CHAPTER 15

Concatenated Coding, Code Decomposition, and Multistage Decoding

Concatenated coding, devised by Forney in 1966 [1], is another powerful technique for constructing long powerful codes from short component codes, besides product coding, $|u|u + v|$-construction. Single-level concatenation using a nonbinary code as an outer code and a binary code as an inner code is widely used in both communication and digital data storage systems to achieve high reliability with reduced decoding complexity. The inner code is generally short and is decoded with a soft-decision decoding algorithm, and the nonbinary outer code is generally longer and decoded with an algebraic decoding method. In most applications, RS codes are used as the outer codes. Concatenation can also be multilevel, with multiple outer codes concatenated with multiple inner codes. Multilevel concatenation provides more flexibility for constructing good codes and in designing error-control systems with different rates for various communication environments. Using multilevel concatenation, many good codes have been constructed from very simple short inner and outer codes. The opposite of the construction of long multilevel concatenated codes from short inner and outer codes is the decomposition of existing long codes as multilevel concatenated codes for which the component codes are short and simple.

Multilevel concatenated codes and decomposable codes can be decoded in multiple stages to achieve good error performance with reduced decoding complexity. In multistage decoding, component codes are decoded one at a time in a sequence of decoding stages. The decoded information at one stage is passed to the next stage for decoding the next component code. Because the component codes are short and simple, they can be decoded with soft-decision decoding to achieve good error performance. Properly devised multistage decoding can achieve suboptimum or near-optimum error performance with a significant reduction in decoding complexity.

In this chapter we first present the single-level and multilevel concatenations with some examples of code construction and concatenated coding systems for error control. Next, we show that RM codes can be decomposed as multilevel concatenated codes. Then, we present two soft-decision multistage decoding methods, one suboptimum and the other optimum.

15.1 SINGLE-LEVEL CONCATENATED CODES

A simple concatenated code is formed from two codes: an (n_1, k_1) binary code C_1 and an (n_2, k_2) nonbinary code C_2 with symbols from $GF(2^{k_1})$. The symbols of C_2 are represented by their corresponding bytes of k_1 binary symbols (or k_1-tuples).

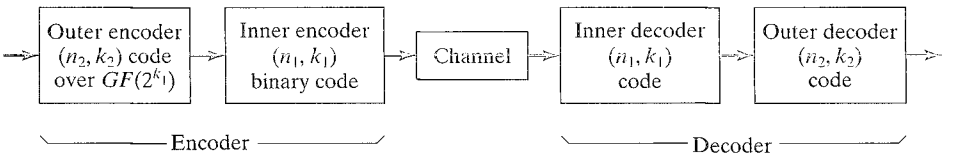

FIGURE 15.1: Communication system using a concatenated code.

Usually, an RS code is used for C_2. Encoding consists of two steps, as shown in Figure 15.1. First, the k_1k_2 binary information digits are divided into k_2 bytes of k_1 information digits each. These k_2 bytes are encoded according to the rules for C_2 to form an n_2-byte codeword. Second, each k_1-digit byte is encoded into a codeword in C_1, resulting in a string of n_2 codewords of C_1, a total of n_2n_1 digits. These digits are then transmitted, one C_1 codeword at a time, in succession. Thus, the resultant code is an (n_1n_2, k_1k_2) binary linear code. The component codes C_1 and C_2 are called the *inner* and *outer* codes, respectively. If the minimum distances of the inner and outer codes are d_1 and d_2, respectively, the minimum distance of their concatenation is at least d_1d_2 (see Problem 15.1). This type of concatenation of two codes is known as *one-level concatenation.*

The concatenated code of C_1 and C_2 is also decoded in two steps, as shown in Figure 15.1. First, each C_1 codeword is decoded as it arrives, and the check digits are removed, leaving a sequence of n_2 k_1-digit bytes. These bytes are then decoded according to the decoding method for C_2, to leave the final corrected message. Decoding implementation is the straightforward combination of the implementations for codes C_1 and C_2, and the amount of hardware required is roughly that required by both codes.

Concatenated codes are effective against a mixture of random errors and bursts, and the pattern of bytes not correctable by the C_1 code must form a correctable error pattern for C_2 if the concatenated code is to correct the error pattern. Scattered random errors are corrected by C_1. Burst errors may affect relatively few bytes, but probably so badly that C_1 cannot correct them. These few bytes can then be corrected by C_2.

EXAMPLE 15.1

Consider the concatenation of the (15, 11) RS code with symbols from $GF(2^4)$ and the (7, 4) binary Hamming code. Each code symbol of the RS code is represented by a byte of four binary digits, as in Table 2.8. Then, each 4-bit byte is encoded into a codeword in the (7, 4) Hamming code. The resultant concatenated code is a (105, 44) binary code. Because the minimum distance of the (7, 4) Hamming code is 3, and the minimum distance of the (15, 11) RS code is 5, the concatenated code has a minimum distance of at least 15. If the code is decoded in two steps, first the inner code and then the outer code, the decoder is capable of correcting any error pattern such that the number of inner codewords with more than a single error is less than 3.

A simple generalization of the concatenated coding is to interleave the outer code. Let the outer code C_2 be an (n_2, k_2) linear block code with symbols from

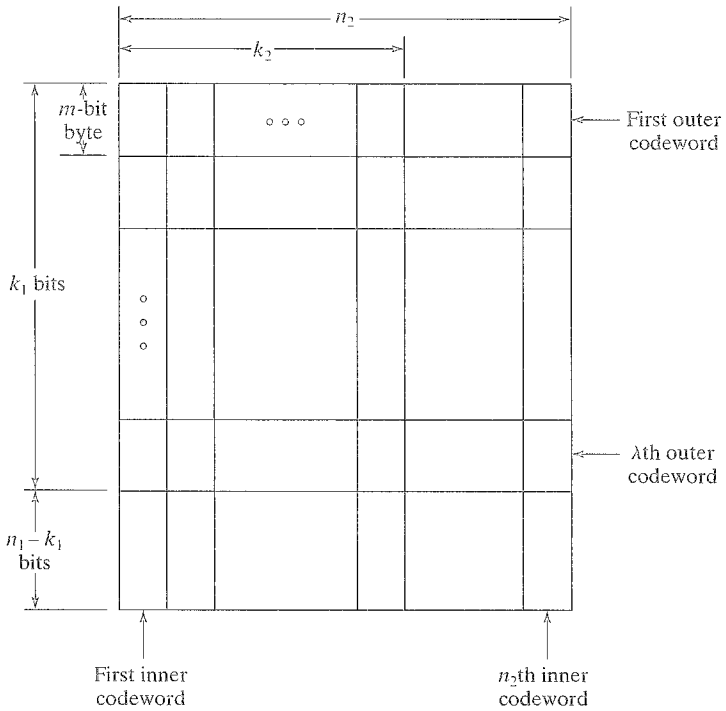

FIGURE 15.2: An interleaved code array.

$GF(2^m)$, and the inner code C_1 be an (n_1, k_1) binary linear code with $k_1 = \lambda m$, where λ is a positive integer. A message of k_2 m-bit bytes (or $k_2 m$ bits) is first encoded into an n_2-byte codeword in C_2. This codeword is then temporarily stored in a buffer as a row in an array, as shown in Figure 15.2. After λ outer codewords have been formed, the buffer stores a $\lambda \times n_2$ array. Each column of the array consists of λ (m-bit) bytes (or λm bits) and is encoded into an n_1-bit codeword in C_1. Each encoded column is transmitted serially. Therefore, the outer code is interleaved by a depth of λ. When an inner codeword with possible errors is received, it is decoded and the parity bits are removed. The decoded λ bytes (or λm bits) are stored in a receiver buffer as a column in a $\lambda \times n_2$ array. After n_2 inner code decodings, the receiver contains a $\lambda \times n_2$ decoded array. Then, each row of this array is decoded based on the outer code C_2. Therefore, there are a total of λ outer code decodings. The overall interleaved concatenated coding system is shown in Figure 15.3.

It is possible to construct a concatenated code from a single outer code and *multiple* inner codes. For example, one may use an (n_2, k_2) code C_2 with symbols from $GF(2^{k_1})$ as the outer code and n_2 (n_1, k_1) binary codes, $C_1^{(1)}, C_1^{(2)}, \cdots, C_1^{(n_2)}$, as inner codes. Again, the encoding consists of two steps. First, the $k_1 k_2$ information digits are divided into k_2-bytes of k_1-digits each. These k_2-bytes are encoded according to the rules for C_2 to form an n_2-byte codeword, $(a_0, a_1, \cdots, a_{n_2-1})$, where each byte a_i is regraded as an element in $GF(2^{k_1})$. Second, the ith byte a_i for $0 \leq i < n_2$ is encoded into a codeword in the ith inner code $C_1^{(i)}$. The overall encoding again

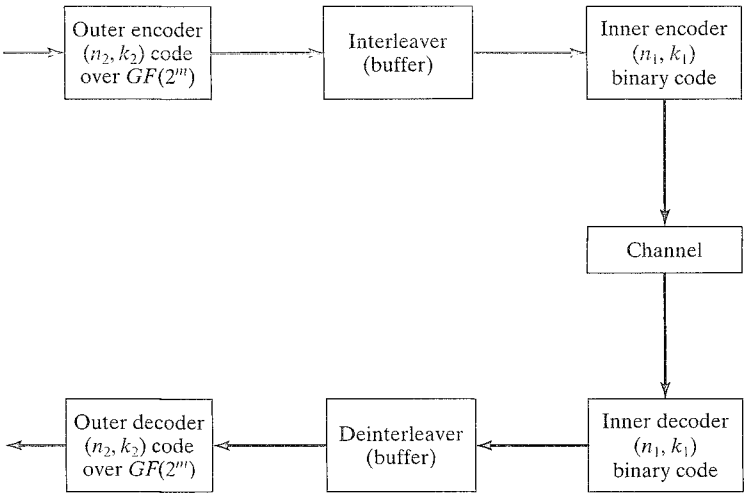

FIGURE 15.3: An interleaved concatenated coding system.

results in an $(n_1 n_2, k_1 k_2)$ concatenated code. By concatenating one outer code with multiple inner codes, Justesen [2] was able to construct a class of asymptotically good concatenated codes.

Concatenated coding has been widely used in digital communication and storage systems to achieve high reliability (low bit-error probability) with reduced decoding complexity. The inner code is usually short and is decoded with soft-decision decoding. An RS code is commonly used as the outer code and is decoded with hard-decision decoding using an algorithm presented in Chapter 7 (the Euclidean algorithm is the most popular one).

