Module 13 - Forward Error Correction (FEC)

Dr. Ivan Djordjevic

Assistant Professor, Electrical and Computer Engineering Dept, University of Arizona

Dr. Djordjevic is an Assistant Professor of Electrical and Computer Engineering at the University of Arizona. His current research interests include optical networks, error control coding, constrained coding, coded modulation, turbo equalization, OFDM applications, and quantum error correction. He is an author/co-author of OFDM for Optical Communications (Elsevier, Oct. 2009) and Coding for Optical Channels (Springer, Mar.2010). Dr. Djordjevic is also an author of over 100 journal publications and over 100 conference papers.

Email: ivan@email.arizona.edu

Introduction

This module describes different forward error correction (FEC) schemes currently in use or suitable for use in optical communication systems. We start with description of standard block codes. The state-of-the-art in optical communication systems standardized by the International Telecommunication Union-Telecommunication Standardization Sector **(**ITU-T) employ concatenated Bose-Ray-Chaudhuri-Hocquenghem (BCH) / Reed-Solomon (RS) codes [1],[2]. The RS(255,239) in particular has been used in a broad range of long-haul communication systems [1],[2], and it is commonly considered as the first-generation of FEC [3],[4]. The elementary FEC schemes (BCH, RS or convolutional codes) may be combined to design more powerful FEC schemes, e.g. RS(255,239)+RS(255,233). Several classes of concatenation codes are listed in ITU-T G975.1. Different concatenation schemes, such as the concatenation of two RS codes or the concatenation of RS and convolutional codes, are commonly considered as second generation of FEC [3],[4].

In recent years, iteratively decodable codes, such as turbo codes [3]-[10] and low-density parity-check (LDPC) codes [11]-[20], have generated significant research attention. In [5] Sab and Lemarie proposed a forward error correction (FEC) scheme based on block turbo code for long-haul dense wavelength division multiplexing (DWDM) optical transmission systems. In several recent papers [12]-[18], we have shown that iteratively decodable LDPC codes outperform turbo product codes in bit-error rate (BER) performance. The decoder complexity of these codes is comparable (or lower) to that of turbo product codes, and significantly lower than that of serial/parallel concatenated turbo codes. For reasons mentioned above, LDPC code is a viable and attractive choice for the FEC scheme of a 40 Gb/s optical transmission systems. The soft iteratively decodable codes, turbo and LDPC codes, are commonly referred to as the third

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

generation of FEC [3],[4]. This chapter is devoted to the first and the second generation of FEC for optical communications. For more details on the third generation of FEC for optical channels an interested reader is referred to [46].

13.1 Channel Coding Preliminaries

Two key system parameters are transmitted power and channel bandwidth, which together with additive noise sources determine the signal-to-noise ratio (SNR) and correspondingly BER. In practice, we very often come into situation when the target BER cannot be achieved with a given modulation format. For the fixed SNR, the only practical option to change the data quality transmission from unacceptable to acceptable is through use of *channel coding*. Another practical motivation of introducing the channel coding is to reduce required SNR for a given target BER. The amount of energy that can be saved by coding is commonly described by coding gain. *Coding gain* refers to the savings attainable in the energy per information bit to noise spectral density ratio (E_b/N_0) required to achieve a given bit error probability when coding is used compared to that with no coding. A typical digital optical communication system employing channel coding is shown in **Figure 13.1**. The discrete source generates the information in the form of sequence of symbols. The channel encoder accepts the message symbols and adds redundant symbols according to a corresponding prescribed rule. The channel coding is the act of transforming a length-*k* sequence into a length-*n* codeword. The set of rules specifying this transformation are called the channel code, which can be represented as the following mapping

$$C: M\text{->}X, \qquad \textit{(Equation 13.1)}$$

where *C* is the channel code, *M* is the set of information sequences of length *k,* and *X* is the set of codewords of length *n.* The decoder exploits these redundant symbols to determine which message symbol was actually transmitted. Encoders and decoders consider the whole digital transmission system as a discrete channel. Other blocks shown in **Figure 13.1**, are already explained in previous chapters, here we are concerned with channel encoders and decoders. Different classes of channel codes can be categorized into three broad categories: (i) *error detection* in which we are concerned only with detecting the errors that occurred during transmission (examples include automatic request for transmission-ARQ), (ii) *forward error correction* (FEC), where we are interested in correcting the errors that occurred during transmission, and (iii) hybrid channel codes that combine the previous two approaches. In this chapter we are concerned only with FEC.

