CHAPTER 21

Burst-Error-Correcting Convolutional Codes

Block codes designed to correct the type of burst errors or combinations of burst and random errors that typically occur on fading channels were discussed in the previous chapter. A number of *burst-error-correcting convolutional codes* have also been constructed for this purpose. Most of these codes are designed for use with hard-decision demodulator outputs and employ the feedback syndrome decoding techniques introduced in Chapter 13.

Convolutional codes for correcting burst errors were first constructed by Hagelbarger [1]. More efficient codes of the same type were later constructed independently by Iwadare [2] and Massey (see Gallager [3]). Convolutional codes for correcting phased burst errors were first studied by Wyner and Ash [4]. Optimal codes of the same type were later discovered independently by Berlekamp [5] and Preparata [6]. Both the Iwadare–Massey codes and the Berlekamp–Preparata codes are presented in Section 21.2.

The interleaving technique presented in Chapter 15 can also be used to convert convolutional codes for correcting random errors into burst-error-correcting codes. Interleaving is discussed in Section 21.3.

Three different approaches to correcting both burst and random errors with convolutional codes are presented in Section 21.4. The diffuse codes of Kohlenberg and Forney [7] and Massey [8] can be decoded with a simple majority-logic decoding rule. Gallager's [3] burst-finding scheme uses an adaptive decoder that separates bursts from random errors. Finally, Tong's [9] burst-trapping scheme applies the same idea to block codes, although the overall code remains convolutional because there is memory in the encoder.

We begin our discussion of burst-error-correcting convolutional codes by considering bounds on burst-error-correcting capability.

21.1 BOUNDS ON BURST-ERROR-CORRECTING CAPABILITY

Assume that $\mathbf{e} = (e_0, e_1, e_2, \cdots)$ represents the channel error sequence on a BSC.

DEFINITION 21.1 A sequence of error bits $e_{l+1}, e_{l+2}, \cdots, e_{l+b}$ is called a *burst of length b* relative to a *guard space of length g* if

1. $e_{l+1} = e_{l+b} = 1$;
2. the g bits preceding e_{l+1} and the g bits following e_{l+b} are all 0's; and
3. the b bits from e_{l+1} through e_{l+b} contain no subsequence of g 0's.

EXAMPLE 21.1 Burst Lengths and Guard Spaces

Consider the error sequence $\mathbf{e} = (\cdots 0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,0\,0\,1\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0 \cdots)$. This sequence contains a burst of length $b = 28$ relative to a guard space of length $g = 6$. Alternatively, it contains two bursts, one of length 7 and the other of length 17, relative to a guard space of length 4, or three bursts, of lengths 7, 6, and 8, relative to a guard space of length 3. This example illustrates that the length of a burst is always determined relative to some guard space and that the two cannot be specified independently.

Gallager [3] has shown that for any convolutional code of rate R that corrects all bursts of length b or less relative to a guard space of length g,

$$\frac{g}{b} \geq \frac{1+R}{1-R}. \tag{21.1}$$

The bound of (21.1) is known as the bound on complete burst-error correction.[1] Massey [10] has also shown that if we allow a small fraction of the bursts of length b to be decoded incorrectly, the guard space requirements can be reduced significantly. In particular, for a convolutional code of rate R that corrects all but a fraction ε of bursts of length b or less relative to a guard space of length g,

$$\frac{g}{b} \geq \frac{R + [\log_2(1-\varepsilon)]/b}{1-R} \approx \frac{R}{1-R} \tag{21.2}$$

for small ε. The bound of (21.2) is known as the bound on "almost all" burst-error correction.

EXAMPLE 21.2 Bounds on Burst-Error Correction

For $R = 1/2$, complete burst-error correction requires $g/b \geq 3$, whereas "almost all" burst-error correction only requires a g/b ratio of approximately 1, a difference of a factor of 3 in the necessary guard space.

21.2 BURST-ERROR-CORRECTING CONVOLUTIONAL CODES

In this section we discuss two classes of convolutional codes for correcting burst errors, the Berlekamp–Preparata codes and the Iwadare–Massey codes.

21.2.1 Berlekamp–Preparata Codes

Consider designing an $(n, n-1, m)$ systematic feedforward convolutional encoder to correct a phased burst error confined to a single block of n bits relative to a guard space of m error-free blocks; that is, a burst can affect at most one block within a span of $(m+1)$ blocks. The code produced by such an encoder would have a phased-burst-error correcting capability of one block relative to a guard space of m blocks. To design such a code, we must assure that each correctable error sequence

[1] Wyner and Ash [4] earlier obtained a special case of this bound for phased-burst-error correction.

$[\mathbf{e}]_m = (\mathbf{e}_0, \mathbf{e}_1, \cdots, \mathbf{e}_m)$ results in a distinct syndrome $[\mathbf{s}]_m = (s_0, s_1, \cdots, s_m)$. This implies that each error sequence with $\mathbf{e}_0 \neq \mathbf{0}$ and $\mathbf{e}_l = \mathbf{0}, l = 1, 2, \cdots, m$, must yield a distinct syndrome and that each of these syndromes must be distinct from the syndrome caused by any error sequence with $\mathbf{e}_0 = \mathbf{0}$ and a single block $\mathbf{e}_l \neq \mathbf{0}, l = 1, 2, \cdots, m$. Under these conditions, the first error block $\mathbf{e}_0$ can be correctly decoded if the first $(m + 1)$ blocks of $\mathbf{e}$ contain at most one nonzero block, and assuming feedback decoding, each successive error block can be decoded in the same way.

An $(n, n-1, m)$ systematic code is described by the set of generator polynomials $\mathbf{g}_1^{(n-1)}(D), \mathbf{g}_2^{(n-1)}(D), \cdots, \mathbf{g}_{n-1}^{(n-1)}(D)$. We can write the transpose of the parity-check matrix as

$$[\mathbb{H}^T]_m = \begin{bmatrix} \mathbb{B}_0 \\ \mathbb{B}_1 \\ \vdots \\ \mathbb{B}_m \end{bmatrix}, \tag{21.3}$$

where

$$\mathbb{B}_0 = \begin{bmatrix} g_{1,0}^{(n-1)} & g_{1,1}^{(n-1)} & \cdots & g_{1,m}^{(n-1)} \\ \vdots & \vdots & & \vdots \\ g_{n-1,0}^{(n-1)} & g_{n-1,1}^{(n-1)} & & g_{n-1,m}^{(n-1)} \\ 1 & 0 & \cdots & 0 \end{bmatrix} \tag{21.4}$$

is an $n \times (m + 1)$ matrix (refer to the transpose of the $\mathbb{H}$ matrix in (11.45) for comparison). For $0 < l \leq m$, we obtain $\mathbb{B}_l$ from $\mathbb{B}_{l-1}$ by shifting $\mathbb{B}_{l-1}$ one column to the right and deleting the last column. Mathematically, this operation can be expressed as

$$\mathbb{B}_l = \mathbb{B}_{l-1} \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \triangleq \mathbb{B}_{l-1}\mathbb{T}, \tag{21.5}$$

where $\mathbb{T}$ is an $(m + 1) \times (m + 1)$ shifting matrix. Using this notation, we can write the syndrome as

$$\begin{aligned} [\mathbf{s}]_m = [\mathbf{e}]_m[\mathbb{H}^T]_m &= \mathbf{e}_0\mathbb{B}_0 + \mathbf{e}_1\mathbb{B}_1 + \mathbf{e}_2\mathbb{B}_2 + \cdots + \mathbf{e}_m\mathbb{B}_m \\ &= \mathbf{e}_0\mathbb{B}_0 + \mathbf{e}_1\mathbb{B}_0\mathbb{T} + \mathbf{e}_2\mathbb{B}_0\mathbb{T}^2 + \cdots + \mathbf{e}_m\mathbb{B}_0\mathbb{T}^m. \end{aligned} \tag{21.6}$$

From (21.1), for an optimum burst-error-correcting code, $g/b = (1 + R)/(1 - R)$. For the preceding case with $R = (n - 1)/n$ and $g = mn = mb$, this implies that

$$\frac{g}{b} = m = \frac{1 + \frac{n-1}{n}}{1 - \frac{n-1}{n}} = 2n - 1; \tag{21.7}$$

that is, $\mathbb{B}_0$ is an $n \times 2n$ matrix. We must now choose $\mathbb{B}_0$ such that the conditions for burst-error correction are satisfied.

If we choose the first n columns of $\mathbb{B}_0$ to be the *skewed* $n \times n$ identity matrix

$$\begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix},$$

then (21.6) implies that each error sequence with $\mathbb{e}_0 \neq \mathbb{0}$ and $\mathbb{e}_l = \mathbb{0}$, $l = 1, 2, \cdots, m$, will yield a distinct syndrome. In this case, we obtain the estimate of $\mathbb{e}_0$ simply by reversing the first n bits in the $2n$-bit syndrome. In addition, for each $\mathbb{e}_0 \neq \mathbb{0}$, the condition

$$\mathbb{e}_0\mathbb{B}_0 \neq \mathbb{e}_l\mathbb{B}_0\mathbb{T}^l, \quad l = 1, 2, \cdots, m, \tag{21.8}$$

must be satisfied for $\mathbb{e}_l \neq \mathbb{0}$. This ensures that an error in some other block will not be confused for an error in block zero. We choose the last n columns of $\mathbb{B}_0$ to be the $n \times n$ matrix

$$\begin{bmatrix} 0 & A & B & D & \cdots \\ 0 & 0 & C & E & \cdots \\ 0 & 0 & 0 & F & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

where $A, B, C, \cdots$ must be chosen so that (21.8) is satisfied.

First, note that for any $\mathbb{e}_l \neq \mathbb{0}$ and $l \geq n$, the first n positions in the vector $\mathbb{e}_l\mathbb{B}_0\mathbb{T}^l$ must be all 0's, since $\mathbb{T}^l$ shifts $\mathbb{B}_0$ such that $\mathbb{B}_0\mathbb{T}^l$ has all 0's in its first l columns; however, for any $\mathbb{e}_0 \neq \mathbb{0}$, the vector $\mathbb{e}_0\mathbb{B}_0$ cannot have all 0's in its first n positions. Hence, condition (21.8) is automatically satisfied for $n \leq l \leq m = 2n - 1$, and we can replace (21.8) with the condition that for each $\mathbb{e}_0 \neq \mathbb{0}$,

$$\mathbb{e}_0\mathbb{B}_0 \neq \mathbb{e}_l\mathbb{B}_0\mathbb{T}^l, \quad l = 1, 2, \cdots, n-1, \tag{21.9}$$

must be satisfied for all $\mathbb{e}_l \neq \mathbb{0}$. Now, note the following:

1. The matrix $\mathbb{B}_0$ has rank n, since it contains a skewed $n \times n$ identity matrix in its first n columns.
2. For $1 \leq l \leq n-1$, the matrix $\mathbb{B}_0\mathbb{T}^l$ also has rank n, since it also contains a skewed $n \times n$ identity matrix in some n columns.
3. Condition (21.9) is equivalent to requiring that the row spaces of $\mathbb{B}_0$ and $\mathbb{B}_0\mathbb{T}^l$ have only the vector $\mathbb{0}$ in common.

Hence, we can replace condition (21.9) with the condition that

$$rank \begin{bmatrix} \mathbb{B}_0\mathbb{T}^l \\ \hline \mathbb{B}_0 \end{bmatrix} = rank\,[\mathbb{B}_0\mathbb{T}^l] + rank\,[\mathbb{B}_0] = n + n = 2n. \tag{21.10}$$

Because $\begin{bmatrix} \mathbb{B}_0\mathbb{T}^l \\ ----- \\ \mathbb{B}_0 \end{bmatrix}$ is a $2n \times 2n$ matrix, condition (21.10) is equivalent to requiring that this matrix be nonsingular, that is, that

$$\det \begin{bmatrix} \mathbb{B}_0\mathbb{T}^l \\ ----- \\ \mathbb{B}_0 \end{bmatrix} \neq 0, \quad l = 1, 2, \cdots, n-1. \tag{21.11}$$

We now show that $A, B, C, \cdots$ can always be chosen so that (21.11) is satisfied. First, for $n = 2$, (21.11) yields the condition

$$\det \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & A \\ 1 & 0 & 0 & 0 \end{bmatrix} = 1, \tag{21.12}$$

which is satisfied by choosing $A = 1$. For $n = 3$, (21.11) yields two conditions,

$$\det \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & B \\ 0 & 1 & 0 & 0 & 0 & C \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = 1, \quad \det \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & B \\ 0 & 1 & 0 & 0 & 0 & C \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = 1, \tag{21.13}$$

which are satisfied by choosing $B = 1$ and $C = 1$. It can be shown by induction that the missing elements of $\mathbb{B}_0$ can always be chosen so that (21.11) is satisfied (see Problem 21.1).