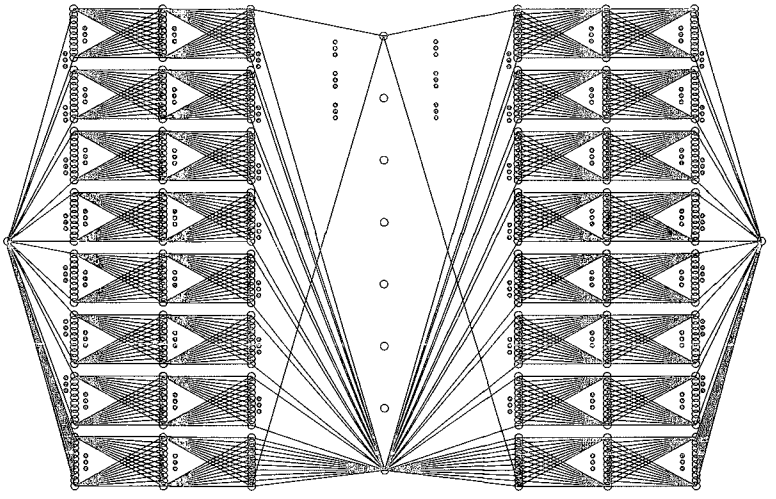

FIGURE 15.4: A subtrellis of the 8-section trellis for the (64, 40) RM subcode.

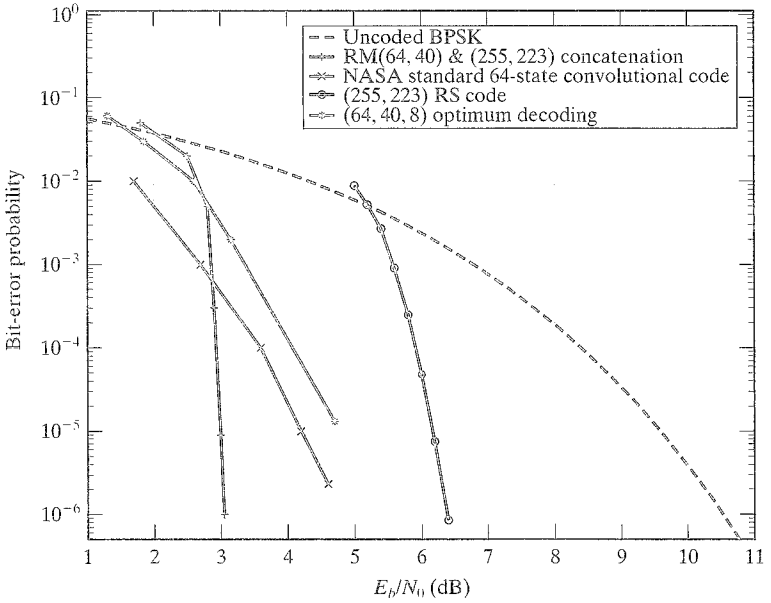

FIGURE 15.5: The bit-error performances of the (64, 40, 8) RM subcode and the concatenated coding scheme with the (64, 40, 8) RM subcode as the inner code and the NASA standard (255, 223, 33) RS code as the outer code.

As an example, we consider a concatenated error-control scheme in which the outer code is the (255, 223, 33) RS code with symbols from $GF(2^8)$, and the inner code is a binary (64, 40) linear code with a minimum distance of 8. The outer code has a minimum distance of 33 and is capable of correcting 16 symbol errors. The generator matrix of the inner code is obtained by removing two rows from the generator matrix of the (64, 42, 8) RM code. The trellis of this inner code consists of 32 parallel and structurally identical subtrellises, each with 64 states. One such subtrellis is shown in Figure 15.4. The parallel structure allows us to devise 32 identical subdecoders to process the subtrellises in parallel to achieve high decoding speed. In concatenation with this (64, 40) inner code, the (255, 223) RS outer code is interleaved to a depth of 5. The overall rate of the concatenated coding system is 0.545, and its bit-error performance is shown in Figure 15.5. It achieves a BER of 10^{-6} at 3.1 dB SNR.

15.2 MULTILEVEL CONCATENATED CODES

Single-level concatenation can be generalized to multilevel concatenation [5, 6]. In a multilevel concatenated coding system, multiple inner and outer codes are used to form multiple concatenated codes, and these concatenated codes are then combined to form an overall concatenated code. Multilevel concatenation provides an even more powerful coding technique for error control than single-level concatenation and offers more flexibility in designing error-control systems for various communication environments. Furthermore, the multilevel structure allows the use of multistage decoding to reduce decoding complexity.

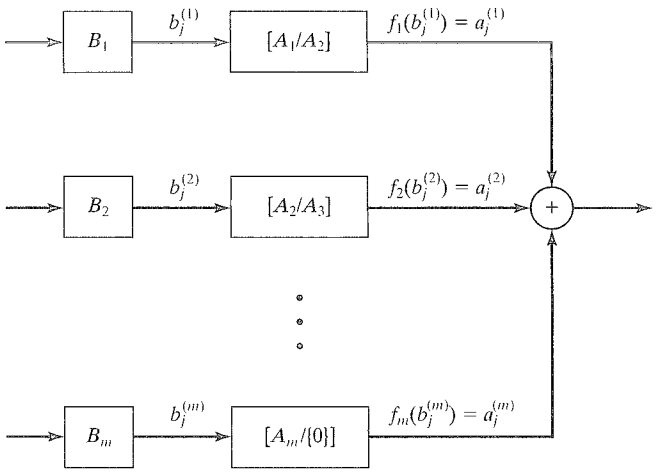

FIGURE 15.6: An encoder for an m-level concatenated code.

An m-level concatenated code is formed from a set of m inner codes and a set of m outer codes, as shown in Figure 15.6. The m inner codes are coset codes formed from a binary (n, k) linear block code A_1 and a sequence of m linear subcodes of A_1, denoted by $A_2, A_3, \ldots, A_{m+1} = \{\mathbb{0}\}$, such that

$$A_1 \supset A_2 \supset \ldots \supset A_m \supset A_{m+1} = \{\mathbb{0}\}, \tag{15.1}$$

where A_{m+1} contains only the all-zero codeword $\mathbb{0}$. For $1 \leq i \leq m$, let

1. k_i and d_{A_i} be the dimension and minimum distance of A_i, respectively.
2. $[A_i/A_{i+1}]$ denote the set of representatives of cosets in the partition A_i/A_{i+1}.

$[A_i/A_{i+1}]$ is a linear code with $2^{k_i - k_{i+1}}$ codewords, and hence its dimension is $k_i - k_{i+1}$. This code is called a *coset code*.

For $1 \leq i \leq m$, let

$$q_i \stackrel{\triangle}{=} |[A_i/A_{i+1}]| = 2^{k_i - k_{i+1}}, \tag{15.2}$$

and

$$[A_i/A_{i+1}] = \{\mathbb{a}_j^{(i)} : 1 \leq j \leq q_i\}. \tag{15.3}$$

Then, a coset in A_i/A_{i+1} is given by

$$\{\mathbb{a}_j^{(i)} + \mathbb{a} : \mathbb{a} \in A_{i+1}\},$$

and

$$\begin{aligned} A_i &= \bigcup_{j=1}^{q_i} (\mathbb{a}_j^{(i)} + A_{i+1}) \\ &= [A_i/A_{i+1}] \oplus A_{i+1}. \end{aligned} \tag{15.4}$$

From (15.4) we readily see that

$$A_1 = [A_1/A_2] \oplus [A_2/A_3] \oplus \ldots \oplus A_m. \tag{15.5}$$

Therefore, each codeword in A_1 is a sum of m coset representatives from the codes $[A_1/A_2], [A_2/A_3], \ldots, [A_m/\{\mathbb{0}\}] = A_m$, respectively. These m coset codes are used as the inner codes for an m-level concatenated codes, as shown in Figure 15.6.

For $1 \leq i \leq m$, the ith outer code, denoted by B_i, is an (N, K_i) linear block code over $GF(q_i)$ with a minimum distance of d_{B_i}, where q_i is defined by (15.2). In the m-level concatenation coding scheme shown in Figure 15.6, the ith level concatenated code has B_i as the outer code and $[A_i/A_{i+1}]$ as the inner code. We denote this concatenated code with

$$B_i \circ [A_i/A_{i+1}].$$

During each encoding interval, a message of K_i symbols over $GF(q_i)$ is encoded into a codeword

$$\mathbb{b}^{(i)} = (b_0^{(i)}, b_1^{(i)}, \ldots, b_{N-1}^{(i)})$$

in the ith outer code B_i for $1 \leq i \leq m$. Then, each symbol $b_j^{(i)}$ of $\mathbb{b}^{(i)}$ is encoded into a codeword $f_i(b_j^{(i)})$ in the ith inner code $[A_i/A_{i+1}]$, where $f_i(\cdot)$ denotes the encoding mapping of the ith inner code encoder. The output of the ith inner code encoder is the following sequence:

$$\mathbb{c}^{(i)} = (f_i(b_0^{(i)}), f_i(b_1^{(i)}), \ldots, f_i(b_{N-1}^{(i)})),$$

which is a codeword in the ith level concatenated code $B_i \circ [A_i/A_{i+1}]$ and is a sequence of coset representatives from $[A_i/A_{i+1}]$. Consequently, the output sequence of the overall encoder of the m-level concatenated coding system shown in Figure 15.6 is the following sum:

$$\begin{aligned} \mathbb{c} &= (\mathbb{c}_0, \mathbb{c}_1, \ldots, \mathbb{c}_{N-1}) \\ &= \mathbb{c}^{(1)} + \mathbb{c}^{(2)} + \ldots + \mathbb{c}^{(m)}, \end{aligned} \tag{15.6}$$

where for $0 \leq j < N$,

$$\mathbb{c}_j = \sum_{i=1}^{m} f_i(b_j^{(i)}). \tag{15.7}$$

From (15.7) we readily see that each component of $\mathbb{c}$ is a codeword in A_1. Therefore, $\mathbb{c}$ is a sequence of N codewords in A_1.

The following collection of sequences,

$$C = \{(\mathbb{c}_0, \mathbb{c}_1, \ldots, \mathbb{c}_{N-1}) : \mathbb{c}_j = \sum_{i=1}^{m} f_i(b_j^{(i)}) \text{ and } (b_0^{(i)}, b_1^{(i)}, \ldots, b_{N-1}^{(i)}) \in B_i \text{ for } 1 \leq i \leq m \text{ and } 0 \leq j < N\}, \tag{15.8}$$

forms an m-level concatenated code. For simplicity, we denote this concatenated code with

$$C \triangleq \{B_1, B_2, \ldots, B_m\} \circ \{A_1, A_2, \ldots, A_m\}. \tag{15.9}$$

Note that for $1 \leq i, j \leq m$, and $i \neq j$, $[A_i/A_{i+1}]$ and $[A_j/A_{j+1}]$ have only the all-zero codeword $\mathbb{0}$ in common; that is,

$$[A_i/A_{i+1}] \bigcap [A_j/A_{j+1}] = \{\mathbb{0}\}. \tag{15.10}$$

It follows from (15.6), (15.8), and (15.10) that the m-level concatenated code C is the direct-sum of m-component concatenated codes; that is,

$$C = B_1 \circ [A_1/A_2] \oplus B_2 \circ [A_2/A_3] \oplus \ldots \oplus B_m \circ [A_m/A_{m+1}]. \tag{15.11}$$

