilon/(1-\varepsilon), 1) .$$

Then (81) follows from (D-5).
Figure 1 A cascaded coding system
Figure 2 Block format
Figure 3 Inner code decoding
Figure 4 Outer code decoding
Figure 5
Figure 6 An interleaved block
Figure 7 $n_2$ decoded segments