EXAMPLE 21.3 An Optimum Phased-Burst-Error-Correcting Code

For $n = 4$, the 4×8 matrix $\mathbb{B}_0$ is given by

$$\mathbb{B}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{21.14}$$

which results in a rate $R = 3/4$ systematic convolutional code with generator polynomials $\mathbf{g}_1^{(3)}(D) = D^3 + D^5 + D^6 + D^7$, $\mathbf{g}_2^{(3)}(D) = D^2 + D^6 + D^7$, and $\mathbf{g}_3^{(3)}(D) = D + D^7$ that is capable of correcting phased bursts of length $n = 4$ bits confined to a single block relative to a guard space of $m = 7$ blocks ($g = nm = 28$ bits). Because $g/b = 28/4 = 7$, and $(1+R)/(1-R) = (7/4)(1/4) = 7$, this code meets the Gallager bound of (21.1) and is optimal for phased-burst-error correction.

The foregoing construction, discovered independently by Berlekamp [5] and Preparata [6], always results in a code that meets the Gallager bound of (21.1). Hence, the *Berlekamp–Preparata codes* are optimum for phased-burst-error correction. We can also extend this construction to generate optimum phased-burst-error correcting codes for $k < n - 1$ (see Problem 21.2). In addition, we can use interleaving (see Section 21.3) to convert any of these codes to phased-burst-error-correcting codes that are capable of correcting bursts confined to λ blocks relative to a guard space of λm blocks, where λ is the degree of interleaving. Because the ratio of guard space to burst length remains the same, these interleaved codes are still optimum.

The Berlekamp–Preparata codes can be decoded using a general decoding technique for burst-error-correcting convolutional codes due to Massey [11]. We recall from (21.6) that the set of possible syndromes for a burst confined to block 0 is simply the row space of the $n \times 2n$ matrix $\mathbb{B}_0$. Hence, if $\mathbf{e}_0 \neq \mathbf{0}$ and $\mathbf{e}_l = \mathbf{0}, l = 1, 2, \cdots, m$, $[\mathbf{s}]_m$ is a codeword in the $(2n, n)$ block code generated by $\mathbb{B}_0$; however, if $\mathbf{e}_0 = \mathbf{0}$ and a single block $\mathbf{e}_l \neq \mathbf{0}$ for some $l, 1 \leq l \leq m$, condition (21.8) ensures that $[\mathbf{s}]_m$ is not a codeword in the block code generated by $\mathbb{B}_0$. Therefore, $\mathbf{e}_0$ contains a correctable error pattern if and only if $[\mathbf{s}]_m$ is a codeword in the block code generated by $\mathbb{B}_0$. This requires determining if $[\mathbf{s}]_m \mathbb{H}_0^T = \mathbf{0}$, where $\mathbb{H}_0$ is the $n \times 2n$ block code parity-check matrix corresponding to $\mathbb{B}_0$. Because

$$\mathbb{B}_0 = \left[\begin{array}{ccccc|cccccc} 0 & \cdots & 0 & 0 & 1 & 0 & A & B & D & \cdots \\ 0 & \cdots & 0 & 1 & 0 & 0 & 0 & C & E & \cdots \\ 0 & \cdots & 1 & 0 & 0 & 0 & 0 & 0 & F & \cdots \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\ 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{array}\right], \tag{21.15}$$

the corresponding block code parity-check matrix is given by

$$\mathbb{H}_0 = \left[\begin{array}{ccccc|ccccc} 0 & \ddots & \vdots & \vdots & \vdots & 0 & \cdots & 0 & 0 & 1 \\ \vdots & \ddots & F & E & D & 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & C & B & 0 & \cdots & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & A & \vdots & \iddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 \end{array}\right]. \tag{21.16}$$

If $[\mathbf{s}]_m \mathbb{H}_0^T = \mathbf{0}$, the decoder must then find the correctable error pattern $\mathbf{e}_0$ that produces the syndrome $[\mathbf{s}]_m$. Because in this case $[\mathbf{s}]_m = \mathbf{e}_0 \mathbb{B}_0$, we obtain the estimate of $\mathbf{e}_0$ simply by reversing the first n bits in $[\mathbf{s}]_m$. For a feedback decoder, the syndrome must then be modified to remove the effect of $\mathbf{e}_0$. But for a correctable error pattern, $[\mathbf{s}]_m = \mathbf{e}_0 \mathbb{B}_0$ depends only on $\mathbf{e}_0$, and hence when the effect of $\mathbf{e}_0$ is removed the syndrome will be reset to all zeros.

EXAMPLE 21.3 (Continued)

For $n = 4$, the 4×8 block code parity-check matrix $\mathbb{H}_0$ is given by

$$\mathbb{H}_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \qquad (21.17)$$

A complete encoder/decoder block diagram for the code described in Example 21.3 is shown in Figure 21.1. The encoder is implemented using the observer canonical form realization for systematic encoders introduced in Example 11.5.[2] Because it takes $m = 7$ time units to form the syndrome $[\mathbf{s}]_m$, the received information sequence must be delayed by $m = 7$ time units before error correction can begin, and thus it is convenient to use a controller canonical form realization to "encode" the received information sequences at the decoder. The OR gate in the decoder has as inputs the four parity checks corresponding to the four rows of $\mathbb{H}_0$. At time unit $i = 7$, the output of this OR gate is 0 if and only if the syndrome $[\mathbf{s}]_7$ corresponds to a correctable burst in $\mathbf{e}_0$. When this happens the NOT gate activates the connections from s_3, s_2, and s_1 to provide the estimates $\hat{e}_0^{(1)}$, $\hat{e}_0^{(2)}$, and $\hat{e}_0^{(2)}$, respectively. In addition, the output of the OR gate is fed back to the AND gates between stages of the syndrome register. This resets the syndrome to all zeros after each burst is corrected.

Finally, we note that unlimited error propagation cannot occur in a decoder of the type just described. When the syndrome is reset the only possible change is that some 1's are changed to 0's. Hence, if the received sequence is error-free for $2(m + 1)$ time units, the syndrome register must be cleared to all zeros. Because a guard space is m time units, this means that approximately two error-free guard spaces are needed to restore a decoder of this type to correct operation following a decoding error.

The phased-burst-error-correcting codes of Berlekamp and Preparata also can be used to correct bursts of length b relative to a guard space of length g, where b and g are not confined to an integral number of blocks. In general, though, the Gallager bound will not be met with equality in this case; that is, the codes will not achieve the optimum ratio of guard space to burst length. However, for a code interleaved to degree λ, the shortest burst affecting $\lambda + 1$ blocks contains $(\lambda - 1)n + 2$ bits. Hence, the code will correct all bursts of length $b = (\lambda - 1)n + 1$ bits. Similarly, the longest guard space required to cover λm blocks is $g = (\lambda m + 1)n - 1$ bits, and

$$\frac{g}{b} = \frac{(\lambda m + 1)n - 1}{(\lambda - 1)n + 1} = \frac{\lambda m + \frac{n-1}{n}}{\lambda - \frac{n-1}{n}} \approx m = \frac{1 + R}{1 + R} \qquad (21.18)$$

when λ is large. Hence, the Berlekamp–Preparata codes are almost optimum for ordinary burst-error correction when the degree of interleaving λ is large.

[2] In this chapter, as in Sections 13.5–13.7, we use the labeling order of input and output sequences for rate $R = (n - 1)/n$ systematic feedforward encoder realizations given in Figure 11.5 for reasons of consistency with the published literature.

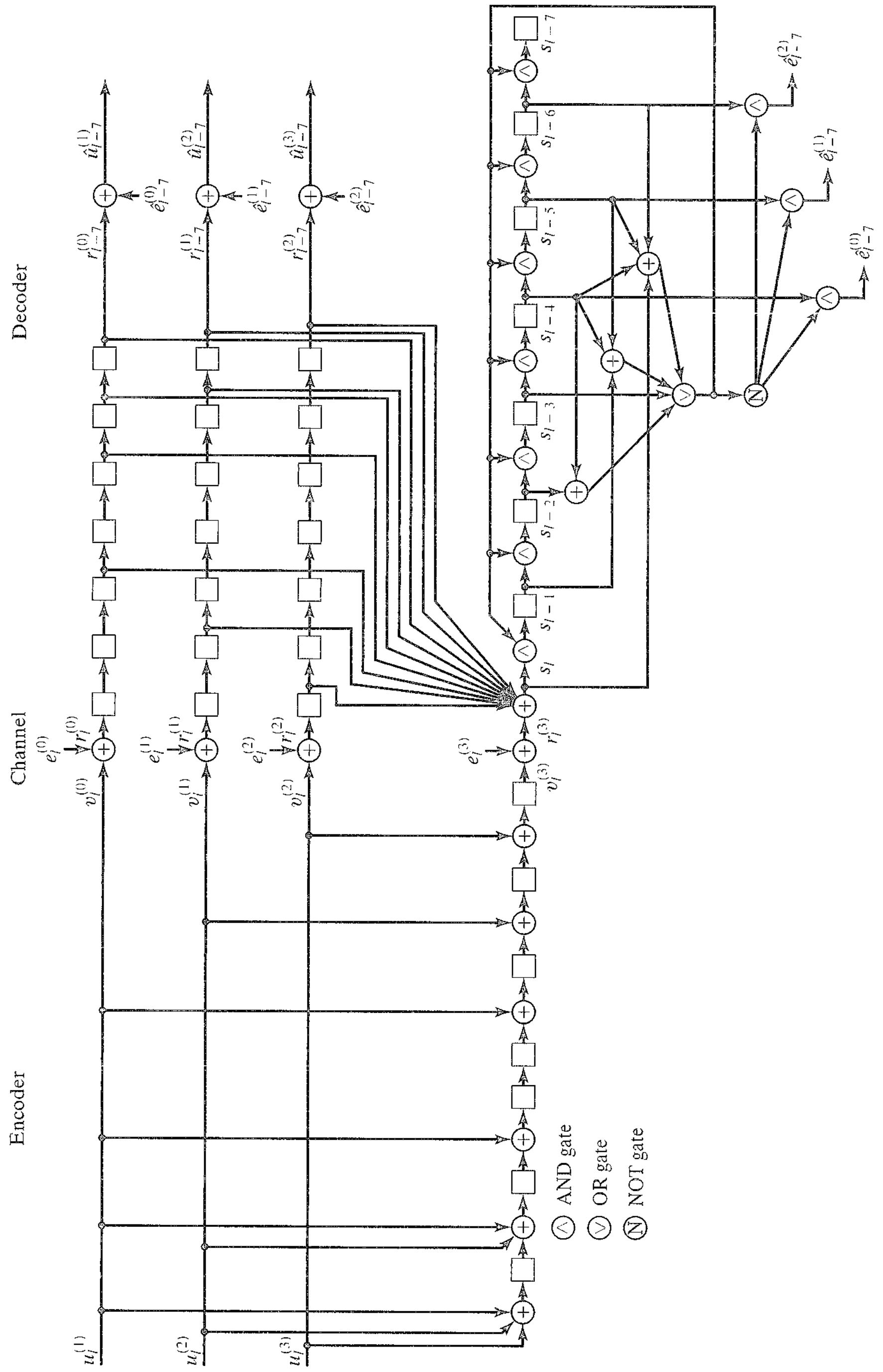


FIGURE 21.1: Complete system block diagram for the (4, 3, 7) Berlekamp–Preparata code.