The minimum distance of C is lower bounded as follows:

$$d_{min}(C) \geq \min\{d_{A_i} d_{B_i} : 1 \leq i \leq m\} \tag{15.12}$$

(see Problem 17.2). The dimension of C is

$$K = \sum_{i=1}^{m} K_i (k_i - k_{i+1}).$$

EXAMPLE 15.2

Let $n = 8$ and $N = 8$. We choose A_1 and its subcodes as the following RM codes of length $2^3 = 8$: $A_1 = RM(3, 3)$, $A_2 = RM(2, 3)$, $A_3 = RM(1, 3)$, $A_4 = RM(0, 3)$, and $A_5 = \{\mathbb{0}\}$. A_1 is simply the $(8, 8)$ universal code with minimum distance $d_{A_1} = 1$; A_2 is the $(8, 7)$ even parity-check code with minimum distance $d_{A_2} = 2$; A_3 is the $(8, 4)$ first-order RM code with minimum distance $d_{A_3} = 4$; and A_4 is the $(8, 1)$ repetition code with minimum distance $d_{A_4} = 8$. Because

$$RM(3, 3) \supset RM(2, 3) \supset RM(1, 3) \supset RM(0, 3) \supset \{\mathbb{0}\},$$

we can form the following coset codes:

$$[A_1/A_2] = [(8, 8)/(8, 7)],$$
$$[A_2/A_3] = [(8, 7)/(8, 4)],$$
$$[A_3/A_4] = [(8, 4)/(8, 1)],$$
$$[A_4/\{\mathbb{0}\}] = [(8, 1)/\{\mathbb{0}\}].$$

The dimensions of these four coset codes are 1, 3, 3, and 1, respectively. Therefore, $q_1 = 2$, $q_2 = 2^3$, $q_3 = 2^3$, and $q_4 = 2$. In the construction of a 4-level concatenated code with the preceding coset codes as the inner codes, the four outer codes B_1, B_2, B_3, and B_4 must be codes over $GF(2)$, $GF(2^3)$, $GF(2^3)$, and $GF(2)$, respectively. Suppose we choose the following four outer codes:

1. B_1 is the binary $(8, 1)$ repetition code with minimum distance $d_{B_1} = 8$;
2. B_2 is the extended $(8, 5)$ RS code over $GF(2^3)$ with $d_{B_2} = 4$;

3. B_3 is the extended (8, 7) RS code over $GF(2^3)$ with $d_{B_3} = 2$; and
4. B_4 is the (8, 8) binary universal code with $d_{B_4} = 1$.

The 4-level concatenated code constructed from the foregoing inner and outer codes is

$$C = B_1 \circ [A_1/A_2] \oplus B_2 \circ [A_2/A_3] \oplus B_3 \circ [A_3/A_4] \oplus B_4 \circ [A_4/\{\mathbb{0}\}],$$

which is a (64, 45) binary linear block code. From (15.12) the minimum distance $d_{min}(C)$ of C is lower bounded by

$$d_{min}(C) \geq \min\{1 \times 8,\ 2 \times 4,\ 4 \times 2,\ 8 \times 1\} = 8.$$

In fact, it can be shown that $d_{min} = 8$. This code has the same code parameters as the (64, 45) extended primitive BCH code obtained by adding an overall parity bit to the (63, 45) BCH code.

EXAMPLE 15.3

Suppose we choose the following codes as the coset inner codes in the construction of a 3-level concatenated code:

$$[A_1/A_2] = [RM(2, 3)/RM(1, 3)] = [(8, 7)/(8, 4)],$$
$$[A_2/A_3] = [RM(1, 3)/RM(0, 3)] = [(8, 4)/(8, 1)],$$
$$[A_3/\{\mathbb{0}\}] = [RM(0, 3)/\{\mathbb{0}\}] = [(8, 1)/\{\mathbb{0}\}].$$

Then, the outer codes B_1, B_2, and B_3 must have symbols from $GF(2^3)$, $GF(2^3)$, and $GF(2)$, respectively. Suppose we choose the following as the outer codes: (1) B_1 is the (8, 1) extended RS code over $GF(2^3)$ with a minimum distance of 8; (2) B_2 is the (8, 5) extended RS code over $GF(2^3)$ with a minimum distance of 4; and (3) B_3 is the (8, 7) binary even parity-check code with a minimum distance of 2. Then, the 3-level concatenated code

$$C = B_1 \circ [RM(2, 3)/RM(1, 3)] \oplus B_2 \circ [RM(1, 3)/RM(0, 3)] \oplus B_3 \circ [RM(0, 3)/\{\mathbb{0}\}]$$

is a (64, 25) binary linear block code with a minimum distance of 16. This code has one information bit more than the (64, 24) extended BCH obtained by adding an overall parity bit to the (63, 24) primitive BCH code.

The preceding two examples show that the multilevel construction of concatenated codes results in good codes. Good multilevel concatenated codes and their trellis complexities can be found in [9].

An m-level concatenated code can be decoded in m stages. The first-level concatenated code $B_1 \circ [A_1/A_2]$ is decoded first, and the mth level concatenated code $B_m \circ [A_m/\{\mathbb{0}\}]$ is decoded at the last stage. Suppose a codeword $\mathbb{c}$ corresponding to m outer codewords $\mathbb{b}^{(1)}, \mathbb{b}^{(2)}, \ldots, \mathbb{b}^{(m)}$ is transmitted. Let $\mathbb{r} = (\mathbb{r}_0, \mathbb{r}_1, \ldots, \mathbb{r}_{N-1})$ be

the received sequence, where each received component $\mathbb{r}_j$ is the sum of a transmitted codeword in A_1 and an error vector. At the first decoding stage, $\mathbb{r}^{(1)} \triangleq \mathbb{r}$ is decoded into a codeword $\mathbb{b}^{(1)} = (b_0^{(1)}, b_1^{(1)}, \ldots, b_{N-1}^{(1)})$ in the first-level outer code B_1. Then, a modified received sequence is formed:

$$\begin{aligned}\mathbb{r}^{(2)} &= (\mathbb{r}_0^{(2)}, \mathbb{r}_1^{(2)}, \ldots, \mathbb{r}_{N-1}^{(2)}) \\ &= \mathbb{r}^{(1)} - (f_1(b_0^{(1)}), f_1(b_1^{(1)}), \ldots, f_1(b_{N-1}^{(1)})).\end{aligned} \tag{15.13}$$

If there are no errors in $\mathbb{r}^{(1)} = \mathbb{r}$, each component $\mathbb{r}_j^{(2)}$ is a codeword in A_2; otherwise, $\mathbb{r}_j^{(2)}$ of $\mathbb{r}^{(2)}$ is the sum of a codeword in A_2 and an error vector. At the second decoding stage, $\mathbb{r}^{(2)}$ is decoded into a codeword $\mathbb{b}^{(2)} = (\mathbb{b}_0^{(2)}, \mathbb{b}_1^{(2)}, \ldots, \mathbb{b}_{N_1}^{(2)})$ in the second-level outer code B_2. Then, another modified received sequence is formed:

$$\begin{aligned}\mathbb{r}^{(3)} &= (\mathbb{r}_0^{(3)}, \mathbb{r}_1^{(3)}, \ldots, \mathbb{r}_{N-1}^{(3)}) \\ &= \mathbb{r}^{(2)} - (f_2(b_0^{(2)}), f_2(b_1^{(2)}), \ldots, f_2(b_{N-1}^{(2)})).\end{aligned} \tag{15.14}$$

Again, if there are no errors in $\mathbb{r}^{(1)} = \mathbb{r}$, each component $\mathbb{r}_j^{(3)}$ is a codeword in A_3; otherwise $\mathbb{r}_j^{(3)}$ is the vector sum of a codeword in A_3 and an error vector. At the third-stage decoding, $\mathbb{r}^{(3)}$ is decoded into a codeword in the third-level outer code B_3. This decoding process continues until $\mathbb{r}^{(m)}$ is formed and decoded into a codeword $\mathbb{b}^{(m)}$ in the mth level outer code B_m.

Each decoding stage consists of two steps, the inner and outer decodings. At the ith decoding stage, the modified received sequence

$$\mathbb{r}^{(i)} = (\mathbb{r}_0^{(i)}, \ldots, \mathbb{r}_{N-1}^{(i)})$$

is first formed based on the decoding results of previous decoding stages. The inner code decoder decodes each component $\mathbb{r}_j^{(i)}$ into a codeword in A_i and identifies the coset in A_i/A_{i+1} that contains the decoded codeword. Let $f_i(\tilde{b}_j^{(i)})$ denote the representative of this coset. By inverse mapping, we obtain $\tilde{b}_j^{(i)}$ from $f_i(\tilde{b}_j^{(i)})$. After N inner code decodings, we obtain a vector over $GF(q_i)$:

$$\tilde{\mathbb{b}}^{(i)} = (\tilde{b}_0^{(i)}, \tilde{b}_1^{(i)}, \ldots, \tilde{b}_{N-1}^{(i)}).$$

This vector is then decoded based on the ith level outer code B_i into a codeword $\mathbb{b}^{(i)}$ in B_i. This completes the ith decoding stage. The decoder proceeds to the next decoding stage.

In the foregoing multistage decoding, the decoded estimate at one stage is passed to the next stage for decoding the next component concatenated code. If a decoding error is committed at a stage before the last stage, *error propagation* is likely to occur.

15.3 A SOFT-DECISION MULTISTAGE DECODING

The multistage decoding presented in the previous section can be modified for soft-decision decoding of multilevel concatenated codes and decomposable codes

[10–19]. To reduce computational complexity, each level is decoded with a trellis-based decoding algorithm.

Consider the multilevel concatenated system shown in Figure 15.6. For convenience we reintroduce some notations that were defined in the previous section:

1. $\mathbb{r} = (\mathbb{r}_0, \mathbb{r}_1, \ldots, \mathbb{r}_{N-1})$ is the soft-decision received sequence at the output of the matched filter, where for $0 \le j < N$, $\mathbb{r}_j = (r_{j,0}, r_{j,1}, \ldots, r_{j,n-1})$ is the jth received n-tuple.
2. $\mathbb{r}^{(i)} = (\mathbb{r}_0^{(i)}, \mathbb{r}_1^{(i)}, \ldots, \mathbb{r}_{N-1}^{(i)})$ is the modified received sequence for the ith stage decoding, where for $0 \le j < N$, $\mathbb{r}_j^{(i)} = (r_{j,0}^{(i)}, r_{j,1}^{(i)}, \ldots, r_{j,n-1}^{(i)})$.
3. $\mathbb{b}^{(i)} = (b_0^{(i)}, b_1^{(i)}, \ldots, b_{N-1}^{(i)})$ is the decoded codeword in outer code B_i at the ith decoding stage.
4. $\mathbb{c}^{(i)} = (\mathbb{c}_0^{(i)}, \mathbb{c}_1^{(i)}, \ldots, \mathbb{c}_{N-1}^{(i)})$ is the decoded codeword in the ith level concatenated code $B_i \circ [A_i/A_{i+1}]$, where for $0 \le j < N$,

$$\begin{aligned}\mathbb{c}_j^{(i)} &= (c_{j,0}^{(i)}, c_{j,1}^{(i)}, \ldots, c_{j,n-1}^{(i)}) \\ &= f_i(b_j^{(i)}).\end{aligned}$$

Suppose the $(i-1)$th stage of decoding has been completed. Then, the received sequence $\mathbb{r}^{(i)} = (\mathbb{r}_0^{(i)}, \mathbb{r}_1^{(i)}, \ldots, \mathbb{r}_{N-1}^{(i)})$ for the ith stage of decoding is obtained from $\mathbb{c}^{(i-1)}$ and $\mathbb{r}^{(i-1)}$ as follows: for $0 \le l < n$ and $0 \le j < N$, the lth symbol of the jth section $\mathbb{r}_j^{(i)}$ of $\mathbb{r}^{(i)}$ is given by

$$r_{j,l}^{(i)} = r_{j,l}^{(i-1)} \cdot (1 - 2c_{j,l}^{(i-1)}) \tag{15.15}$$

(assuming BPSK signaling).