The key idea behind the forward error correcting codes is to add extra redundant symbols to the message to be transmitted, and use those redundant symbols in the decoding procedure to correct the errors introduced by the channel. The redundancy can be introduced in time, frequency or space domain. For example, the redundancy in time domain is introduced if the same message is transmitted at least twice, the technique is known as the *repetition code*. The space redundancy is

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

used as a means for achieving high spectrally efficient transmission, in which the modulation is combined with error control.

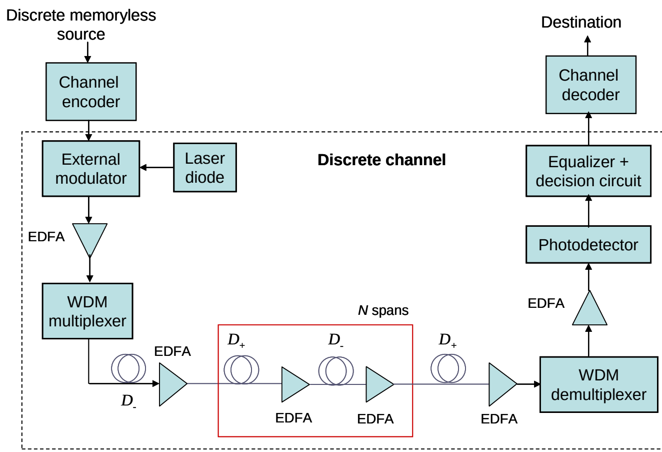

Figure 13.1 *Block diagram of a point-to-point digital optical communication system.*

The codes commonly considered in fiber-optics communications belong either to the class of *block codes* or to the class of *convolutional codes*. In an (n,k) *block code* the channel encoder accepts information in successive k-symbol blocks, adds n-k redundant symbols that are algebraically related to the k message symbols; thereby producing an overall encoded block of n symbols (n>k), known as a *codeword*. If the block code is *systematic*, the information symbols stay unchanged during the encoding operation, and the encoding operation may be considered as adding the n-k generalized parity checks to k information symbols. Since the information symbols are statistically independent (a consequence of source coding or scrambling), the next codeword is independent of the content of the current codeword. The *code rate* of an (n,k) block code is defined as R=k/n, and *overhead* by OH=(1/R-1)·100%. In *convolutional code*, however, the encoding operation may be considered as the discrete-time convolution of the input sequence with the impulse response of the encoder. Therefore, the n-k generalized parity checks are functions of not only k information symbols but also the functions of m previous k-tuples, with m+1 being the encoder impulse response length. The statistical dependence is introduced to the window of length n(m+1), the parameter known as *constraint length* of convolutional codes.

Example 1: *Repetition Code*. In repetition code each bit is transmitted n=2m+1 times. For example, for n=3 the bits 0 and 1 are represented as 000 and 111, respectively. On receiver side we first perform threshold decision, if the received sample is the above threshold we decide in favor of 1, otherwise in favor of bit 0. The decoder then applies the following *majority decoding rule*: if in block of n bits the number of ones exceeds the number of zeros, decoder decides in favor of 1; otherwise in favor of 0. This code is capable of correcting up to m errors. The probability of error that remains upon decoding can be evaluated by the following expression:

$$P_e = \sum_{i=m+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}, \qquad \textit{(Equation 13.2)}$$

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

where p is the probability of making an error on a given position. In **Figure 13.2** we illustrate the importance of channel coding by using this trivial example. We show how the BER remains after decoding against the code rate $R=1/n$ for different values of channel BERs p. It is interesting to notice that a target BER of 10^{-15} can be achieved even with this simple code, but the code rate is unacceptably low.

Example 2: *Hamming codes*. The Hamming codes are single error-correcting codes, for which the code parameters (n,k) satisfy the following inequality: $2^{n-k}\geq n+1$. In Hamming codes, the $(n-k)$ parity bits are located on positions 2^j (j=0,1,…,n-k-1), while the information bits are located on remaining bit positions. To identify the location of error we determine the syndrome ("checking number"). For (7,4) Hamming code, the codeword can be represented by: $p_1p_2i_1p_3i_2i_3i_4$, where p_j (j=1,2,3) are parity bits, and i_j (j=1,2,3,4) are information bits. For information bits 1101, the codeword is obtained by $p_1p_21p_3101$, and the parity bits are determined as:

$$\begin{aligned} p_1 &= i_1 + i_2 + i_4 = 1 \\ p_2 &= i_1 + i_3 + i_4 = 0 \\ p_3 &= i_2 + i_3 + i_4 = 0 \end{aligned} \qquad \textit{(Equation 13.3)}$$