21.2.2 Iwadare–Massey Codes

Another efficient class of convolutional codes for correcting burst errors of length b relative to a guard space of length g was discovered independently by Iwadare [2] and Massey (see Gallager [3]). These are called the *Iwadare–Massey codes*. For any n, a systematic $(n, n-1, m)$ burst-error-correcting convolutional code can be constructed with the following parameters:

$$\begin{aligned} m &= (2n-1)\lambda + 2n - 3 \\ b &= n\lambda \\ g &= n_A - 1 = n(m+1) - 1, \end{aligned} \tag{21.19}$$

where λ is any positive integer. The $(n-1)$ generator polynomials are given by

$$\mathbf{g}_{(i)}^{(n-1)}(D) = D^{a(i)} + D^{b(i)}, \quad i = 1, 2, \cdots, n-1, \tag{21.20}$$

where $a(i) \triangleq (\lambda+1)(n-i) - 1$, and $b(i) \triangleq (\lambda+1)(2n-i) + i - 3$. The decoding of these codes can best be explained by an example.

EXAMPLE 21.4 Decoding Iwadare–Massey Codes

Consider the rate $R = 2/3$ Iwadare–Massey code with $n = 3$ and $\lambda = 3$. In this case $m = 18$, $b = 9$, and $g = 56$. The generator polynomials are $\mathbf{g}_1^{(2)}(D) = D^7 + D^{18}$ and $\mathbf{g}_2^{(2)}(D) = D^3 + D^{15}$. The observer canonical form encoding circuit for this code is shown in Figure 21.2.

Assume that a burst error of length $b = 9$ begins with the first bit of block 0:

$$\mathbf{e} = (e_0^{(0)} e_0^{(1)} e_0^{(2)} e_1^{(0)} e_1^{(1)} e_1^{(2)} e_2^{(0)} e_2^{(1)} e_2^{(2)} 0 0 0 \cdots). \tag{21.21}$$

Then, from (13.57) the syndrome sequence $\mathbf{s}(D)$ is given by

$$\begin{aligned} \mathbf{s}(D) &= \mathbf{e}^{(0)}(D)\mathbf{g}_1^{(2)}(D) + \mathbf{e}^{(1)}(D)\mathbf{g}_2^{(2)}(D) + \mathbf{e}^{(2)}(D) \\ &= e_0^{(2)} + e_1^{(2)} D + e_2^{(2)} D^2 + e_0^{(1)} D^3 + e_1^{(1)} D^4 + e_2^{(1)} D^5 \\ &\quad + e_0^{(0)} D^7 + e_1^{(0)} D^8 + e_2^{(0)} D^9 + e_0^{(1)} D^{15} + e_1^{(1)} D^{16} + e_2^{(1)} D^{17} \\ &\quad + e_0^{(0)} D^{18} + e_1^{(0)} D^{19} + e_2^{(0)} D^{20}. \end{aligned} \tag{21.22a}$$

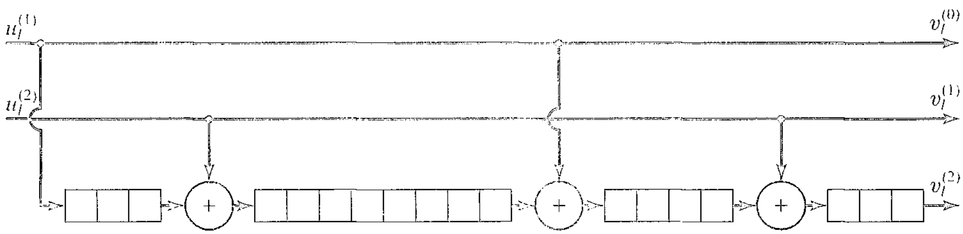


FIGURE 21.2: Encoding circuit for the (3, 2, 18) Iwadare–Masseey code with $\lambda = 3$.

or

$$\mathbf{s} = (e_0^{(2)} e_1^{(2)} e_2^{(2)} e_0^{(1)} e_1^{(1)} e_2^{(1)} 0 e_0^{(0)} e_1^{(0)} e_2^{(0)} 0\,0\,0\,0\,0 e_0^{(1)} e_1^{(1)} e_2^{(1)} e_0^{(0)} e_1^{(0)} e_2^{(0)}). \tag{21.22b}$$

Examining this syndrome sequence, we note the following:

1. Each information error bit appears twice in $\mathbf{s}$.
2. $e_l^{(1)}$ appears in $\mathbf{s}$ before $e_l^{(0)}$, $l = 0, 1, 2$.
3. The number of positions between the two appearances of $e_l^{(j)}$ is $10 + j$, for $j = 0, 1$ and $l = 0, 1, 2$.

The integer $10+j$ is called the *repeat distance* of $e_l^{(j)}$, and we see that the information error bits $e_l^{(0)}$ and $e_l^{(1)}$ have distinct repeat distances of 10 and 11, respectively. This fact will be useful in decoding.

If the burst starts with the second bit of block 0, then

$$\mathbf{e} = (0 e_0^{(1)} e_0^{(2)} e_1^{(0)} e_1^{(1)} e_1^{(2)} e_2^{(0)} e_2^{(1)} e_2^{(2)} e_3^{(0)} 0\,0\,0 \cdots) \tag{21.23}$$

and

$$\mathbf{s} = (e_0^{(2)} e_1^{(2)} e_2^{(2)} e_0^{(1)} e_1^{(1)} e_2^{(1)} 0\,0 e_1^{(0)} e_2^{(0)} e_3^{(0)} 0\,0\,0\,0 e_0^{(1)} e_1^{(1)} e_2^{(1)} 0 e_1^{(0)} e_2^{(0)} e_3^{(0)}). \tag{21.24}$$

If the burst starts with the third bit of block 0,

$$\mathbf{e} = (0\,0 e_0^{(2)} e_1^{(0)} e_1^{(1)} e_1^{(2)} e_2^{(0)} e_2^{(1)} e_2^{(2)} e_3^{(0)} e_3^{(1)} 0\,0\,0 \cdots) \tag{21.25}$$

and

$$\mathbf{s} = (e_0^{(2)} e_1^{(2)} e_2^{(2)} 0 e_1^{(1)} e_2^{(1)} e_3^{(1)} 0 e_1^{(0)} e_2^{(0)} e_3^{(0)} 0\,0\,0\,0\,0 e_1^{(1)} e_2^{(1)} e_3^{(1)} e_1^{(0)} e_2^{(0)} e_3^{(0)}). \tag{21.26}$$

In each of these cases, the repeat distance for the information error bit $e_l^{(j)}$ is still $10 + j$.

The decoding circuit for this code is shown in Figure 21.3. It consists of a systematic feedforward encoder (in controller canonical form) for syndrome calculation and a correction circuit. The two inputs of AND gate A_1 are separated by 11 stages of the syndrome register (the repeat distance of $e_l^{(1)}$). Two of the inputs of AND gate A_0 are separated by 10 stages of the syndrome register (the repeat distance of $e_l^{(0)}$). The third input to A_0 ensures that both AND gates cannot have a 1 output at the same time. A careful examination of the syndromes reveals that A_1 cannot be activated until error bit $e_0^{(1)}$ appears at both its inputs, and then its output will be the correct value of $e_0^{(1)}$. At the same time, the received information bit $r_0^{(1)}$ is at the output of the fifteenth stage of the buffer register that stores the received information sequence $\mathbf{r}^{(1)}$. Correction is then achieved by adding the output of A_1 to $r_0^{(1)}$. The output of A_1 is also fed back to reset the syndrome register. After one shift of the syndrome register, $e_1^{(1)}$ appears at both inputs of A_1. Hence, the output of A_1 will be the correct value of $e_1^{(1)}$, which is then used to correct the received information bit $r_1^{(1)}$. After the next shift of the syndrome register, the decoder estimates $e_2^{(1)}$ and corrects $r_2^{(1)}$ in exactly the same way.

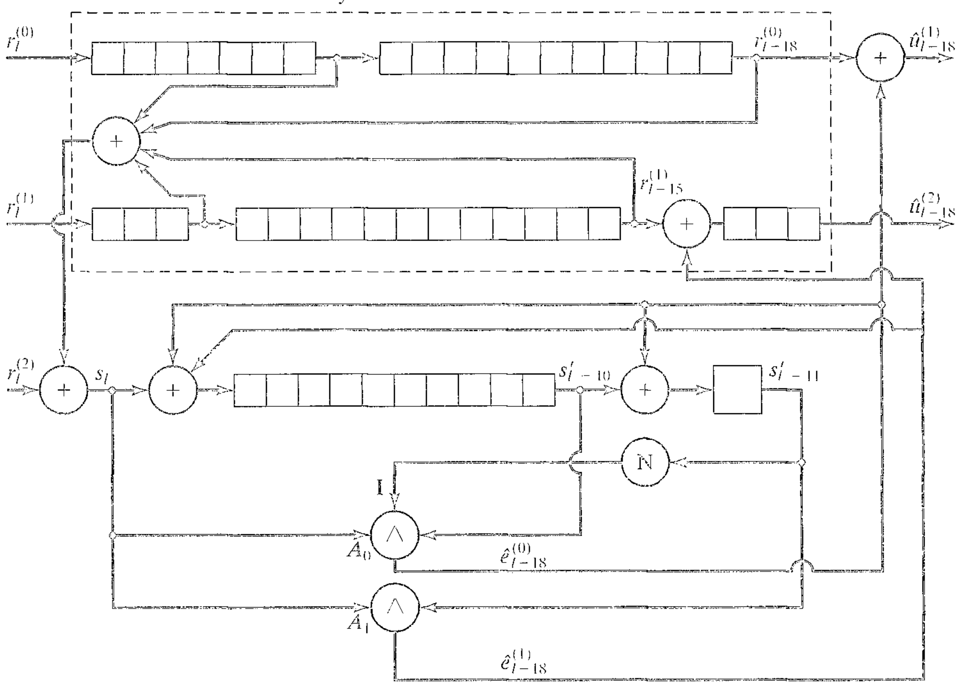


FIGURE 21.3: Decoding circuit for the (3, 2, 18) Iwadare–Massey code with $\lambda = 3$.

Note that the output of the last stage of the syndrome register is inverted and fed into A_0 as input I, which prevents A_0 from making any erroneous estimates while $e_0^{(1)}$, $e_1^{(1)}$, and $e_2^{(1)}$ are being estimated by A_1. For example, when $e_0^{(1)}$ appears at both inputs of A_1, $e_0^{(1)}$ and $e_1^{(1)}$ appear as inputs of A_0. If input I is not provided, and if $e_0^{(1)} = e_1^{(1)} = 1$, then the output of A_0 will be a 1, which would cause an erroneous correction at the output of the buffer register that stores the received information sequence $\mathbf{r}^{(0)}$.