Each stage of decoding consists of two steps, the inner and outer decodings. At the ith decoding stage, the inner decoder processes the N sections of the received sequence $\mathbb{r}^{(i)}$ independently and forms N metric tables for the outer code decoding. Let $\{\mathbb{a}^{(i)}\} \triangleq \{\mathbb{a}^{(i)} + \mathbb{a} : \mathbb{a} \in A_{i+1}\}$ be a coset in A_i/A_{i+1} with $\mathbb{a}^{(i)}$ as the coset representative. The metric of $\{\mathbb{a}^{(i)}\}$ with respect to $\mathbb{r}_j^{(i)}$ is defined as the largest correlation metric between $\mathbb{r}_j^{(i)}$ and the vectors in $\{\mathbb{a}^{(i)}\}$, denoted by $M(\{\mathbb{a}^{(i)}\})$. Let $\mathbb{a}^{(i)} + \mathbb{a}^*$ be the vector in $\{\mathbb{a}^{(i)}\}$ with the largest metric. This vector is called the label of $\{\mathbb{a}^{(i)}\}$ with respect to $\mathbb{r}_j^{(i)}$. A metric table, denoted by $\mathrm{MT}_j^{(i)}$, is formed that for each coset $\{\mathbb{a}^{(i)}\} \in A_i/A_{i+1}$, stores its metric $M(\{\mathbb{a}^{(i)}\})$ and its label $\mathbb{a}^{(i)} + \mathbb{a}^*$. This table is called the metric table for $\mathbb{r}_j^{(i)}$. The inner decoding at the ith stage is to form N metric tables, one for each section of the received sequence $\mathbb{r}^{(i)}$. These N metric tables are then passed to the outer decoder. For a codeword $\mathbb{b}^{(i)}$ in the ith outer code B_i, the metric of $\mathbb{b}^{(i)}$ is defined as the following sum:

$$M(\mathbb{b}^{(i)}) \triangleq \sum_{j=0}^{N-1} M(\{f_i(b_j^{(i)})\}), \tag{15.16}$$

where $M(\{f_i(b_j^{(i)})\})$ is the metric of the coset $\{f_i(b_j^{(i)})\}$ with $f_i(b_j^{(i)})$ as the coset leader that is stored in table $\mathrm{MT}_j^{(i)}$. The outer code decoding is to find the codeword

$\mathbb{b}^{(i)} \in B_i$ that has the largest metric among all the codewords in B_i, by using a trellis-based decoding algorithm, such as the Viterbi algorithm or RMLD, to search through the trellis for B_i.

Owing to possible error propagation from one decoding stage to the next, the foregoing multistage decoding is not optimum (i.e., it does not achieve MLD performance), even though each stage of decoding is MLD. It is a suboptimum decoding algorithm; however, it provides an efficient trade-off between error performance and decoding complexity for multilevel concatenated codes and decomposable codes.

The performance of multistage decoding can be improved by passing a list of L best estimates from one stage to another and then choosing the estimate with the largest metric at the last decoding stage as the final decoded codeword, as with the list Viterbi algorithm [20] at each stage. The improvement and the additional decoding complexity is determined by the size of the list.

15.4 DECOMPOSITION OF CODES

The opposite of multilevel construction of concatenated codes presented in the previous section is decomposition of codes into component codes. A code is said to be *μ-level decomposable* if it can be expressed as a μ-level concatenated code. Such a decomposable code can be decoded in multiple stages, as described in the previous section. Multistage decoding of long decomposable codes reduces decoding complexity significantly, because after decomposition, component codes are shorter in length and smaller in dimension, and require less decoding complexity.

The most well known class of decomposable codes is the class of RM codes. In Section 4.4, it was shown that a RM code can be decomposed in many ways using squaring construction; however, a RM code also can be decomposed as a multilevel concatenated code [10, 11, 14–17]. Consider the rth-order RM code of length 2^m, $RM(r, m)$. Let $k(r, m)$ and $q(r, m)$ denote the dimensions of the $RM(r, m)$ code and the coset code, $[RM(r, m)/RM(r-1, m)]$, respectively. (Note that $RM(-1, m) = \{\mathbb{0}\}$). From Section 4.2 we find that

$$k(r, m) = \sum_{i=0}^{r} \binom{m}{i}$$

and

$$q(r, m) = \binom{m}{r}. \tag{15.17}$$

For $0 \leq l \leq r$, we can readily show that

$$k(r, m) = k(r - l, m) + \sum_{i=0}^{l-1} q(r - i, m). \tag{15.18}$$

Let $(n, k)^{\lambda}$ denote the linear block code over $GF(2^{\lambda})$ obtained by interleaving the binary (n, k) code to a depth of λ with each group of λ interleaved bits regarded as a symbol in $GF(2^{\lambda})$. Then, from (4.71), it is possible to decompose the $RM(r, m)$ code

as a $(\mu + 1)$-level concatenated code as follows [14]:

$$RM(r, m) = \{RM(0, \nu)^{q(r,m-\nu)}, RM(1, \nu)^{q(r-1,m-\nu)}, \cdots, RM(\mu, \nu)^{k(r-\mu,m-\nu)}\}$$
$$\circ\{RM(r, m-\nu), RM(r-1, m-\nu), \cdots, RM(r-\mu, m-\nu)\}, \quad (15.19)$$

where $1 \leq \nu \leq m-1$, $\mu = \nu$ for $r > \nu$, and $\mu = r$ otherwise.

Expression (15.19) shows that a RM code can be decomposed as a multilevel concatenated code in many ways with various levels. We also note that in any decomposition of a RM code, all the component inner and outer codes are also RM codes (or interleaved RM codes) of shorter lengths and smaller dimensions.

EXAMPLE 15.4

Let $m = 6$ and $r = 3$. Consider the $RM(3, 6)$ code that is a $(64, 42)$ code with a minimum distance of 8. Set $\mu = 3$ and $\nu = 3$. Then, it follows from (15.19) that the $RM(3, 6)$ code can be decomposed into a 4-level concatenated code as follows:

$$RM(3, 6) = \{RM(0, 3)^{q(3,3)}, RM(1, 3)^{q(2,3)}, RM(2, 3)^{q(1,3)}, RM(3, 3)^{q(0,3)}\}$$
$$\circ\{RM(3, 3), RM(2, 3), RM(1, 3), RM(0, 3)\}.$$

We find that $q(3, 3) = 1$, $q(2, 3) = 3$, $q(1, 3) = 3$, $q(0, 3) = 1$; $RM(0, 3)$ is the $(8, 1)$ repetition code; $RM(1, 3)$ is the $(8, 4)$ code with a minimum distance of 4; $RM(2, 3)$ is the $(8, 7)$ even parity-check code; and $RM(3, 3)$ is the $(8, 8)$ universal code. Hence,

$$\begin{aligned} RM(3, 6) &= \{(8, 1), (8, 4)^3, (8, 7)^3, (8, 8)\} \circ \{(8, 8), (8, 7), (8, 4), (8, 1)\} \\ &= (8, 1) \circ [(8, 8)/(8, 7)] \oplus (8, 4)^3 \circ [(8, 7)/(8, 4)] \\ &\quad \oplus (8, 7)^3 \circ [(8, 4)/(8, 1)] \oplus (8, 8) \circ [(8, 1)/\{\emptyset\}]. \end{aligned} \quad (15.20)$$

A trellis diagram can be constructed for each level of concatenated code. The $(8, 1)$ code has a trivial 2-state 8-section trellis. Therefore, the first-level concatenated code $(8, 1) \circ [(8, 8)/(8, 7)]$ has a 2-state 8-section trellis, and each section is 8 bits long. The $(8, 4)$ code has a 4-state 4-section trellis, as shown in Figure 9.17. By Cartesian product, the $(8, 4)^3$ interleaved code has a 4-section trellis with 64-states that consists of 8 parallel and structurally identical subtrellises, and each has 8 states. Consequently, the second-level concatenated code $(8, 4)^3 \circ [(8, 7)/(8, 5)]$ has a 4-section 64-state trellis, each section is 16 bits long, and the 16 bits of a branch label represent two codewords in the inner code $[(8, 7)/(8, 4)]$. If the 8-section bit-level trellis as shown in Figure 9.6 is used for the $(8, 4)$ code, then the $(8, 4)^3$ code has an 8-section trellis with 512 states at time-3 and time-5. Consequently, the second-level concatenated code $(8, 4)^3 \circ [(8, 7)/(8, 4)]$ has an 8-section trellis, each section is 8 bits long, and each branch label is a codeword in $[(8, 7)/(8, 4)]$. Now, consider the third-level concatenated code. The $(8, 7)$ even parity-check code has an 8-section 2-state trellis, as shown in Figure 15.7. Then, the $(8, 7)^3$ interleaved code has an 8-section 8-state trellis. The third-level concatenated code $(8, 7)^3 \circ [(8, 4)/(8, 1)]$ has an 8-section 8-state trellis, each section is 8 bits long, and each 8-bit branch label is a codeword in the inner code $[(8, 4)/(8, 1)]$. Finally, the fourth-level concatenated code $(8, 8) \circ [(8, 1)/\{\emptyset\}]$ has a trivial 1-state 8-section trellis, each section is 8 bits long, and each 8-bit branch label is a codeword in the $(8, 1)$ code.

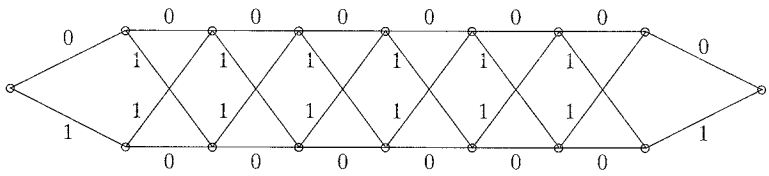

FIGURE 15.7: The 8-section bit-level trellis for the (8, 7) even parity-check code.

The preceding example shows that when we decompose a code into an m-level concatenated code, we also decompose its trellis into m trellises, one for each level of concatenated code. The trellis for each level of concatenated code has much smaller state and branch complexities than those of the original code. In multistage decoding of a decomposable code, each level can be decoded by using a trellis-based decoding algorithm, which reduces decoding computational complexity significantly.