The resulting codeword is $x_1x_2x_3x_4x_5x_6$ =1010101. Let's assume that bit six was received incorrectly, the corresponding word will be: $y_1y_2y_3y_4y_5y_6$ =10101**1**1. The syndrome computation proceeds as follows:

$$\begin{aligned} s_1 &= y_1 + y_3 + y_5 + y_7 = 1+1+1+1 = 0 \\ s_2 &= y_2 + y_3 + y_6 + y_7 = 0+1+\mathbf{1}+1 = 1 \\ s_3 &= y_4 + y_5 + y_6 + y_7 = 0+1+\mathbf{1}+1 = 1 \\ S&\left(s_3, s_2, s_1\right) = 110_2 = 6 \end{aligned} \qquad \textit{(Equation 13.4)}$$

The syndrome computes the error's location, which is 6 in this example. The located error can simply be corrected by flipping the content of corresponding bit. The syndrome-check s_1 performs parity checks on bit positions having 1 at position 2^0 in binary representation: 001_2=1, 011_2=3, 101_2=5, 111_2=7. On the other hand, the syndrome-check s_2 performs parity checks on bit positions having 1 at position 2^1 in binary representation: 010_2=2, 011_2=3, 110_2=6, 111_2=7. Finally, the syndrome-check s_3 performs parity checks on bit positions having 1 at position 2^2 in binary representation: 100_2=4, 101_2=5, 110_2=6, 111_2=7.

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

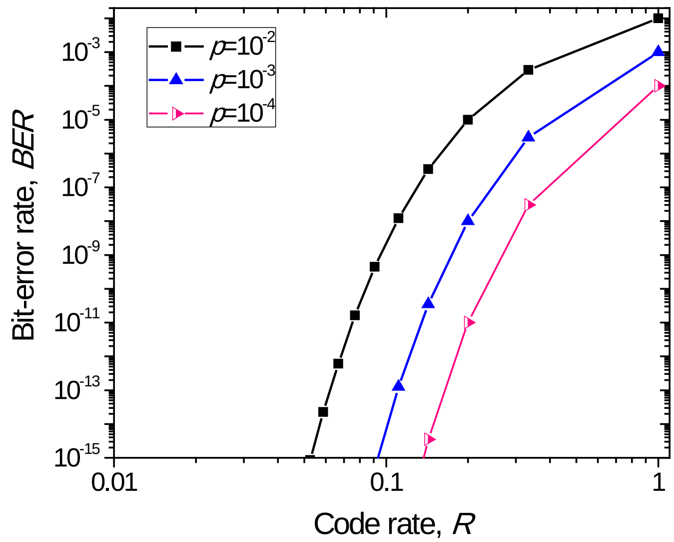

Figure 13.2 *Illustrating the importance of channel coding.*

In the rest of this Section is an elementary introduction to linear block codes, cyclic codes, RS codes, concatenated codes, and product codes. These classes of codes are already employed in fiber-optics communication systems. For a detailed treatment of different error-control coding schemes an interested reader is referred to [21]-[26],[32],[46]-[48].

13.2 Linear Block Codes

The *linear block code* (n,k), using the language of vector spaces, can be defined as a subspace of a vector space over finite field GF(q), with q being the prime power. Every space is described by its *basis* - a set of linearly independent vectors. The number of vectors in the basis determines the dimension of the space. Therefore, for an (n,k) linear block code the dimension of the space is n, and the dimension of the code subspace is k.

Example 3: $(n,1)$ repetition code. The repetition code has two code words $\boldsymbol{x}_0$=(00 … 0) and $\boldsymbol{x}_1$=(11 … 1). Any linear combination of these two code words is another code word as shown below

$$\begin{aligned}\boldsymbol{x}_0 + \boldsymbol{x}_0 &= \boldsymbol{x}_0\\ \boldsymbol{x}_0 + \boldsymbol{x}_1 &= \boldsymbol{x}_1 + \boldsymbol{x}_0 = \boldsymbol{x}_1\\ \boldsymbol{x}_1 + \boldsymbol{x}_1 &= \boldsymbol{x}_0\end{aligned}$$ ***(Equation 13.5)***

The set of code words from a linear block code forms a group under the addition operation, because the all-zero code word serves as the identity element, and the code word itself serves as the inverse element. This is the reason why the linear block codes are also called the group codes. The linear block code (n,k) can be observed as a k-dimensional subspace of the vector space of all n-tuples over the binary filed GF(2)={0,1}, with addition and multiplication rules given in Table 1. All n-tuples over GF(2) form the vector space. The sum of two n-tuples $\boldsymbol{a}=(a_1\ a_2\ \ldots\ a_n)$