After $e_2^{(1)}$ is corrected, the syndrome register is shifted once, and $e_0^{(0)}$ appears at two inputs of A_0. The last stage of the syndrome register contains a 0, and hence A_1 is prevented from making any erroneous estimates, and input I of A_0 is a 1. Therefore, the output of A_0 is the correct value of $e_0^{(0)}$. At the same time, the received information bit $r_0^{(0)}$ is at the output of the last stage of the buffer register that stores the received information sequence $\mathbf{r}^{(0)}$. Correction is then achieved by adding the output of A_0 to $r_0^{(0)}$. The output of A_0 is also fed back to reset the syndrome register. Hence, after one shift, the last stage of the syndrome register again contains a 0, and the inputs of A_0 are 1, $e_1^{(0)}$, and $e_1^{(0)}$. Therefore, the output of A_0 is the correct value of $e_1^{(0)}$, which is then used to correct the received information bit $r_1^{(0)}$. After the next shift of the syndrome register, the decoder estimates $e_2^{(0)}$ and corrects $r_2^{(0)}$ in exactly the same way. Because a correctable error burst of length $b = 9$ must be followed by an error-free guard space of length $g = 56$, a careful examination of the error bursts and syndromes reveals that the next burst cannot

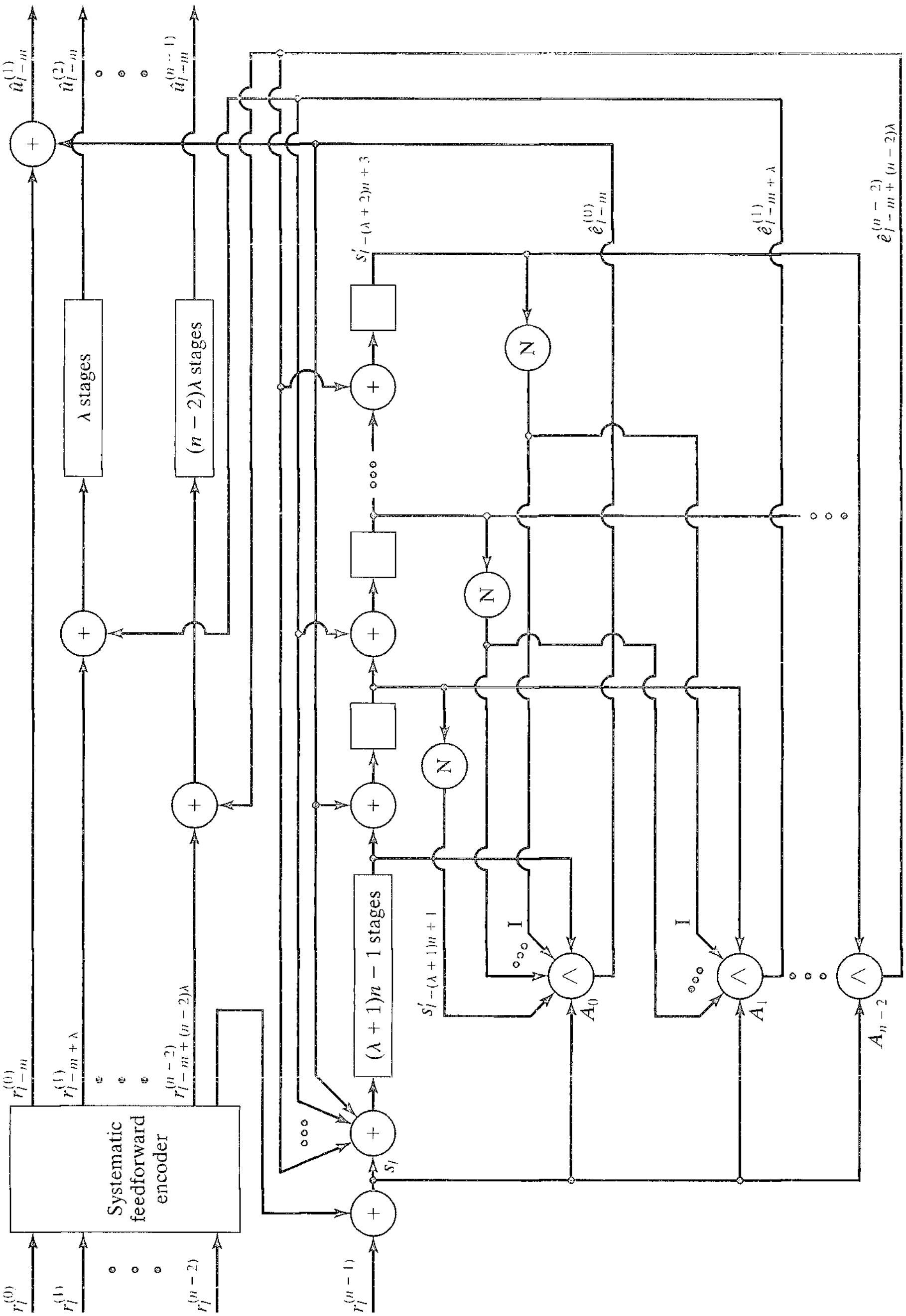


FIGURE 21.4: General decoding circuit for an Iwadare–Massey burst-error-correcting code.

enter the decoder until after the decoder corrects $r_2^{(0)}$, at which time the syndrome register contains all zeros.

A general decoder for an Iwadare–Massey code is shown in Figure 21.4. Note that additional inverted syndrome outputs are fed into AND gates $A_0, A_1, \cdots, A_{n-3}$ to prevent erroneous decoding estimates. It can be shown that unlimited error propagation cannot occur in a decoder of this type, and if a decoding error is followed by $n[m + (\lambda + 2)n - 1] - 1$ error-free bits, the syndrome will be cleared and error propagation terminated (see Problem 21.7).

For $k = n - 1$, the Gallager bound on complete burst-error correction is given by

$$\frac{g}{b} \geq \frac{1+R}{1-R} = \frac{1+\frac{n-1}{n}}{1-\frac{n-1}{n}} = 2n - 1. \tag{21.27}$$

For the Iwadare–Massey codes,

$$\frac{g}{b} = \frac{n(m+1)-1}{n\lambda} = \frac{(2n-1)n\lambda + n(2n-2) - 1}{n\lambda}, \tag{21.28}$$

and we see that these codes require an excess guard space of $2n(n-1) - 1$ bits compared with an optimum code. Hence, they are very efficient for small values of n. For large n, however, the interleaved Berlekamp–Preparata codes require smaller guard spaces if the interleaving degree λ is large enough; but a comparison of decoder hardware shows that Iwadare–Massey codes are simpler to implement (see Problem 21.11). There is a second class of Iwadare–Massey codes that requires a somewhat larger guard space than the class described here, but for large λ results in simpler encoding and decoding circuits (see Problems 21.5 and 21.6).

21.3 INTERLEAVED CONVOLUTIONAL CODES

The technique of interleaving used to obtain good long burst-error-correcting codes from good short random- or burst-error-correcting codes, discussed in the last chapter for block codes, also can be applied to convolutional codes. The idea of interleaving is simply to multiplex the outputs of λ separate encoders for transmission over the channel, where λ is the *interleaving degree*. The received bits are then demultiplexed and sent to λ separate decoders. A burst of length λ on the channel will then look like single errors to each of the separate decoders. Hence, if each decoder is capable of correcting single errors in a decoding length n_A, then, with interleaving, all bursts of length λ or less relative to a guard space of length $(n_A - 1)\lambda$ will be corrected. Similarly, t bursts of length λ on the channel will look like weight-t error sequences to each of the separate decoders. Hence, if each decoder is capable of correcting t errors in a decoding length, then, with interleaving, all sequences of t or fewer bursts of length λ or less relative to a guard space of at most $(n_A - t)\lambda$ will be corrected.[3] More generally, a burst of length $b'\lambda$ on the channel will look like bursts of length b' or less to each of the separate decoders. In this case

[3]The actual guard space requirements between groups of t or fewer bursts depends on how the bursts are distributed. In essence, there can be no more than t bursts of length λ or less in any λ decoding lengths of received bits (see Problem 21.8).

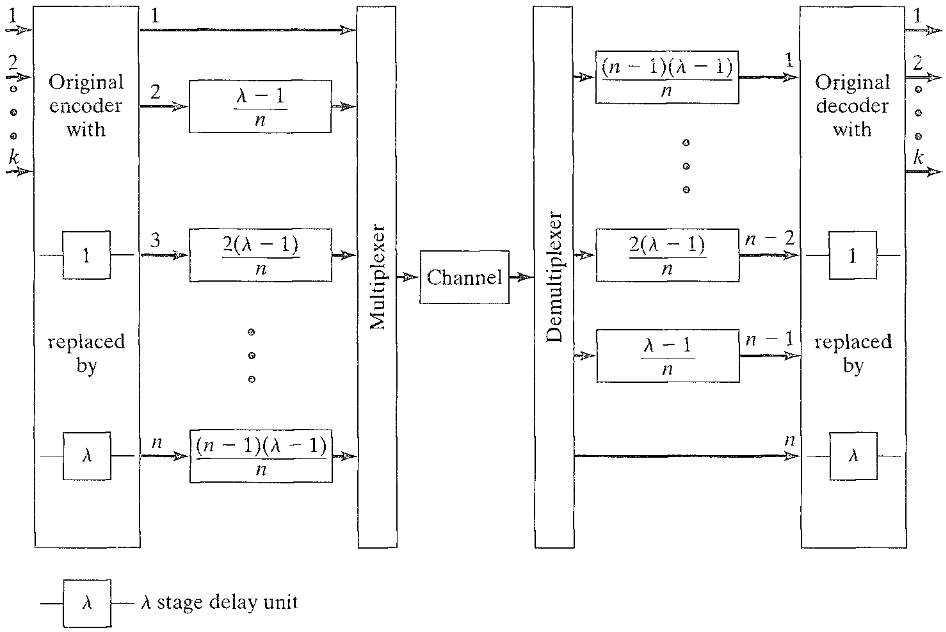


FIGURE 21.5: An (n, k, m) convolutional coding system with interleaving degree λ.

if each decoder is capable of correcting bursts of length b' relative to a guard space g', then the interleaved code will correct all bursts of length $b'\lambda$ or less relative to a guard space $g'\lambda$. In practice it is not necessary to use λ separate encoders and decoders but only to implement one encoder and one decoder in such a way that their operation is equivalent to λ separate encoders and decoders.

An (n, k, m) convolutional coding system with interleaving degree λ is shown in Figure 21.5, where it is assumed that $\lambda - 1$ is a multiple of n. The interleaver is placed between the encoder and the multiplexer and separates the n encoded bits in a block by $\lambda - 1$ interleaved bits prior to transmission over the channel. In addition, the encoder is modified by replacing each delay unit with a string of λ delay units. This makes the encoder equivalent to λ separate encoders whose n-bit encoded blocks are formed in succession and ensures that there will be $\lambda - 1$ interleaved bits between the last bit in one block and the first bit in the next block corresponding to the same encoder. Hence, the encoder of Figure 21.5 achieves an interleaving degree of λ relative to the original convolutional code.

The interleaver of Figure 21.5 requires

$$\frac{(\lambda - 1)}{n} + 2\frac{(\lambda - 1)}{n} + \cdots + (n-1)\frac{(\lambda - 1)}{n} = \frac{(\lambda - 1)(n - 1)}{2} \tag{21.29}$$

delay units. In addition, assuming k m-bit registers for encoding the received information sequences in controller canonical form and $(n - k)$ m-bit syndrome registers, the decoder in Figure 21.5 requires a total of λnm delay units. Hence, the total memory required in the interleaved decoder is given by $(\lambda - 1)(n - 1)/2 + \lambda nm$ delay units.

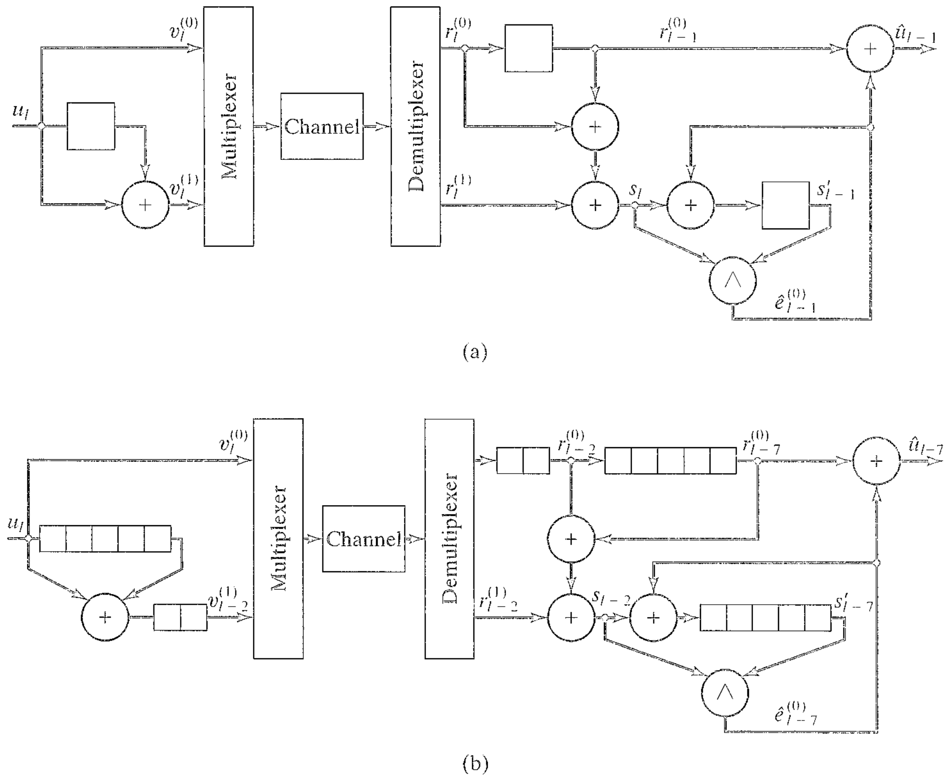


FIGURE 21.6: An interleaved system for a (2, 1, 1) convolutional code with $\lambda = 5$.