As an example, consider the fourth-order RM code of length 128, $RM(4, 7)$. This code is a (128, 99) code with a minimum distance of 8. This code has a uniform 16-section trellis with maximum state complexity of 2^{19} states. Implementation of any decoding algorithm based on this full-code trellis is practically impossible; however, this code can be decomposed into a 3-level concatenated code as follows [17]:

$$RM(4, 7) = \{(16, 5)(16, 11), (16, 11)^2, (16, 15)^3(16, 16)\} \circ \{(8, 8), (8, 6), (8, 4)\}$$

where (16, 5) and (16, 11) are first- and second-order RM codes of length 16. Both (16, 5) and (16, 11) RM codes have 16-section bit-level trellises with maximum state complexity of 16 states (see Figures 9.7 and 9.8). The first-level outer code $B_1 = (16, 5)(16, 11)$ is an interleaved code and is regarded as a code of length 16 over $GF(2^2)$. This code has a 16-section trellis with maximum state complexity of 256 states. The second-level outer code $B_2 = (16, 11)^2$ has a 16-section trellis with maximum state complexity of 256 states. The third-level outer code $B_3 = (16, 15)^3(16, 16)$ has a 16-section trellis with 8 states. The binary (8, 6) code is a subcode of the (8, 7) RM code. The preceding decomposition results in relatively small trellis state complexity compared with the 2^{19} states of the original undecomposed code. Therefore, three-stage decoding of this code requires relatively little decoding complexity. The bit-error performance of this code with the 3-stage soft-decision decoding presented in the previous section is shown in Figure 15.8. Performance is degraded by 0.9 dB compared with optimum MLD at BER of 10^{-5}, but there is a 1.7-dB coding gain over the majority-logic decoding of the code.

As another example, consider the third-order RM code of length 128, $RM(3, 7)$, which is a (128, 64) code with a minimum distance of 16. The 16-section trellis for this code has a maximum state complexity of 2^{26} states! This code can be decomposed into a 3-level concatenated code as follows [17]:

$$RM(3, 7) = \{(16, 1)(16, 5)^2, (16, 5)(16, 11), (16, 11)^2(16, 15)\} \circ \{(8, 8), (8, 5), (8, 3)\}.$$

With this decomposition, the third-level outer code $B_3 = (16, 11)^2(16, 5)$ has the largest trellis complexity, 512 states, which is very small compared with 2^{26}. The bit-error performance of this code with 3-stage decoding is shown in Figure 15.9.

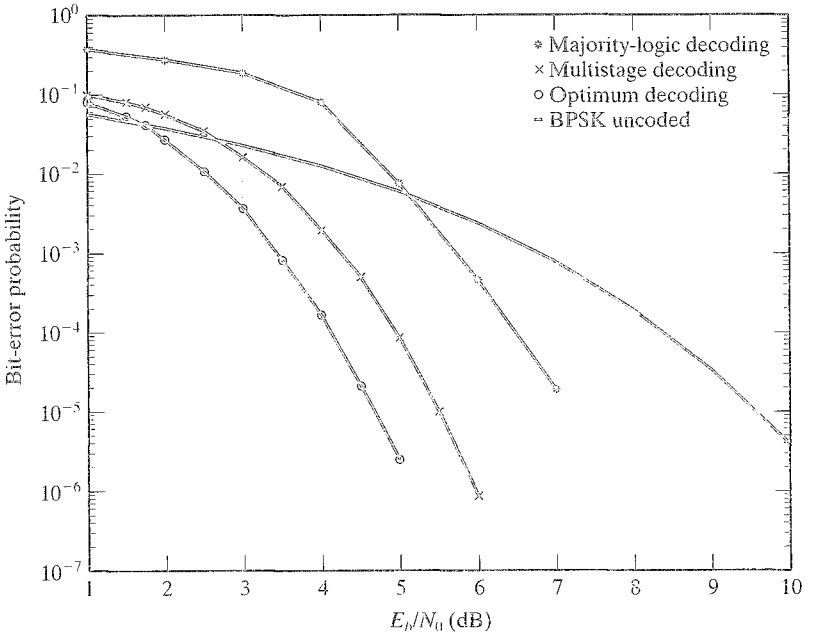

FIGURE 15.8: The error performance of the (128, 99) RM code with various decodings.

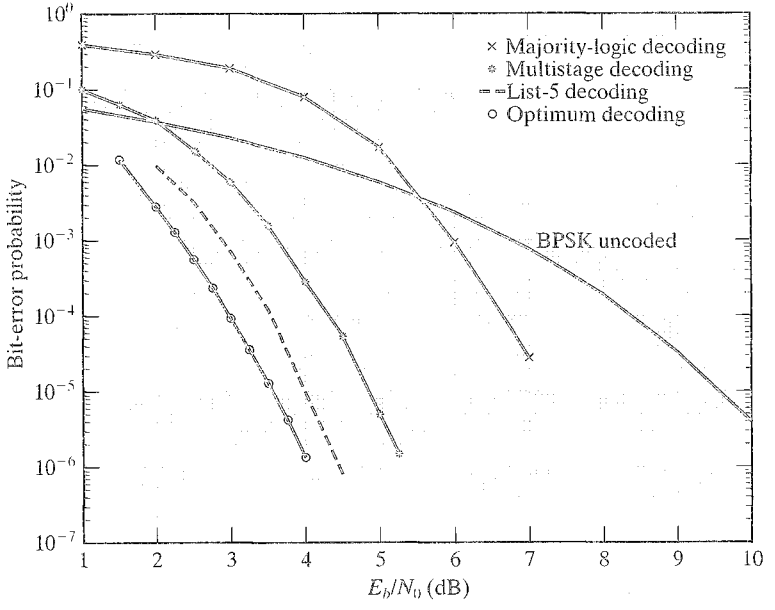

FIGURE 15.9: The error performance of the (128, 64) RM code with various decodings.

There is a 1.3-dB loss in coding gain compared with the optimum MLD; however the reduction in decoding complexity is enormous. If a list of five best estimates is passed from one decoding stage to the next stage, a 0.8-dB coding gain loss versus the optimum MLD can be recovered, as shown in Figure 15.9.

15.5 AN ITERATIVE MULTISTAGE MLD ALGORITHM

In fact, MLD can be achieved with the multistage decoding presented in Section 15.3 in conjunction with decoding iterations between stages and an optimality test at each stage. Iteration is initiated only when the optimality test fails. In the following an iterative multistage MLD (IMS-MLD) algorithm [17–19]. For simplicity, we first describe this algorithm using a two-level concatenated code, then we generalize it to m-level codes.

Let $\mathbb{b}^{(1)} = (b_1^{(1)}, b_2^{(1)}, \cdots, b_N^{(1)})$ be the decoded codeword at the first stage. Then, $\mathbb{b}^{(1)}$ has the best metric with respect to the received sequence $\mathbb{r}^{(1)} = \mathbb{r}$. The metric of $\mathbb{b}^{(1)}$ is given by (15.16):

$$M(\mathbb{b}^{(1)}) \triangleq \sum_{j=1}^{N} M(\{f_i(b_j^{(1)})\}),$$

where $M(\{f_1(b_j^{(1)})\})$ is the metric of the coset $\{f_1(b_j^{(1)})\} \in A_1/A_2$. Let $f_1(b_j^{(1)}) + \mathbb{a}_j^*$ be the label of coset $\{f_1(b_j^{(1)})\}$ with respect to the jth section $\mathbb{r}_j^{(1)}$ of $\mathbb{r}^{(1)}$. Then,

$$M(\{f_1(b_j^{(1)})\}) = M(f_1(b_j^{(1)}) + \mathbb{a}_j^*), \text{ and } M(\mathbb{b}^{(1)}) = \sum_{j=1}^{N} M(f_1(b_j^{(1)}) + \mathbb{a}_j^*). \quad (15.21)$$

If we replace each component $b_j^{(1)}$ of the decoded codeword $\mathbb{b}^{(1)}$ with its corresponding coset label $f_1(b_j^{(1)}) + \mathbb{a}^*$, we obtain the following *coset label sequence*:

$$L(\mathbb{b}^{(1)}) \triangleq (f_1(b_1^{(1)}) + \mathbb{a}_1^*, f_1(b_2^{(1)}) + \mathbb{a}_2^*, \cdots, f_1(b_N^{(1)}) + \mathbb{a}_N^*). \quad (15.22)$$

From (15.21) and (15.22) we see that the metric of $\mathbb{b}^{(1)}$ is the metric of its corresponding coset label sequence.

If the coset label sequence $L(\mathbb{b}^{(1)})$ is a codeword in the overall concatenated code $C = \{B_1, B_2\} \circ \{A_1, A_2\}$, then it is the codeword in C that has the best metric and hence is the most likely codeword. In this case the decoding is completed and the sequence $\mathbb{a}^* = (\mathbb{a}_1^*, \mathbb{a}_2^*, \cdots, \mathbb{a}_N^*)$ is a codeword in the second-level concatenated code $C_2 = B_2 \circ A_2$. The second outer codeword $\mathbb{b}^{(2)} = (b_1^{(2)}, b_2^{(2)}, \cdots, b_N^{(2)})$ can be recovered from $\mathbb{a}^*$ as follows: $\mathbb{b}^{(2)} = (f_2^{-1}(\mathbb{a}_1^*), f_2^{-1}(\mathbb{a}_2^*), \cdots, f_2^{-1}(\mathbb{a}_N^*))$, where $f_2^{-1}(\cdot)$ is the inverse mapping of $f_2(\cdot)$. Alternatively, for systematic encoding, the information bits are obtained directly from the coset label sequence. If the coset label sequence $L(\mathbb{b}^{(1)})$ is not a codeword in C, then it is the sum of a codeword $(f_1(b_1^{(1)}), f_1(b_2^{(1)}), \cdots, f_1(b_N^{(1)}))$ in $C_1 = B_1 \circ [A_1/A_2]$ and a sequence of codewords in A_2 that is not a codeword in $C_2 = B_2 \circ A_2$. In this case, decoding continues. Therefore, the condition that $L(\mathbb{b}^{(1)})$ is a codeword in C is an optimality condition, which will be used in the IMS-MLD algorithm.

At the second stage of decoding, the received sequence $\mathbb{r}^{(2)} = (\mathbb{r}_1^{(2)}, \mathbb{r}_2^{(2)}, \cdots, \mathbb{r}_N^{(2)})$ for decoding is obtained by removing the effect of the decoded codeword $\mathbb{b}^{(1)}$ at the first stage from $\mathbb{r}^{(1)} = \mathbb{r}$, as given by (15.15). The inner decoder forms N metric tables, one for each section of $\mathbb{r}^{(2)}$. These tables are then passed to the outer decoder. The outer decoder finds the codeword $\mathbb{b}^{(2)} = (b_1^{(2)}, b_2^{(2)}, \cdots, b_N^{(2)})$ in B_2 that has the best metric with respect to $\mathbb{r}^{(2)}$. The metric of $\mathbb{b}^{(2)}$ is given by

$$M(\mathbb{b}^{(2)}) = \sum_{j=1}^{N} M(f_2(b_j^{(2)})). \tag{15.23}$$

For $1 \leq j \leq N$, let $f_2(b_j^{(2)}) = \mathbb{a}_j^{(2)}$, which is a codeword in A_2. Then, $\mathbb{a}^{(2)} \triangleq (\mathbb{a}_1^{(2)}, \mathbb{a}_2^{(2)}, \cdots, \mathbb{a}_N^{(2)})$ is a codeword in $C_2 = B_2 \circ A_2$, and

$$M(\mathbb{b}^{(2)}) = M(\mathbb{a}^{(2)}) = \sum_{j=1}^{N} M(\mathbb{a}_j^{(2)}), \tag{15.24}$$

where $M(\mathbb{a}_j^{(2)})$ is the metric of $\mathbb{a}_j^{(2)}$ with respect to $\mathbb{r}_j^{(2)}$.