EXAMPLE 21.5 An Interleaved Random-Error-Correcting Code

Consider the (2, 1, 1) systematic convolutional code with $\mathbf{g}^{(1)}(D) = 1 + D$. This code can correct single errors in a decoding length of $n_A = n(m+1) = 4$ bits using the simple feedback decoding circuit shown in Figure 21.6(a). Hence, when interleaved to degree λ, where $\lambda - 1$ is a multiple of $n = 2$, this code will correct all bursts of length λ or less with a guard space of 3λ. Because $3\lambda/\lambda = 3$, the ratio of guard space to burst length meets the Gallager bound for rate $R = 1/2$ codes, and this simple interleaved code is optimum for complete burst-error correction! The total memory required in the interleaved decoder is $(\lambda - 1)(n-1)/2 + \lambda nm = (5\lambda - 1)/2$, which equals 12 for $\lambda = 5$. The complete interleaved convolutional coding system is shown in Figure 21.6(b) for $\lambda = 5$.

EXAMPLE 21.6 An Interleaved Burst-Error-Correcting Code

Consider the (4, 3, 7) Berlekamp–Preparata code of Example 21.3 and Figure 21.1, which is capable of correcting phased bursts of one block relative to a guard space of $m = 7$ blocks. If this code is interleaved to degree λ, it can correct bursts confined to λ blocks relative to a guard space of 7λ blocks. Alternatively, it can correct all bursts of length $b = 4(\lambda - 1) + 1$ bits, relative to a guard space of $g = 4(7\lambda + 1) - 1$ bits, and for $\lambda = 5$, $g/b = 143/17 \approx 8.4$, which is about 20% above the Gallager bound of $g/b = 7$ for rate $R = 3/4$ codes.

21.4 BURST-AND-RANDOM-ERROR-CORRECTING CONVOLUTIONAL CODES

Several convolutional coding techniques are available for correcting errors on channels that are subject to a combination of random and burst errors. As noted in the previous section, interleaving a code with random-error-correcting capability t to degree λ results in a code that can correct t or fewer bursts of length λ or less. This is called *multiple-burst-error correction* and requires only that there be no more than t bursts of length λ or less in any λ decoding lengths of received bits. Because some of the bursts may contain only scattered errors, this code in effect corrects a combination of burst and random errors.

Codes also can be constructed to correct a specific combination of burst and random errors. The *diffuse convolutional codes* of Kohlenberg and Forney [7] and Massey [8] are an example of this type of construction. Adaptive decoding algorithms also can be employed to determine which type of error pattern has been received and then switch to the appropriate correction circuit. Gallager's [3] *burst-finding codes* and Tong's [9] *burst-trapping codes* make use of adaptive decoding algorithms.

21.4.1 Diffuse Codes

Consider the $(2, 1, m)$ systematic convolutional code with $m = 3\lambda + 1$ and $\mathbf{g}^{(1)}(D) = 1 + D^{\lambda} + D^{2\lambda} + D^{3\lambda+1}$, where λ is any positive integer greater than 1. The syndrome sequence is given by

$$\mathbf{s}(D) = \mathbf{e}^{(0)}(D)\mathbf{g}^{(1)}(D) + \mathbf{e}^{(1)}(D), \tag{21.30}$$

and four orthogonal check-sums on $e_0^{(0)}$ can be formed as follows:

$$\begin{aligned}
s_0 &= e_0^{(0)} & &+ e_0^{(1)} \\
s_\lambda &= e_0^{(0)} + e_\lambda^{(0)} & &+ e_\lambda^{(1)} \\
s_{2\lambda} + s_{3\lambda} &= e_0^{(0)} & + e_{3\lambda}^{(0)} &+ e_{2\lambda}^{(1)} + e_{3\lambda}^{(1)} \\
s_{3\lambda+1} &= e_0^{(0)} + e_{\lambda+1}^{(0)} + e_{2\lambda+1}^{(0)} & + e_{3\lambda+1}^{(0)} &+ e_{3\lambda+1}^{(1)}.
\end{aligned} \tag{21.31}$$

Hence, if there are two or fewer errors among the 11 error bits checked in (21.31), they can be corrected by a majority-logic decoder.

Now, suppose that a burst of length 2λ or less appears on the channel. If $e_0^{(0)} = 1$, then the only other error bit in (21.31) that could have value 1 is $e_0^{(1)}$, since the other error bits in (21.31) are at least 2λ positions away from $e_0^{(0)}$; however, if $e_0^{(0)} = 0$, a burst of length 2λ can affect at most two of the four check-sums. In either case, the estimate $\hat{e}_0^{(0)}$ made by a majority-logic decoder will be correct. Hence, the

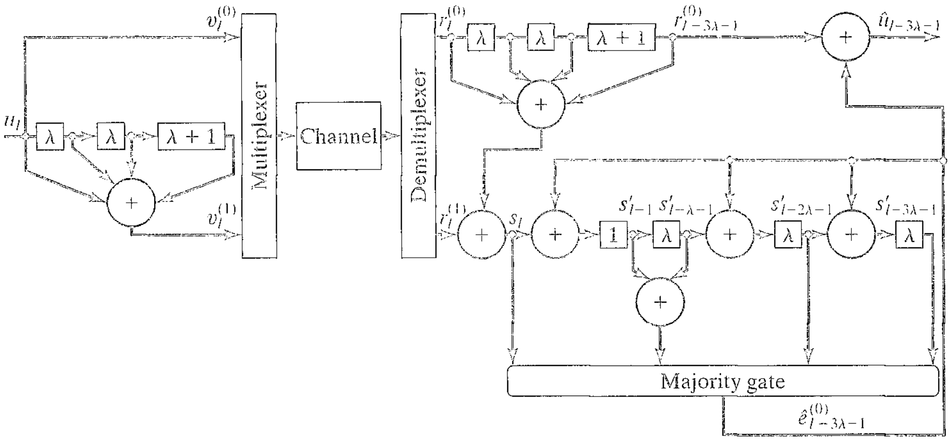


FIGURE 21.7: Complete system block diagram for a rate $R = 1/2$, $t_{ML} = 2$-error-correcting, λ-diffuse code.

code corrects any $t_{ML} = 2$ or fewer random errors among the 11 error bits in (21.31) as well as bursts of length $b = 2\lambda$ or less with a guard space $g = 2(3\lambda + 1) = 6\lambda + 2$. With a feedback decoder, if all past decoding decisions have been correct, the same error-correcting capability applies to the decoding of all information error bits. A complete encoder/decoder block diagram for this code is shown in Figure 21.7. Note that for large λ,

$$\frac{g}{b} = \frac{6\lambda + 2}{2\lambda} \approx 3, \tag{21.32}$$

which is optimum for a rate $R = 1/2$ code according to the Gallager bound for complete burst-error correction.

The code of Figure 21.7 is an example of a diffuse convolutional code. A convolutional code is λ-diffuse and t_{ML}-*error correcting* if $2t_{ML}$ orthogonal check-sums can be found on each block-0 information error bit $e_0^{(i)}$ such that for each $i, 0 \leq i \leq k-1$:

1. Error bits other than $e_0^{(i)}$ from a burst of length $n\lambda$ or less that starts in block 0 and includes $e_0^{(i)}$ are checked by no more than $t_{ML} - 1$ of the check-sums orthogonal on $e_0^{(i)}$.
2. Error bits from a burst of length $n\lambda$ or less that starts anywhere after the ith position in block 0 are checked by no more than t_{ML} of the check-sums orthogonal on $e_0^{(i)}$.

Hence, a majority-logic decoder will correctly estimate each information error bit $e_0^{(i)}$ when there are t_{ML} or fewer random errors in the $2t_{ML}$ orthogonal sums checking $e_0^{(i)}$, or when the first decoding length of received bits contains a burst of length $b = n\lambda$ or less with a guard space $g = n_A - t_{ML}$. With feedback decoding, the same error-correcting capability applies to all information error bits if all past decoding decisions have been correct. The error-propagation properties of diffuse codes are examined in Problem 21.13.

For the special case in which $t_{ML} = 1$ and $R = (n-1)/n$, the Iwadare–Massey codes discussed in Section 21.2 form a class of λ-diffuse, single-error-correcting codes. For any t_{ML}, Ferguson [12] has constructed a class of λ-diffuse codes with rate $R = 1/2$ and $g/b \approx 4$, and Tong [13] has constructed a similar class of asymptotically optimum rate $R = 1/2$, λ-diffuse codes for which $g/b \to 3$ as λ becomes large. Some of Tong's rate $R = 1/2$ diffuse codes are listed in Table 21.1. In addition, Tong [14] has constructed a class of λ-diffuse, t_{ML}-error-correcting, self-orthogonal codes. These codes are easy to implement, have limited error propagation, and their g/b ratio, although much larger than the Gallager bound, is optimum within the class of self-orthogonal diffuse codes. A list of Tong's self-orthogonal diffuse codes with rates $R = 1/2, 2/3, 3/4$, and $4/5$ is given in Table 21.2. Note that the g/b ratio for the diffuse codes in Table 21.1 is much less than for the

TABLE 21.1: Rate $R = 1/2$, λ-diffuse, t_{ML}-error-correcting orthogonalizable codes.

t_{ML}	m	$\lambda^*_{\min}$	$\mathbf{g}^{(1)}$	Orthogonalization rules†
2	$3\lambda+3$	2	$\{0, \lambda, 2\lambda+3, 3\lambda+3\}$	$(3\lambda+3, 2\lambda+1)$
3	$3\lambda+12$	5	$\{0, 1, \lambda+1, 2\lambda+7, 3\lambda+9, 3\lambda+12\}$	$(3\lambda+9, 2\lambda+3)(3\lambda+12, \lambda+4)$
4	$3\lambda+37$	17	$\{0, 2, 3, \lambda+3, 2\lambda+18, 2\lambda+23, 3\lambda+27, 3\lambda+37\}$	$(2\lambda+23, \lambda+8)$ $(3\lambda+27, 2\lambda+12, 2\lambda+7)$ $(3\lambda+37, \lambda+13)$
5	$3\lambda+88$	44	$\{0, 3, 4, 5, \lambda+5, 2\lambda+40, 2\lambda+54, 3\lambda+60, 3\lambda+67, 3\lambda+88\}$	$(5, 1)(2\lambda+54, \lambda+19)(2\lambda+67, \lambda+12)(3\lambda+60, 2\lambda+11, 2\lambda+25)$ $(3\lambda+88, \lambda+33, \lambda+26)$
6	$3\lambda+217$	120	$\{0, 2, 3, 7, 8, \lambda+8, 2\lambda+88, 2\lambda+118, 2\lambda+138, 3\lambda+147, 3\lambda+157, 3\lambda+217\}$	$(8, 4, 5, 6)(2\lambda+118, \lambda+38)$ $(2\lambda+138, \lambda+58, \lambda+28)$ $(3\lambda+147, 2\lambda+17, 2\lambda+67, 2\lambda+37)(3\lambda+157, \lambda+18)$ $(3\lambda+217, \lambda+68, \lambda+78)$
7	$3\lambda+374$	233	$\{0, 6, 7, 9, 10, 11, \lambda+11, 2\lambda+141, 2\lambda+154, 2\lambda+245, 3\lambda+257, 3\lambda+296, 3\lambda+322, 3\lambda+374\}$	$(10, 3, 1)(11, 8, 4)(2\lambda+154, \lambda+24)(2\lambda+245, \lambda+115, \lambda+102)(3\lambda+257, 2\lambda+23, 2\lambda+127, \lambda+114)(3\lambda+296, \lambda+50)(3\lambda+322, \lambda+76, \lambda+37)(3\lambda+374, \lambda+89, \lambda+63)$

Adapted from [13].

*The minimum value of λ for which these codes are λ-diffuse and t_{ML}-error-correcting.

†$(x, y, \cdots)$ indicates that the sum $s_x + s_y + \cdots$ forms an orthogonal check-sum on $e_0^{(0)}$. Only those orthogonal equations that require a sum of syndrome bits are listed.

TABLE 21.2: λ-diffuse, t_{ML}-error-correcting self-orthogonal codes.

t_{ML}	m	λ^*_{min}	$g^{(1)}$
1	3λ	1	$\{0, \lambda, 3\lambda\}$
2	$4\lambda + 1$	2	$\{0, \lambda, 3\lambda, 4\lambda + 1\}$
3	$5\lambda + 4$	4	$\{0, 1, \lambda + 3, 3\lambda + 3, 4\lambda + 3, 5\lambda + 4\}$
4	$6\lambda + 10$	8	$\{0, 1, 3, \lambda + 7, 3\lambda + 7, 4\lambda + 7, 5\lambda + 8, 6\lambda + 10\}$
5	$7\lambda + 19$	13	$\{0, 1, 4, 6, \lambda + 12, 3\lambda + 12, 4\lambda + 12, 5\lambda + 13, 6\lambda + 15, 7\lambda + 19\}$

(a) Rate $R = 1/2$ codes

t_{ML}	m	λ^*_{min}	$g_1^{(2)}$	$g_2^{(2)}$
2	$8\lambda + 3$	3	$\{0, \lambda, 4\lambda, 8\lambda + 3\}$	$\{0, 2\lambda, 6\lambda + 1, 7\lambda + 2\}$
3	$10\lambda + 10$	7	$\{0, 1, \lambda + 2, 4\lambda + 4, 8\lambda + 5, 10\lambda + 10\}$	$\{0, 2, 2\lambda + 2, 6\lambda + 4, 7\lambda + 4, 9\lambda + 7\}$
4	$12\lambda + 26$	12	$\{0, 1, 4, \lambda + 7, 4\lambda + 11, 8\lambda + 12, 9\lambda + 13, 11\lambda + 24\}$	$\{0, 2, 7, 2\lambda + 7, 6\lambda + 11, 7\lambda + 11, 10\lambda + 20, 12\lambda + 26\}$

(b) Rate $R = 2/3$ codes

t_{ML}	m	λ^*_{min}	$g_1^{(3)}$	$g_2^{(3)}$	$g_3^{(3)}$
2	$12\lambda + 5$	6	$\{0, \lambda, 6\lambda, 9\lambda + 2\}$	$\{0, 2\lambda, 11\lambda + 3, 12\lambda + 5\}$	$\{0, 4\lambda, 7\lambda, 8\lambda + 1\}$
3	$15\lambda + 12$	9	$\{0, 1, 2\lambda + 3, 11\lambda + 6, 12\lambda + 7, 15\lambda + 12\}$	$\{0, 2, 4\lambda + 3, 7\lambda + 3, 8\lambda + 5, 13\lambda + 7\}$	$\{0, 3, \lambda, 6\lambda, 9\lambda + 2, 14\lambda + 8\}$

(c) Rate $R = 3/4$ codes

t_{ML}	m	λ^*_{min}	$g_1^{(4)}$	$g_2^{(4)}$	$g_3^{(4)}$	$g_4^{(4)}$
2	$16\lambda + 8$	7	$\{0, \lambda, 10\lambda + 3, 12\lambda + 4\}$	$\{0, 2\lambda, 13\lambda + 5, 14\lambda + 7\}$	$\{0, 3\lambda, 7\lambda, 16\lambda + 8\}$	$\{0, 5\lambda, 8\lambda + 1, 9\lambda + 2\}$

(d) Rate $R = 4/5$ codes

Adapted from [14].

*The minimum value of λ for which these codes are λ-diffuse and t_{ML}-error-correcting.

corresponding rate $R = 1/2$ self-orthogonal diffuse codes in Table 21.2; however, the self-orthogonal diffuse codes are easier to implement and less sensitive to error propagation.

EXAMPLE 21.7 A Rate $R = 1/2$ Self-Orthogonal Diffuse Code

Consider the (2, 1, 9) systematic code with $g^{(2)}(D) = 1 + D^3 + D^7 + D^9$. The parity triangle for this code is given by

$$
\begin{array}{cccccccccccc}
\rightarrow & 1 & & & & & & & & & \\
 & 0 & 1 & & & & & & & & \\
 & 0 & 0 & 1 & & & & & & & \\
\rightarrow & 1 & 0 & 0 & \boxed{1} & & & & & & \\
 & 0 & 1 & 0 & 0 & 1 & & & & & \\
 & 0 & 0 & 1 & 0 & 0 & 1 & & & & \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & & & \\
\rightarrow & 1 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & \boxed{1} & & \\
 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & \\
\rightarrow & 1 & 0 & \boxed{1} & 0 & 0 & 0 & \boxed{1} & 0 & 0 & \boxed{1},
\end{array}
$$

and we see that the code is self-orthogonal with majority-logic error-correcting-capability $t_{ML} = 2$. The four syndrome bits orthogonal on $e_0^{(0)}$ are given by

$$
\begin{array}{lllllll}
s_0 = e_0^{(0)} & & & & & & + e_0^{(1)} \\
s_3 = e_0^{(0)} & & + e_3^{(0)} & & & & + e_3^{(1)} \\
s_7 = e_0^{(0)} & & & + e_4^{(0)} & & + e_7^{(1)} & + e_7^{(1)} \\
s_9 = e_0^{(0)} & + e_2^{(0)} & & & + e_6^{(0)} & + e_9^{(0)} & + e_9^{(1)}.
\end{array}
\tag{21.33}
$$

The error bits other than $e_0^{(0)}$ that belong to a burst of length $b = 4$ or less including $e_0^{(0)}$ can affect only syndrome bit s_0. In addition, error bits from a burst of length $b = 4$ or less that starts after $e_0^{(0)}$ can affect at most two of the syndrome bits orthogonal on $e_0^{(0)}$. Hence, this is a $\lambda = 2$ diffuse, $t_{ML} = 2$ error-correcting code that can correct any two or fewer random errors in a decoding length or any burst of length $b = 4$ or less with a guard space $g = 18$.

21.4.2 Burst-Finding Codes

Consider the $(2, 1, L+M+5)$ systematic convolutional code with $\mathbf{g}^{(1)}(D) = 1 + D^3 + D^4 + D^5 + D^{L+M+5}$. In general, a Gallager burst-finding code corrects "almost all" bursts of length $b = 2(L - 5)$ or less with a guard space $g = 2(L + M + 5)$ as well as t'_{ML} or fewer random errors in a decoding length. In all cases $t'_{ML} < t_{ML}$, and in this example $t'_{ML} = 1$. Typically, L is on the order of hundreds of bits, whereas M is on the order of tens of bits. The encoding circuit for this code is shown in Figure 21.8.

The first five delay units in the encoder along with their associated connections form a set of $J = 4$ orthogonal check-sums on each information error bit (see Example 13.11). By themselves, these orthogonal check-sums could be used to correct $t_{ML} = 2$ or fewer random errors in a decoding length. The key to the Gallager burst-finding code is that only patterns of t'_{ML} or fewer random errors are corrected, where $t'_{ML} < t_{ML}$, and the additional error-correcting capability of the orthogonal check-sums is used to detect bursts.

The decoding circuit for this burst-finding code is shown in Figure 21.9. To understand the functioning of the decoder, assume that the lth block has just been received, that all past decoding estimates have been correct, and that the decoder is in the "random mode" (or r-mode) rather than the "burst mode" (or b-mode).

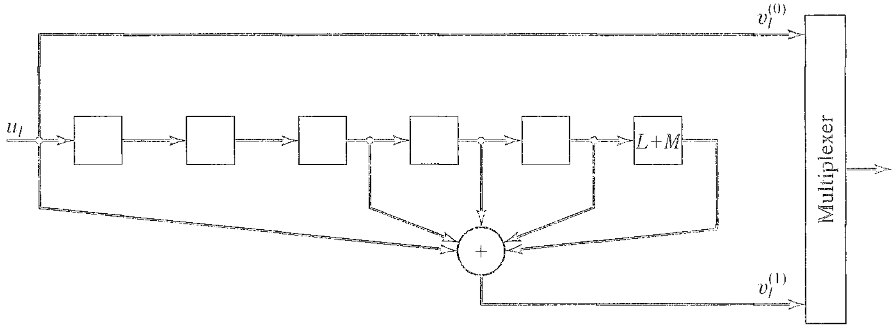
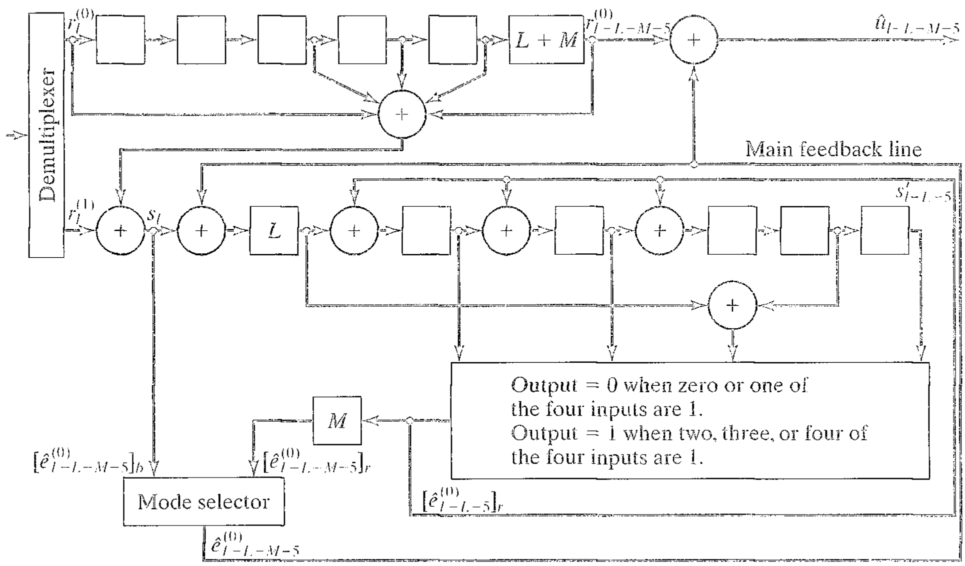


FIGURE 21.8: Encoding circuit for the rate $R = 1/2$ Gallager burst-finding code.



Notes:
1. Mode selector switches to b-mode when 2 or 3 of the 4 check-sums have value 1.
2. Mode selector switches to r-mode when M consecutive r-mode estimates have value 0.

FIGURE 21.9: Decoding circuit for the rate $R = 1/2$ Gallager burst-finding code.

In the r-mode, the input to the mode selector with subscript r is taken as the decoding estimate, whereas in the b-mode, the input with subscript b is taken as the decoding estimate. Because all past estimates have been correct, the feedback of these estimates along the "main feedback line" removes the effect of $e_{l-L-M-5}^{(0)}$ from the syndrome, since at this time $e_{l-L-M-5}^{(0)}$ is the error bit that is fed to the syndrome from the last stage of the encoder replica. Hence, the modified syndrome bits entering the syndrome register are precisely those that would enter if the last $M + L$ stages of the encoder replica and the associated connection were removed.

This modified code is decoded just as if it were a random-error-correcting code, except that the syndrome bits are delayed by L time units before decoding begins.

Now, assume that there are $t'_{ML} = 1$ or fewer errors in the $n_E = 11$ error bits affecting the $J = 4$ orthogonal check-sums on $e^{(0)}_{l-L-5}$. Then, if $e^{(0)}_{l-L-5} = 1$, all four check-sums will equal 1, and if $e^{(0)}_{l-L-5} = 0$, at most one of the check-sums will equal 1. Hence, the estimate $[\hat{e}^{(0)}_{l-L-5}]_r$, which is the decision accepted by the mode selector in the r-mode, will be correct, and the decoder will stay in the r-mode and decode correctly as long as there is at most one error in a decoding length. Note also that there is an additional M time unit delay before the r-mode decision is actually accepted by the mode selector.