Now, we compare metric $M(\mathbb{b}^{(1)})$ and metric $M(\mathbb{b}^{(2)})$. Recall that

$$M(f_1(\mathbb{b}_j^{(1)}) + \mathbb{a}_j^*) = \max_{\mathbb{a}_j \in A_2} M(f_1(\mathbb{b}_j^{(1)}) + \mathbb{a}_j).$$

Then, for any $\mathbb{a}_j \in A_2$,

$$M(f_1(\mathbb{b}_j^{(1)}) + \mathbb{a}_j^*) \geq M(f_1(\mathbb{b}_j^{(1)}) + \mathbb{a}_j). \tag{15.25}$$

Because $\mathbb{a}_j^{(2)} \in A_2$, it follows from (15.25) that

$$M(f_1(\mathbb{b}_j^{(1)}) + \mathbb{a}_j^*) \geq M(f_1(\mathbb{b}_j^{(1)}) + \mathbb{a}_j^{(2)}). \tag{15.26}$$

Let $\text{corr}(\cdot, \cdot)$ denote the correlation function (any other metric can be used as well). It is easy to show that

$$corr(f_1(\mathbb{b}_j^{(1)}) + \mathbb{a}_j, \mathbb{r}_j^{(1)}) = corr(\mathbb{a}_j, \mathbb{r}_j^{(2)}) \tag{15.27}$$

for any $\mathbb{a}_j \in A_2$. Then, it follows from (15.21), (15.24), (15.26), and (15.27) that

$$M(\mathbb{b}^{(1)}) \geq M(\mathbb{b}^{(2)}), \tag{15.28}$$

where equality holds if and only if the sequence $\mathbb{a}^* = (\mathbb{a}_1^*, \mathbb{a}_2^*, \cdots, \mathbb{a}_N^*)$ is a codeword in C_2.

For the iterative two-stage MLD algorithm, decoding iterations are based on the generation of a sequence of estimates at the first stage. The estimates are generated in decreasing order of metrics, one at a time, as by using the list Viterbi algorithm [20]. At the ith iteration, the first-stage decoder generates the ith best estimate, denoted by $\mathbb{b}^{(1),i} = (b_1^{(1),i}, b_2^{(1),i}, \cdots, b_N^{(1),i})$. Let $\mathbb{b}^{(2),i}$ denote the decoded codeword at the second stage based on $\mathbb{b}^{(1),i}$ and $\mathbb{r}^{(2)}$. The algorithm is based on the following two theorems. Theorem 15.1 is a direct consequence of (15.28).

THEOREM 15.1 For $i > 0$, $M(\mathbb{b}^{(1),i}) \geq M(\mathbb{b}^{(2),i})$, where equality holds if and only if the coset label sequence for $\mathbb{b}^{(1),i}$, $L(\mathbb{b}^{(1),i}) \triangleq (f_1(b_1^{(1),i}) + \mathbb{a}_1^*, f_1(b_2^{(1),i}) + \mathbb{a}_2^*, \cdots, f_1(b_N^{(1),i}) + \mathbb{a}_N^*)$, is a codeword in C; that is, $(\mathbb{a}_1^*, \mathbb{a}_2^*, \cdots, \mathbb{a}_N^*)$ is a codeword in $C_2 = B_2 \circ A_2$.

Let i_0 be the integer such that $1 \leq i_0 < i$ and

$$M(\mathbb{b}^{(2),i_0}) = \max_{1 \leq j < i} M(\mathbb{b}^{(2),j}). \tag{15.29}$$

Then, $\mathbb{b}^{(2),i_0}$ is the best decoded codeword at the second decoding stage during the first $i-1$ iterations. Theorem 15.2 follows from Theorem 15.1 and the fact that for $i < j$, $M(\mathbb{b}^{(1),i}) \geq M(\mathbb{b}^{(1),j})$.

THEOREM 15.2 For $i > i_0$, if $M(\mathbb{b}^{(2),i_0}) \geq M(\mathbb{b}^{(1),i})$, then the codeword in C that corresponds to $\mathbb{b}^{(1),i_0}$ and $\mathbb{b}^{(2),i_0}$ is the most likely codeword with respect to the received sequence $\mathbb{r}$. If $M(\mathbb{b}^{(2),i_0}) < M(\mathbb{b}^{(1),i})$ and the coset label sequence $L(\mathbb{b}^{(1),i})$ is a codeword in C, then $L(\mathbb{b}^{(1),i})$ is the most likely codeword in C with respect to $\mathbb{r}$.

In fact, the optimality conditions of Theorem 15.2 are also necessary conditions for the iterative decoding algorithm.

Decoding Algorithm

Step 1. Compute the first (best) estimate $\mathbb{b}^{(1),1}$ of the first decoding stage and its metric $M(\mathbb{b}^{(1),1})$. Check whether the coset label sequence $L(\mathbb{b}^{(1),1})$ is a codeword in C. If it is, $L(\mathbb{b}^{(1),1})$ is the most likely codeword and the decoding stops; otherwise, go to step 2.

Step 2. Perform second-stage decoding and obtain the estimate $L(\mathbb{b}^{(2),1})$ with metric $M(\mathbb{b}^{(2),1})$. Set $i_0 = 1$, and store $\mathbb{b}^{(1),1}$ and $\mathbb{b}^{(2),1}$. Go to step 3.

Step 3. For $i > i_0$, $\mathbb{b}^{(1),i_0}$ and $\mathbb{b}^{(2),i_0}$ are currently stored in a buffer register together with the metric $M(\mathbb{b}^{(2),i_0})$. Determine the ith best estimate $\mathbb{b}^{(1),i}$ of the outer code B_1, and its metric $M(\mathbb{b}^{(1),i})$. If $M(\mathbb{b}^{(2),i_0}) \geq M(\mathbb{b}^{(1),i})$, then $\mathbb{b}^{(1),i_0}$ and $\mathbb{b}^{(2),i_0}$ together give the most likely code in C, and decoding is finished; otherwise, go to step 4.

Step 4. Check if the coset label sequence $L(\mathbb{b}^{(1),i})$ is a codeword in C. If it is, $L(\mathbb{b}^{(1),i})$ is the most likely codeword in C and decoding is finished; otherwise, go to step 5.

Step 5. Generate $\mathbb{b}^{(2),i}$. Update i_0, $\mathbb{b}^{(1),i_0}$, $\mathbb{b}^{(2),i_0}$, and $M(\mathbb{b}^{(2),i_0})$. Go to step 3.

The decoding process iterates until the most likely codeword is found. Therefore, the maximum number of iterations is $q_1^{K_1}$. This is the extreme case. In general, the number of iterations required to obtain the most likely codeword is very small compared with $q_1^{K_1}$. We may limit the number of iterations to keep decoding computational complexity down and decoding delay small. In this case, the decoding algorithm achieves near-optimal error performance.

Theorems 15.1and 15.2, and the two-stage iterative MLD decoding can be generalized to m stages. In m-stage decoding, new decoding iteration can be initiated

at any stage above the final stage. Decoding iteration begins with the generation of a new estimate at the starting stage, say stage l. If all the $q_l^{K_l}$ estimates at stage l (resulting from a particular sequence of codewords from stages above stage l) have already been generated and tested, the decoder moves up to the $(l-1)$th stage and starts a new iteration with a new estimate. Decoding iterations continue until the ML codeword is found. Just as in two-stage decoding, the final decoding decision is made at the first stage.

Suppose the decoding process is at the ith decoding stage of the jth iteration. Let $\mathbb{b}^{(i),j}$ denote the decoded codeword in the outer code B_i. Let $L(\mathbb{b}^{(i),j})$ denote the coset label sequence corresponding to $\mathbb{b}^{(i),j}$. The metric of $L(\mathbb{b}^{(i),j})$ or $\mathbb{b}^{(i),j}$ is denoted by $M(\mathbb{b}^{(i),j})$. Let $\mathbb{b}^{(i_0),j_0}$ denote the codeword whose metric $M(\mathbb{b}^{(i_0),j_0})$ is the *best* (*largest*) among the codewords that have been generated before the ith decoding stage of the jth iteration and whose coset label sequence $L(\mathbb{b}^{(i_0),j_0})$ is a codeword in the overall m-level code C. Then, the buffer contains $M(\mathbb{b}^{(i_0),j_0})$, $L(\mathbb{b}^{(i_0),j_0})$, the estimates of the stages $1, 2, \cdots, (i_0-1)$ from which the estimate $\mathbb{b}^{(i_0),j_0}$ resulted, and the estimates of the stages $1, 2, \cdots, (i-1)$ from which the estimate $\mathbb{b}^{(i),j}$ resulted.

At the completion of the ith decoding stage of the jth iteration, the decoder makes one of the following three moves:

1. If $M(\mathbb{b}^{(i),j}) \leq M(\mathbb{b}^{(i_0),j_0})$, the decoder moves up to stage $(i-1)$ and starts a new iteration with a new estimate.
2. Otherwise, if $M(\mathbb{b}^{(i),j}) > M(\mathbb{b}^{(i_0),j_0})$, the coset label sequence $L(\mathbb{b}^{(i),j})$ is tested. If it is a codeword in C, then the buffer is updated ($M(\mathbb{b}^{(i_0),j_0}) = M(\mathbb{b}^{(i),j})$, etc.). The decoder moves up to stage $(i-1)$ and starts a new iteration with a new estimate.
3. If $M(\mathbb{b}^{(i),j}) > M(\mathbb{b}^{(i_0),j_0})$, and the coset label sequence $L(\mathbb{b}^{(i),j})$ is not a codeword in C, then the decoder moves down to the $(i+1)$th stage of the jth iteration.

When the decoder reaches the last (mth) stage, it must move up to the $(m-1)$th stage and start a new iteration (since the label sequence of the estimate $\mathbb{b}^{(m),j}$ is always a codeword in C).

Whenever the decoder reaches the first stage at the beginning of an iteration, a decision is made at the completion of the first-stage decoding whether the decoding is to be terminated or to continue. Suppose the decoder has reached and completed the first stage decoding at the jth iteration. The decoder makes one of the following moves:

1. If $M(\mathbb{b}^{(1),j}) \leq M(\mathbb{b}^{(i_0),j_0})$, then the decoding is finished. The ML codeword is formed from $\mathbb{b}^{(i_0),j_0}$ and the codewords above i_0th stage that resulted in the generation of $\mathbb{b}^{(i_0),j_0}$.
2. Otherwise, if $M(\mathbb{b}^{(1),j}) > M(\mathbb{b}^{(i_0),j_0})$, and the coset label sequence $L(\mathbb{b}^{(1),j})$ is a codeword in C, then $L(\mathbb{b}^{(1),j})$ is the ML codeword. Decoding stops.
3. If $M(\mathbb{b}^{(1),j}) > M(\mathbb{b}^{(i_0),j_0})$, and $L(\mathbb{b}^{(1),j})$ is not a codeword in C, then the decoder moves down to the second stage and continues the decoding for the jth iteration.