Now, suppose that at some time two or three of the four check-sums have value 1. This situation will always occur when there are two or three errors in the $n_E = 11$ error bits affecting the check-sums, and the r-mode estimate is incorrect. (Note that it is possible for two or three errors to cause zero, one, or all four of the check-sums to equal 1, but in this case the estimate will be correct, since two of the errors must have canceled their effect on the check-sums.) When this occurs, the mode selector changes to the b-mode, and $[\hat{e}^{(0)}_{l-L-M-5}]_b$ is now chosen as the estimate. This causes the preceding M decisions of the r-mode decoder to be rejected, which ensures that when a burst occurs the r-mode decoder will have detected the burst before any of its estimates of error bits in the burst are accepted by the mode selector. If the bits in a burst have probability $\frac{1}{2}$ of being in error, then when the first bit in the burst reaches the end of the syndrome register, all $2^4 = 16$ possible outcomes for the four check-sums are equally likely. Because $\binom{4}{2} + \binom{4}{3} = 10$ of these outcomes cause the mode selector to switch to the b-mode, the probability that the burst is not detected in time for the mode selector to switch to the b-mode before accepting an r-mode estimate for any error bit in the burst is

$$\text{Pr (undetected burst)} \approx \left(\frac{6}{16}\right)^{M+1}. \tag{21.34}$$

Clearly, this probability is quite small for M greater than about 10.

The decoding estimate in the b-mode is given by

$$[\hat{e}^{(0)}_{l-L-M-5}]_b = s_l = e^{(0)}_{l-L-M-5} + e^{(0)}_{l-5} + e^{(0)}_{l-4} + e^{(0)}_{l-3} + e^{(0)}_{l} + e^{(1)}_{l}. \tag{21.35}$$

If $e^{(0)}_{l-L-M-5}$ is part of a burst, the other error bits in (21.35) must come from the guard space if the guard space has length $g = 2(L + M + 5)$ or more. Hence, the decoding estimate in the b-mode will be correct. Note that it is possible for a burst to cause a switch to the b-mode as soon as its first bit reaches the first input to a check-sum in the syndrome register, which would cause up to $M + 5$ guard-space error bits to be estimated in the b-mode. For these estimates to be correct, the burst length should not exceed $L - 5$ blocks, or $b = 2(L - 5)$ bits, since otherwise, error bits from the burst would affect the decoding of these guard-space error bits.

While the decoder is in the b-mode, the r-mode estimates continue to be monitored as an indicator of when the decoder should switch back to the r-mode. When M consecutive r-mode estimates have value 0, this is taken as evidence that the most recent $L+M+5$ received blocks have been in a guard space, and the mode selector is returned to the r-mode. Because, during a burst, only $\binom{4}{0}+\binom{4}{1}=5$ of the $2^4=16$ possible outcomes for the four check-sums give a value of 0 for the r-mode estimate, the probability of a false return to the r-mode during a burst is

$$\Pr(\text{false return to } r\text{-mode}) \approx \left(\frac{5}{16}\right)^M. \tag{21.36}$$

Again, this probability is quite small for M greater than about 10.

The adaptive decoding scheme just described corrects "almost all" bursts of length $b=2(L-5)$ with a guard space $g=2(L+M+5)$, since the probabilities of an undetected burst and of a false return to the random mode, given by (21.34) and (21.36), respectively, can be made quite small by choosing M large enough. Because L can be chosen much greater than M, we see that

$$\frac{g}{b}=\frac{2(L+M+5)}{2(L-5)} \approx 1, \tag{21.37}$$

which meets the lower bound of (21.2) on "almost-all" burst-error correction for rate $R=1/2$ codes.

Gallager originally described his burst-finding codes only for rate $R=(n-1)/n$, but Reddy [15] generalized this to any rate $R=k/n$. All these codes have the property that the lower bound of (21.2) on "almost all" burst-error correction is met with near equality. Note that (21.35) implies that when the decoder is in the b-mode, it is sensitive to errors in the guard space. Sullivan [16] has shown how burst-finding codes can be modified to provide some protection against errors in the guard space by lowering the code rate.

21.4.3 Burst-Trapping Codes

Tong's burst-trapping codes are similar to Gallager's burst-finding codes, except that they are based on block codes rather than convolutional codes. The overall code remains convolutional, however, since there is memory in the encoder.

Consider an $(n=3M, k=2M)$ systematic block code with rate $R=2/3$, error-correcting capability t, and generator matrix

$$\mathbb{G}=\left[\begin{array}{c|c} \mathbb{I}_{2M} & \begin{array}{c}\mathbb{G}_1\\ \hline \mathbb{G}_2\end{array}\end{array}\right], \tag{21.38}$$

where $\mathbb{G}_1$ and $\mathbb{G}_2$ are $M\times M$ submatrices of $\mathbb{G}$. The codeword $\mathbf{v}=[\mathbf{u}^{(1)},\mathbf{u}^{(2)}]\mathbb{G}=[\mathbf{u}^{(1)},\mathbf{u}^{(2)},\mathbf{p}]$, where $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ contain M information bits each, and $\mathbf{p}$ is the M-bit parity vector given by

$$\mathbf{p}=\mathbf{u}^{(1)}\mathbb{G}_1+\mathbf{u}^{(2)}\mathbb{G}_2. \tag{21.39}$$

The encoder for this Tong burst-trapping code is shown in Figure 21.10. Note that, in addition to the operations required by the block encoder, memory has been added to the encoder. This converts the block code into an $(n=3M, k=2M, m=2L)$

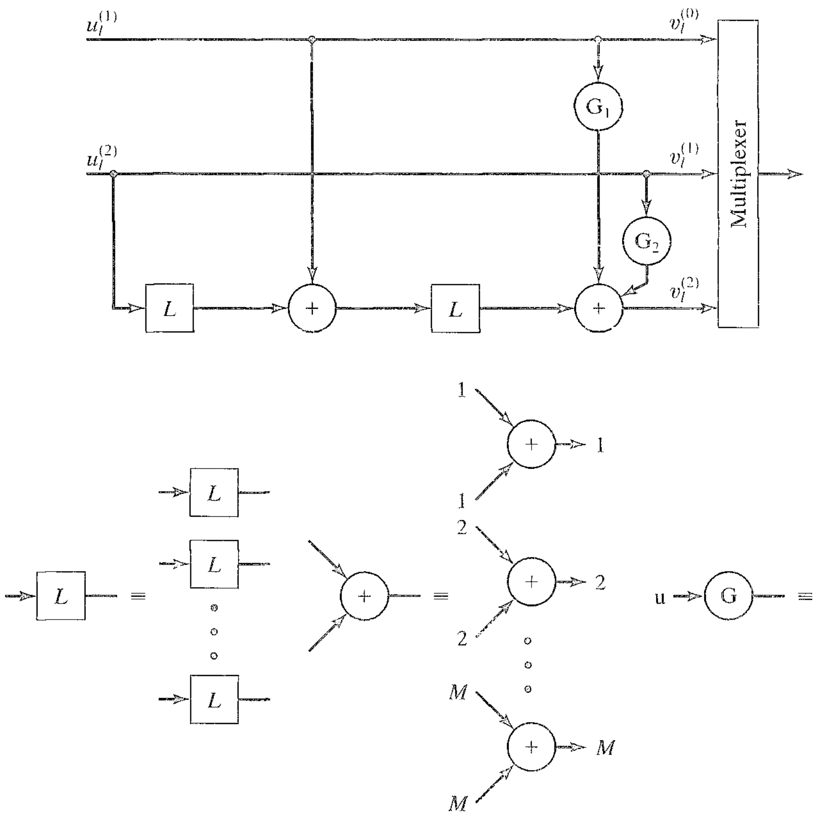


FIGURE 21.10: Encoding circuit for the rate $R = 2/3$ Tong burst-trapping code.

convolutional code. The code is still systematic, and the encoding equations at time unit l are given by

$$\begin{aligned}
\mathbf{v}_l^{(0)} &= \mathbf{u}_l^{(1)} \\
\mathbf{v}_l^{(1)} &= \mathbf{u}_l^{(2)} \\
\mathbf{v}_l^{(2)} &= \mathbf{u}_l^{(1)}\mathbf{G}_1 + \mathbf{u}_l^{(2)}\mathbf{G}_2 + \mathbf{u}_{l-L}^{(1)} + \mathbf{u}_{l-2L}^{(2)}.
\end{aligned} \tag{21.40}$$

The decoding circuit for the Tong burst-trapping code is shown in Figure 21.11. Assume that the lth block has just been received, that all past decoding decisions have been correct, and that the decoder is in the r-mode. The feedback of the decoding estimates $\hat{\mathbf{u}}_{l-L}^{(1)}$ and $\hat{\mathbf{u}}_{l-2L}^{(2)}$ to the parity input line of the block decoder removes the effect of past decisions from the encoding equations, so that the block decoder simply decodes the original block code; however, the block decoder is designed to correct only t' or fewer random errors, where $t' < t$, and the additional error-correcting capability of the code is used to detect patterns of $t' + 1$ or more errors. As long as the block decoder estimates t' or fewer errors in a block it remains

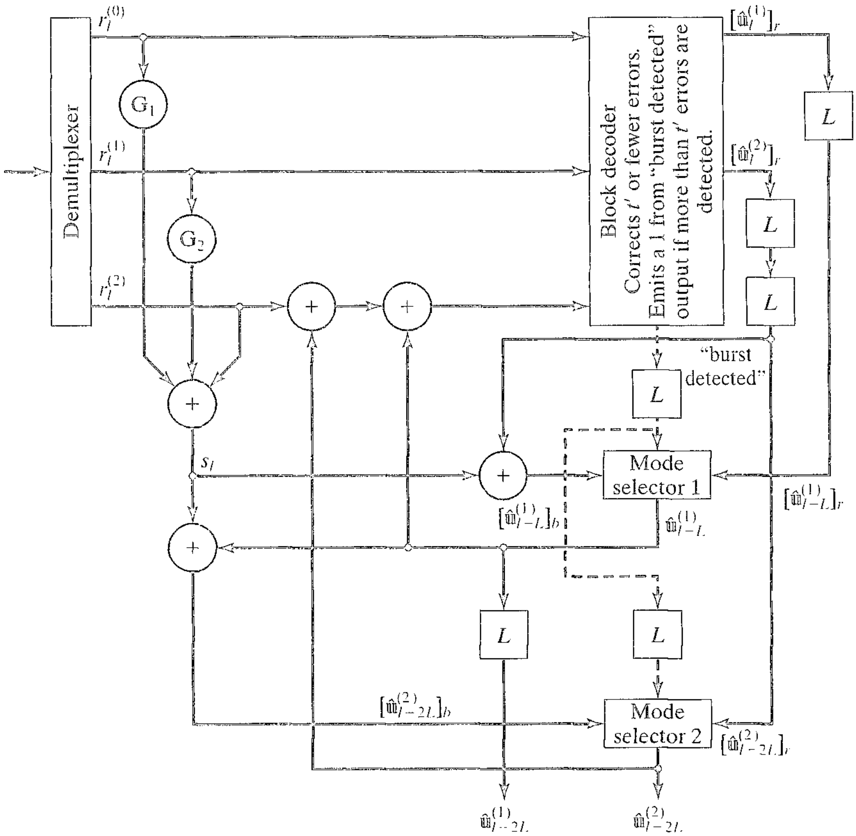


FIGURE 21.11: Decoding circuit for the rate $R = 2/3$ Tong burst-trapping code.

in the r-mode. Note that, just as for burst-finding codes, the decoding estimates of the block decoder are delayed by $2L$ time units before they are accepted as final.

When the block decoder detects a pattern of $t' + 1$ or more errors, say at time unit $l - L$, it emits a single 1 from its "burst detected" output. L time units later, at time unit l, this 1 reaches the first mode selector in Figure 21.11 and causes it to switch to the b-mode for that block only. At this time the output of the first mode selector is

$$\begin{aligned}\hat{\mathbf{u}}^{(1)}_{l-L} &= [\hat{\mathbf{u}}^{(1)}_{l-L}]_b = \mathbf{s}_l + [\hat{\mathbf{u}}^{(2)}_{l-2L}]_r \\ &= (\mathbf{u}^{(1)}_l + \mathbf{e}^{(0)}_l)\mathbb{G}_1 + (\mathbf{u}^{(2)}_l + \mathbf{e}^{(1)}_l)\mathbb{G}_2 + \mathbf{u}^{(1)}_l\mathbb{G}_1 + \mathbf{u}^{(2)}_l\mathbb{G}_2 \\ &\quad + \mathbf{u}^{(1)}_{l-L} + \mathbf{u}^{(2)}_{l-2L} + \mathbf{e}^{(2)}_l + [\mathbf{u}^{(2)}_{l-2L}]_r \\ &= \mathbf{u}^{(1)}_{l-L} + \mathbf{u}^{(2)}_{l-2L} + [\hat{\mathbf{u}}^{(2)}_{l-2L}]_r + \mathbf{e}^{(0)}_l\mathbb{G}_1 + \mathbf{e}^{(1)}_l\mathbb{G}_2 + \mathbf{e}^{(2)}_l.\end{aligned} \tag{21.41}$$

From our earlier assumption that past decoding decisions are all correct, $[\hat{\mathbf{u}}^{(2)}_{l-2L}]_r = \mathbf{u}^{(2)}_{l-2L}$. Also, assuming that time unit l comes from the error-free guard space following the burst at time unit $l - L$, it follows that

$$\hat{\mathbf{u}}_{l-L}^{(1)} = \mathbf{u}_{l-L}^{(1)}, \tag{21.42}$$

and the b-mode decoding estimate will be correct. A similar argument shows that at time unit $l + L$, when the "burst detected" output reaches the second mode selector and causes it to switch to the b-mode for that block only, the output of the second mode selector will be the correct estimate for $\mathbf{u}_{l-L}^{(2)}$, provided that time unit $l + L$ also comes from the error-free guard space.

The probability of failing to switch to the b-mode during a burst can be estimated as follows. Assume that an (n, k) block code is designed to correct t' or fewer errors. There are then a total of $N(t') \triangleq \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t'}$ correctable error patterns. Hence, for each of the 2^k codewords, there are $N(t')$ received blocks that will be corrected and not cause a switch to the b-mode. Assuming that during a burst all 2^n received blocks are equally likely, the probability of failing to switch to the b-mode during a burst is given by

$$\Pr(\text{undetected burst}) \approx N(t')\frac{2^k}{2^n} = N(t')/2^{n-k}. \tag{21.43}$$

EXAMPLE 21.8 Burst-Error-Detection Failure Rate

Consider a (30, 20) shortened BCH code with $t' = 1$. Then, $N(t') = 1 + 30 = 31$, and

$$\Pr(\text{undetected burst}) \approx 31/2^{10} = 3.0 \times 10^{-2}, \tag{21.44}$$

which implies about a 3% failure rate in detecting bursts.

The burst-trapping code just described corrects "almost all" bursts that can affect at most L consecutive received blocks. Hence, $b = (n - 1)L + 1$ bits. The guard space must include at least $2L$ consecutive error-free blocks following the burst. Hence, $g = 2nL + (n - 1)$. Therefore, for large L and n,

$$\frac{g}{b} = \frac{2nL + (n-1)}{(n-1)L + 1} \approx 2, \tag{21.45}$$

which meets the lower bound of (21.2) on "almost all" burst-error correction for rate $R = 2/3$ codes.

Although the preceding discussion concerned only rate $R = 2/3$ codes, burst-trapping codes can easily be generalized to any rate $R = \lambda M/(\lambda + 1)M$. These systems all meet the lower bound of (21.2) on "almost all" burst-error correction with near equality. Like the closely related burst-finding codes, burst-trapping codes are sensitive to errors in the guard space when the decoder is in the b-mode. Burton et al. [17] have shown, however, that the system can be modified to provide some protection against errors in the guard space by lowering the code rate.

PROBLEMS

21.1 Using mathematical induction, show that the unknown elements of the matrix $\mathbb{B}_0$ can always be chosen so that (21.11) is satisfied.

21.2 Show how to construct optimum phased-burst-error-correcting Berlekamp–Preparata codes with $k < n - 1$.

21.3 Consider the Berlekamp–Preparata code with $n = 3$.

a. Find m, b, and g for this code.

b. Find the $\mathbb{B}_0$ matrix.

c. Find the generator polynomials $\mathbf{g}_1^{(2)}(D)$ and $\mathbf{g}_2^{(2)}(D)$.

d. Find the $\mathbb{H}_0$ matrix.

e. Draw the complete encoder/decoder block diagram for this code.

21.4 Consider the Iwadare–Massey code with $n = 2$ and $\lambda = 4$.

a. Find m, b, and g for this code.

b. Find the generator polynomial $\mathbf{g}^{(1)}(D)$.

c. Find the repeat distance of the information error bit $e_l^{(0)}$.

d. Draw the complete encoder/decoder block diagram for this code.

21.5 A second class of Iwadare–Massey codes exists with the following parameters:

$$m = (2n - 1)\lambda + (n^2 - n - 2)/2$$

$$b = n\lambda$$

$$g = n(m + 1) - 1$$

The $n - 1$ generator polynomials are given by (21.20), where $a(i) \triangleq \frac{1}{2}(n - i)(4\lambda + n - i - 3) + n - 1$, and $b(i) \triangleq \frac{1}{2}(n - i)(4\lambda + n - i - 1) + n + \lambda - 2$. Consider the code with $n = 3$ and $\lambda = 3$.

a. Find m, b, and g for this code.

b. Find the generator polynomials $\mathbf{g}_1^{(2)}(D)$ and $\mathbf{g}_2^{(2)}(D)$.

c. Find the repeat distance of the information error bits $e_l^{(0)}$ and $e_l^{(1)}$.

d. Construct a decoding circuit for this code.

21.6 Construct a general decoding circuit for the class of Iwadare–Massey codes in Problem 21.5. For the two classes of Iwadare–Massey codes:

a. compare the excess guard space required beyond the Gallager bound; and

b. compare the number of register stages required to implement a general decoder.

21.7 Show that for the Iwadare–Massey code of Example 21.4, if $n[m+(\lambda+2)n-1]-1 = 95$ consecutive error-free bits follow a decoding error, the syndrome will return to the all-zero state.

21.8 Consider the $(2, 1, 5)$ double-error-correcting orthogonalizable code from Table 13.3 interleaved to degree $\lambda = 7$.

a. Completely characterize the multiple-burst-error-correcting capability and the associated guard-space requirements of this interleaved code.

b. Find the maximum single-burst length that can be corrected and the associated guard space.

c. Find the ratio of guard space to burst length for (b).

d. Find the total memory required in the interleaved decoder.

e. Draw a block diagram of the complete interleaved system.

21.9 Consider the interleaved encoder shown in Figure 21.6(b). Assume that an information sequence $u_0, u_1, u_2, \cdots$ enters the encoder. Write down the string of encoded bits and verify that an interleaving degree of $\lambda = 5$ is achieved.

21.10 Consider the Berlekamp–Preparata code of Problem 21.3 interleaved to degree $\lambda = 7$.
  a. Find the g/b ratio and compare it with the Gallager bound.
  b. Draw a block diagram of the complete interleaved system.

21.11 Consider the $n = 3$ Berlekamp–Preparata code interleaved to degree $\lambda = 7$ and the $n = 3$ Iwadare–Massey code with $\lambda = 7$.
  a. Compare the g/b ratios of the two codes.
  b. Compare the number of register stages required to implement the decoder in both cases.

21.12 Consider the (2, 1, 9) systematic code with $\mathbf{g}^{(1)}(D) = 1 + D^2 + D^5 + D^9$.
  a. Is this code self-orthogonal? What is t_{ML} for this code?
  b. Is this a diffuse code? What is the burst-error-correcting capability b and the required guard space g?
  c. Draw a complete encoder/decoder block diagram for this code.

21.13 For the diffuse code of Figure 21.7, find the minimum number of error-free bits that must be received following a decoding error to guarantee that the syndrome returns to the all-zero state.

21.14 Consider using the (2, 1, 11) triple-error-correcting orthogonalizable code from Table 13.3 in the Gallager burst-finding system.
  a. Draw a block diagram of the encoder.
  b. Draw a block diagram of the decoder.
  c. With $t'_{ML} = 1$, choose M and L such that the probabilities of an undetected burst and of a false return to the r-mode are less than 10^{-2} and the g/b ratio is within 1% of the bound on "almost all" burst-error correction for rate $R = 1/2$ codes.
  d. Repeat (c) for $t'_{ML} = 2$.

21.15 Consider the rate $R = 2/3$ burst-trapping code of Example 21.8.
  a. Choose L such that the g/b ratio is within 1% of the bound on "almost all" burst-error correction for rate $R = 2/3$ codes.
  b. Describe the generator matrix $\mathbb{G}$ of the (30, 20, 2L) convolutional code.

BIBLIOGRAPHY

1. D. W. Hagelbarger, "Recurrent Codes: Easily Mechanized Burst-Correcting, Binary Codes," *Bell System Tech. J.*, 38: 969–84, July 1959.

2. Y. Iwadare, "On Type B1 Burst-Error-Correcting Convolutional Codes," *IEEE Trans. Inform. Theory*, IT-14: 577–83, July 1968.

3. R. G. Gallager, *Information Theory and Reliable Communication*. McGraw-Hill, New York, 1968.

4. A. D. Wyner and R. B. Ash, "Analysis of Recurrent Codes," *IEEE Trans. Inform. Theory*, IT-9: 143–56, July 1963.

5. E. R. Berlekamp, "Note on Recurrent Codes," *IEEE Trans. Inform. Theory*, IT-10: 257–58, July 1964.

6. F. P. Preparata, "Systematic Construction of Optimal Linear Recurrent Codes for Burst Error Correction," *Calcolo*, 2: 1–7, 1964.

7. A. Kohlenberg and G. D. Forney, Jr., "Convolutional Coding for Channels with Memory," *IEEE Trans. Inform. Theory*, IT-14: 618–26, September 1968.

8. J. L. Massey, "Advances in Threshold Decoding," in *Advances in Communication Systems*, vol. 2, edited by A. V. Balakrishnan, Academic Press, New York, 1968.

9. S. Y. Tong, "Burst-Trapping Techniques for a Compound Channel," *IEEE Trans. Inform. Theory*, IT-15: 710–15, November 1969.

10. J. L. Massey, "Coding for Everyday Channels," in *Coding Techniques for Digital Communication*, notes for a tutorial session, IEEE International Conference on Communications, Seattle, Wash., June 1973.

11. J. L. Massey, "Implementation of Burst-Correcting Convolutional Codes," *IEEE Trans. Inform. Theory*, IT-11: 416–22, July 1965.

12. M. J. Ferguson, "Diffuse Threshold Decodable Rate 1/2 Convolutional Codes," *IEEE Trans. Inform. Theory*, IT-17: 171–80, March 1971.

13. S. Y. Tong, "Construction of Asymptotically Optimal Rate 1/2 Diffuse Codes," in *Proc. IEEE Intl. Conf. Commun. (ICC 70)*, San Francisco, Calif., June 1970.

14. S. Y. Tong, "Systematic Construction of Self-Orthogonal Diffuse Codes," *IEEE Trans. Inform. Theory*, IT-16: 594–604, September 1970.

15. S. M. Reddy, "Linear Convolutional Codes for Compound Channels," *Inform. Control*, 19: 387–400, December 1971.

16. D. D. Sullivan, "A Generalization of Gallager's Adaptive Error Control Scheme," *IEEE Trans. Inform. Theory*, IT-17: 727–35, November 1971.

17. H. O. Burton, D. D. Sullivan, and S. Y. Tong, "Generalized Burst-Trapping Codes," *IEEE Trans. Inform. Theory*, IT-17: 736–42, November 1971.