Thus, the tests performed at the first decoding stage are actually the optimality conditions.

In m-stage decoding, a new decoding iteration can be initiated at any stage above the final stage. Decoding iteration begins with the generation of a new estimate at the starting stage, say stage l. From there, the decoder can either move down to the next stage, or if the coset label sequence is a codeword in C_l, move up and start a new iteration at stage $(l-1)$. If all the $q_l^{K_l}$ estimates at stage l (resulting from a particular sequence of codewords from stages above stage l) have already been generated and tested, the decoder moves up to the $(l-1)$th stage and starts a new iteration with a new estimate. Decoding iterations continue until the ML codeword is found. Just as in two-stage decoding, the final decoding decision is made at the first stage.

In decoding with the IMS-MLD algorithm, the number of iterations and the computational complexity depend on the SNR; they decrease as the SNR increases. At a certain point the computational complexity of the IMS-MLD algorithm may become even smaller than that of the conventional multistage decoding presented in Section 15.3. To reduce the worst-case computational complexity, we may set a limit on the number of iterations to be performed. Of course, this limitation results in a suboptimum version of the IMS-MLD algorithm. If the limit on the maximum number of iterations is chosen properly, performance degradation will be small.

Consider the third-order RM code $RM(3, 7)$, which is a (128, 64) code with a minimum distance of 16. Suppose this code is decoded with the IMS-MLD algorithm based on the 3-level decomposition given in Section 15.4. Its bit-error performance is shown in Figure 15.10. We see the performance curve of

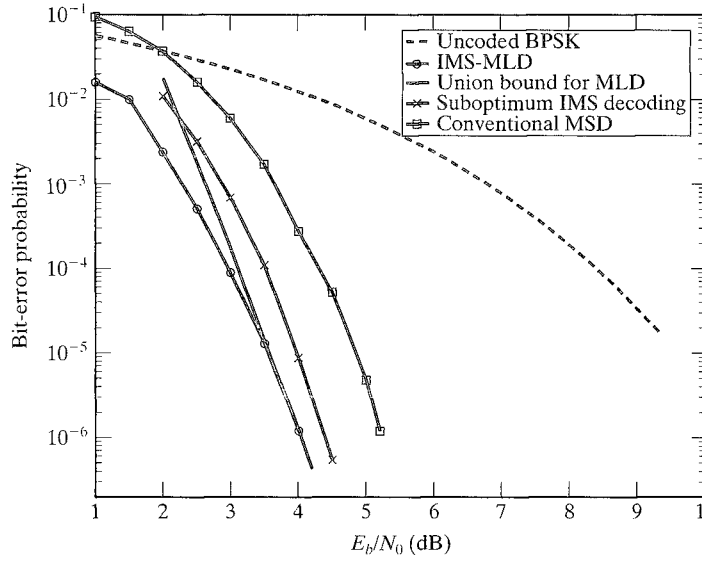

FIGURE 15.10: Bit-error performance of the (128, 64) RM code with various multi-stage decodings.

the IMS-MLD algorithm agrees with the union bound for MLD. Also included in the figure are the bit performances of the code with a suboptimum version of the IMS-MLD and the conventional multistage decoding presented in Section 15.3. We see that the IMS-MLD outperforms the conventional multistage decoding by 1.35 dB at the BER $= 10^{-6}$. For the suboptimum version, the maximum number of iterations is set to 7 (5 for the first stage and 2 for the second stage). From Figure 15.10 we see that performance degradation of the suboptimum algorithm is 0.3 dB compared with the IMS-MLD at the BER $= 10^{-6}$. Thus, IMS-MLD has a 1.05-dB coding gain over conventional multistage decoding.

The computational complexity for decoding the (128, 64) RM code with various decoding algorithms is given in Table 15.1. The computational complexity is expressed in terms of number of real operations (additions and comparisons). We see that the suboptimum IMS-MLD provides an effective trade-off between the performance of the IMS-MLD algorithm and the computational complexity of the conventional multistage decoding.

As another example, consider the fourth-order (128, 99, 8) RM code, $RM(4, 7)$, with a 3-level decomposition given in Section 15.4. The bit-error performances of this code with various decodings are shown in Figure 15.11. The IMS-MLD outperforms conventional multistage decoding by 0.75 dB; however the suboptimum IMS-MLD decoding with a limit of 7 iterations (5 for the first stage and 2 for the second stage) achieves almost optimum performance. The computational complexities of the three decodings are given in Table 15.2.

TABLE 15.1: Computational complexities of various multistage decodings of the (128, 64) RM code.

Decoding algorithm		Average number of real operations at SNR							Upper bound on complexity
		2.0	2.5	3.0	3.5	4.0	4.5	5.0	
Conventional MSD	$[10^4]$	2.57	1.88	1.42	1.03	0.80	0.65	0.55	4.26
Suboptimum IMS	$[10^4]$	6.25	3.33	1.82	1.15	0.83	—	—	36.6
Optimum IMS-MLD	$[10^4]$	45.0	12.6	10.0	4.0	2.60	—	—	—
Viterbi	$[10^9]$	9.3				9.29	9.28	9.23	9.3

TABLE 15.2: Computational complexities of various multistage decodings of the (128, 99) RM code.

Decoding algorithm		Average number of real operations at SNR							Upper bound on complexity
		2.5	3.0	3.5	4.0	4.5	5.0	5.5	
Conventional MSD	$[10^4]$	1.2	1.1	0.9	0.7	0.56	0.4	0.3	1.5
Suboptimum IMS	$[10^4]$	3.3	2.1	1.3	0.8	0.56	0.4	—	7.8
Optimum IMS-MLD	$[10^4]$	22.0	7.9	6.3	1.0	0.56	0.4	—	—
Viterbi	$[10^9]$	4.62	4.61	4.56	4.44	4.20	3.82	3.29	4.63

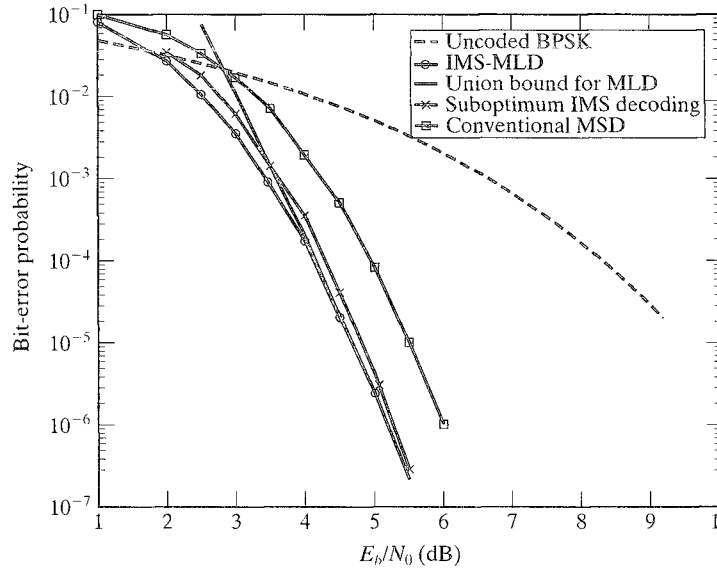

FIGURE 15.11: Bit-error performance of the (128, 99) RM code with various multi-stage decodings.

15.6 CONCATENATED CODING SCHEMES WITH CONVOLUTIONAL INNER CODES

So far only concatenated coding schemes with block inner codes have been considered; however, convolutional codes can be used as inner codes for constructing concatenated codes as well. Consider the interleaved concatenated coding system shown in Figure 15.1. Suppose we replace the block inner code with an (n_1, k_1, ν) convolutional code. We obtain a concatenated convolutional coding system. At the first stage of encoding, λ consecutive outer codewords are formed and stored in the interleaver buffer as a $\lambda \times n_2$ code array. At the second stage of encoding, the code array is read out column by column in binary form (λm bits per column), and the binary sequence, regarded as an information sequence, is then encoded by the inner convolutional code encoder continuously. For inner code encoding, we set $\lambda m = lk_1$. In this case each column of the code array in the interleaver buffer is encoded into l n_1-bit code blocks. The encoding results in a terminated convolutional code sequence. At the receiving end, the received sequence is first decoded by a 2^{ν}-state truncated Viterbi decoder. The decoded information bits are then grouped into m-bit bytes as symbols in $GF(2^m)$ and stored by column as a $\lambda \times n_2$ array in a deinterleaving buffer. The array in the deinterleaving buffer is read out row by row, and each row is decoded by the outer decoder.

In applications, a rate-$1/n_1$ convolutional code (or its punctured version) is often used as the inner code in a concatenated convolutional coding system. One such concatenated convolutional coding system is the error-control coding scheme used in the NASA Tracking Data Relay Satellite System (TDRSS). In this system, the (255, 223, 33) RS code over $GF(2^8)$ is used as the outer code, and the 64-state rate-1/2 convolutional code generated by the polynomials $\mathbf{g}_1(D) = 1 + D + D^3 + D^4 + D^6$

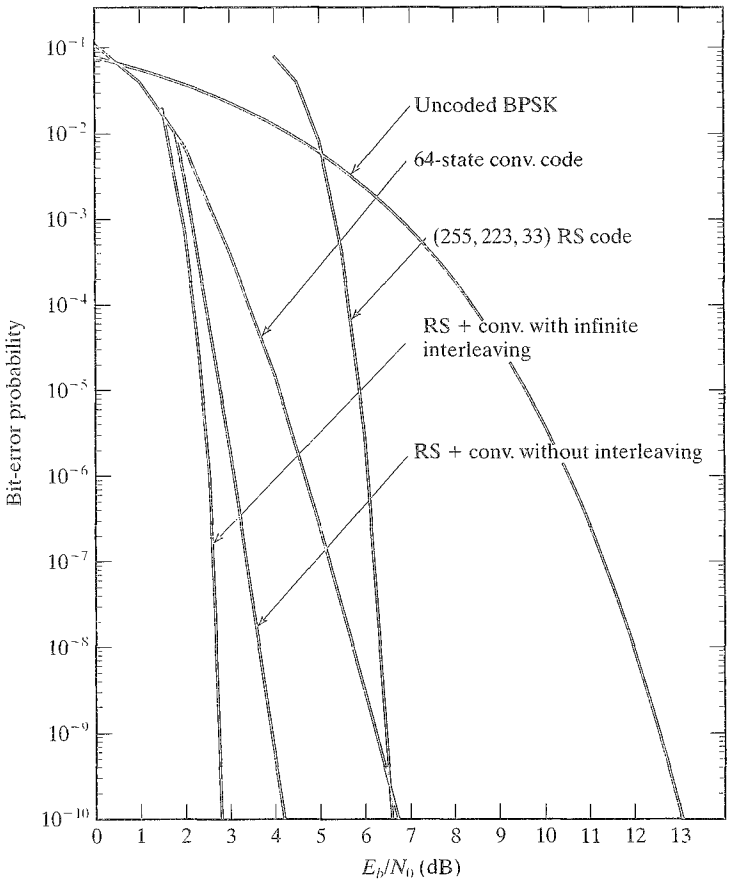

FIGURE 15.12: Error performance of the error-control coding scheme used in the NASA Tracking Data Relay Satellite System.

and $g_2(D) = 1 + D^3 + D^4 + D^5 + D^6$ is used as the inner code. The convolutional inner code has free distance $d_{free} = 10$. The overall rate of the system is 0.437. The bit-error performance of this concatenated convolutional coding system is shown in Figure 15.12. This system (with infinite interleaving) achieves a BER of 10^{-6} at 2.53 dB of SNR. Comparing this concatenated convolutional coding system with the concatenated block coding system presented in Section 15.1, we see that the concatenated convolutional coding system has a 0.57 dB coding gain over the concatenated block coding system presented in Section 15.1; however, it has a lower rate, 0.437 compared with 0.545. If finite interleaving is used in the concatenated convolutional coding system, then the coding gain over the concatenated block coding system presented in Section 15.1 will be reduced.

15.7 BINARY CONCATENATION

So far we have considered only concatenation schemes in which the outer codes are nonbinary and the inner codes are binary; however, concatenation can also

be achieved with binary outer and inner codes. Let (n_1, k_1) and (n_2, k_2) be two binary codes, denoted C_1 and C_2, respectively. Suppose C_1 is used as the outer code, and C_2 is used as the inner code. In encoding, an information sequence k_1k_2 bits long is divided into k_2 subsequences, each consisting of k_1 information bits. These k_2 subsequences are encoded into k_2 codewords in the outer code C_1 and are stored as a $k_2 \times n_1$ array in an interleaver buffer. Then, each column of the array is encoded into a codeword in the inner code C_2, which is transmitted as it is formed. In decoding, each received sequence of n_2 symbols is decoded based on the inner code C_2. The decoded k_2 information bits are stored as a column in a deinterleaver buffer. After n_1 inner code decodings, a $k_2 \times n_1$ array is formed in the deinterleaver buffer. Then, each row of the array is decoded based on the outer code C_1. At the completion of each outer decoding, the k_1 decoded information bits are delivered to the user. Again, to achieve good error performance while maintaining low decoding complexity, we use a short inner code and decode it with a soft-decision decoding algorithm, and we use a long powerful binary code as the outer code and decode it with a hard-decision algebraic decoding algorithm. The binary concatenation scheme described here is actually the product coding scheme presented in Section 4.7.

If the inner code has a simple trellis, it can be decoded with a soft-input and soft-output decoding algorithm, such as the MAP algorithm (or its simplified version), discussed in Chapters 12 and 14. The reliability information of each decoded symbol provided by the inner code decoder can be used as soft-input information to the outer code decoder for soft-decision decoding of the outer code. Using soft-decision decoding for both inner and outer code decoders significantly improves the error performance of the concatenated coding system, but the decoding complexity also increases significantly. To reduce the decoding complexity, a simple reliability-based algebraic soft-decision decoding algorithm presented in Chapter 10, such as the Chase-2 algorithm or the ordered statistic decoding algorithm, can be used for decoding the outer code.

If decoding complexity and decoding delay are not critical issues, the error performance of a binary concatenated coding scheme can be further improved by using soft-input and soft-output decoding for both inner and outer code decoders and performing the inner and outer decodings iteratively. During each decoding iteration, the inner code decoder provides the outer code decoder with reliability information of the decoded symbols; this reliability information is then used as the soft input for the outer code decoder. Based on this soft-input information (together with the channel information), the outer code decoder performs soft-output decoding. The reliability information of the decoded symbols is then fed back to the input of the inner code decoder as a soft input to start the next decoding iteration. Decoding iterations continue until a certain stopping criterion is satisfied. To prevent successive decoding iterations from becoming correlated, the interleaver at the encoding side pseudorandomly permutes the bits in the code array after the outer code encoding. The permuted array is then encoded by the inner code encoder. At the decoding side, after inner code decoding, the decoded symbols must be permuted inversely (inverse permutation) to allow the outer code decoder to perform outer code decoding. To start the next decoding iteration, the decoded symbols at the end of the outer code decoding are permuted back to the

original form for the inner code decoding. This alternate permutation and inverse permutation is performed in each decoding iteration. Binary concatenation with this type of iterative decoding results in amazingly good error performance very close to the Shannon limit—of course, at the expense of decoding complexity and decoding delay.

The two-dimensional product codes without the checks on checks presented in Section 4.7 are quite suitable for the described type of iterative decoding. After the row (or column) encoding, the information bits of the information array are permuted pseudorandomly before the column (or row) encodings. This permutation allows the two sets of parity bits to provide two sets of uncorrelated estimates for the same set of information bits with iterative decoding. Row and column decodings are carried out alternately in an iterative manner.

The binary concatenation described here is in serial form; however, it can also be implemented in parallel form, in which the information sequence is encoded by two encoders independently using a pseudorandom interleaver. This encoding generates two independent sets of parity bits for the same information sequence. At the decoding side, iterative decoding is performed by two decoders based on these two sets of parity bits. Parallel concatenation is usually implemented using two convolutional encoders.

Binary concatenated coding schemes in parallel form using pseudorandom interleaving and iterative decoding, commonly called *turbo coding*, is the subject of Chapter 16.

PROBLEMS

15.1 Prove that the concatenation of an (n_1, k_1) inner code with minimum distance d_1 and an (n_2, k_2) outer code with minimum distance d_2 has a minimum distance of at least $d_1 d_2$.

15.2 Prove the lower bound of the minimum distance of an m-level concatenated code given by 15.12.

15.3 Consider the concatenation of a RS outer code over $GF(2^m)$ and the binary $(m+1, m, 2)$ single parity-check inner code. Devise an error-erasure decoding for this concatenated code. [*Hint:* During the inner code decoding, if parity failure is detected in $m+1$ received bits, an erasure is declared. If no parity failure is detected, the parity bit is removed to form a symbol in $GF(2^m)$].

15.4 Form a 5-level concatenated code with a minimum distance of 16 using RM codes of length 16 to form inner codes. Choose either binary or RS codes (or shortened RS codes) of length 16 as outer codes to maximize the overall code rate.

15.5 Decompose the $RM(2, 5)$ code into a 3-level concatenated code, and describe the trellis complexities of the component concatenated codes at the three levels.

15.6 Decompose the $RM(2, 6)$ code into a 3-level concatenated code, and give the trellis complexities of the component concatenated codes at the three levels.

15.7 Decode the $RM(2, 5)$ code with 3-stage decoding based on the decomposition obtained in Problem 17.5. Plot the bit- and block-error performances versus SNR.

15.8 Decode the $RM(2, 6)$ code with 3-stage decoding based on the decomposition obtained in Problem 17.6. Plot the bit- and block-error performances versus SNR.

15.9 Repeat Problem 17.7 with the IMS-MLD algorithm.

15.10 Repeat Problem 17.8 with the IMS-MLD algorithm.

BIBLIOGRAPHY

1. G. D. Forney, Jr., *Concatenated Codes*, MIT Press, Cambridge, 1966.

2. J. Justesen, "A Class of Constructive Asymptotically Good Algebraic Codes," *IEEE Trans. Inform. Theory*, IT-18; 652–56, September 1972.

3. H. T. Moorthy, S. Lin, and G. Uehara, "Good Trellises for IC Implementation of Viterbi Decoders for Linear Block Codes," *IEEE Trans. Commun.*, 45: 52–63, January 1997.

4. E. Nakamura, G. Uehara, C. Chu, and S. Lin, "A 755 Mb/s Viterbi Decoder for $RM(64, 35, 8)$ Subcode," *Proc. IEEE Intl. Solid-State Circuit Conf.*, San Francisco, Calif., February 15–17, 1999.

5. E. L. Blokh and V. V. Zyablov, *Generalized Concatenated Codes*, Moscow, USSR, Nauka, 1976.

6. V. A. Zinoviev, "Generalized Cascade Codes," Probl. Peredachi Inform., 12: 5–15, 1976.

7. S. Hirasawa, M. Kasahara, Y. Sugiyama, and T. Namekawa, "Certain Generalization of Concatenated Codes—Exponential Error Bounds and Decoding Complexity," *IEEE Trans. Inform. Theory*, IT-26: 527–34, September 1980.

8. T. Kasami, T. Fujiwara, and S. Lin, "A Concatenated Coding Scheme for Error Control," *IEEE Trans. Commun.*, 34: 481–88, May 1986.

9. R. Morelos-Zaragoza, T. Fujiwara, T. Kasami, and S. Lin, "Construction of Generalized Concatenated Codes and Their Trellis-Based Decoding Complexity," *IEEE Trans. Inform. Theory*, 45: 725–31, March 1999.

10. A. R. Calderbank, "Multilevel Codes and Multistage Decoding," *IEEE Trans. Commun.*, 37: 222–29, March 1989.

11. G. D. Forney, Jr., "Coset Codes—Part II: Binary Lattices and Related Codes," *IEEE Trans. Inform. Theory*, 34: 1152–87, September 1988.

12. F. Hemmati, "Closest Coset Decoding of $|\mathbf{u}|\mathbf{u}+\mathbf{v}|$ Codes," *IEEE J. Select. Areas Commun.*, 7: 982–88, August 1989.

13. U. Dettmar, J. Portugheis and H. Hentsch, "New Multistage Decoding Algorithm," *Electronics Letter,* 28 (no. 7): 635–36, 1992.

14. J. Wu, S. Lin, T. Kasami, T. Fujiwara, and T. Takata, "An Upper Bound on the Effective Error Coefficient of Two-Stage Decoding, and Good Two-Level Decomposition of Some Reed–Muller Codes," *IEEE Trans. Commun.*, 42: 813–18, February/March/April, 1994.

15. T. Takata, Y. Yamashita, T. Fujiwara, T. Kasami, and S. Lin,"Suboptimum Decoding of Decomposable Codes," *IEEE Trans. Inform. Theory*, 40: 1392–1405, September 1944.

16. G. Schnabl and M. Bossert, "Soft-Decision Decoding of Reed–Muller Codes as Generalized Multiple Concatenated Codes," *IEEE Trans. Inform. Theory*, 41: 304–8, January 1995.

17. D. Stojanovic, M. P. C. Fossorier, and S. Lin, "Iterative Multistage Maximum Likelihood Decoding of Multilevel Codes," *Proc. Coding and Cryptography*, pp. 91–101, Paris, France, January 11–14, 1999.

18. D. Stojanovic, S. Lin, and M. P. C. Fossorier, "Iterative Multistage Decoding of BCM Codes," *Proc 2000 IEEE Intl. Symp. Inform. Theory*, p. 248, Sorrento, Italy, June 25–30, 2000.

19. D. Stojanovic, "Iterative Multistage Maximum Likelihood Decoding Algorithm for Multilevel Codes and its Applications," Ph.D. dissertation, Dept. of Electrical Engineering, University of Hawaii, Honolulu, June 2003.

20. N. Seshadri and C. W. Sundberg, "List Viterbi Decoding Algorithms with Applications," *IEEE Trans. Commun.*, 42: 313–22, February/March/April, 1994.