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

and $\boldsymbol{b}=(b_1\ b_2\ \ldots\ b_n)$ is clearly an n-tuple and commutative rule is valid because $\boldsymbol{c}=\boldsymbol{a}+\boldsymbol{b}=(a_1+b_1\ a_2+b_2\ \ldots\ a_n+b_n)=(b_1+a_1\ b_2+a_2\ \ldots\ b_n+a_n)=\boldsymbol{b}+\boldsymbol{a}$. The all-zero vector $\mathbf{0}=(0\ 0\ \ldots\ 0)$ is the identity element, while n-tuple $\boldsymbol{a}$ itself is the inverse element $\boldsymbol{a}+\boldsymbol{a}=0$. Therefore, the n-tuples form the Abelian group with respect to the addition operation. The scalar multiplication is defined by: $\alpha\boldsymbol{a}=(\alpha\, a_1\ \alpha a_2\ \ldots\ \alpha a_n)$, $\alpha\in GF(2)$. The distributive laws

$$\alpha(\boldsymbol{a}+\boldsymbol{b})=\alpha\boldsymbol{a}+\alpha\boldsymbol{b} \qquad \textit{(Equation 13.6)}$$

$$(\alpha+\beta)\boldsymbol{a}=\alpha\boldsymbol{a}+\beta\boldsymbol{a},\ \forall\ \alpha,\beta\in GF(2) \qquad \textit{(Equation 13.7)}$$

are also valid. The associate law $(\alpha\cdot\beta)\boldsymbol{a}=\alpha\cdot(\beta\boldsymbol{a})$ is clearly satisfied. Therefore, the set of all n-tuples is a vector space over GF(2). The set of all code words from an (n,k) linear block code forms an abelian group under the addition operation. It can be shown, in a fashion similar to that above, that all code words of an (n,k) linear block codes form the vector space of dimensionality k. There exist k basis vectors (code words) such that every code word is a linear combination of these code words.

Example 4: $(n,1)$ repetition code: $C=\{(0\ 0\ \ldots\ 0), (1\ 1\ \ldots\ 1)\}$. Two code words in C can be represented as linear combination of all-ones basis vector: $(11\ \ldots\ 1)=1\cdot(11\ \ldots\ 1)$, $(00\ \ldots\ 0)=1\cdot(11\ \ldots\ 1)+1\cdot(11\ \ldots\ 1)$.

Table 1: Addition (+) and multiplication (·) rules in GF(2).

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

Generator Matrix for Linear Block Code

Any code word $\boldsymbol{x}$ from the (n,k) linear block code can be represented as a linear combination of k basis vectors $\boldsymbol{g}_i$ $(i=0,1,..,k\text{-}1)$ as given below:

$$\boldsymbol{x}=m_0\boldsymbol{g}_0+m_1\boldsymbol{g}_1+\ldots+m_{k-1}\boldsymbol{g}_{k-1}=\boldsymbol{m}\begin{bmatrix}\boldsymbol{g}_0\\ \boldsymbol{g}_1\\ \ldots\\ \boldsymbol{g}_{k-1}\end{bmatrix}=\boldsymbol{mG};\ \ \boldsymbol{G}=\begin{bmatrix}\boldsymbol{g}_0\\ \boldsymbol{g}_1\\ \ldots\\ \boldsymbol{g}_{k-1}\end{bmatrix},\ \boldsymbol{m}=\begin{pmatrix}m_0 & m_1 & \ldots & m_{k-1}\end{pmatrix} \qquad \textit{(Equation 13.8)}$$

where $\boldsymbol{m}$ is the message vector, and $\boldsymbol{G}$ is the generator matrix (of dimensions kxn), in which every row represents a vector from the coding subspace. Therefore, in order to encode, the message vector $\boldsymbol{m}(m_0,m_1,\ldots,m_{k-1})$ has to be multiplied with a generator matrix $\boldsymbol{G}$ to get $\boldsymbol{x}=\boldsymbol{mG}$, where $\boldsymbol{x}(x_0,x_1,\ldots,x_{n-1})$ is a codeword.

Example 5: Generator matrices for repetition $(n,1)$ code $\boldsymbol{G}_{\text{rep}}$ and $(n,n\text{-}1)$ single-parity-check code $\boldsymbol{G}_{\text{par}}$ are given respectively as

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$\boldsymbol{G}_{\mathrm{rep}} = [11...1] \qquad \boldsymbol{G}_{\mathrm{par}} = \begin{bmatrix} 100...01 \\ 010...01 \\ ... \\ 000...11 \end{bmatrix}. \qquad \textit{(Equation 13.9)}$$

By elementary operations on rows in the generator matrix, the code may be transformed into systematic form

$$\boldsymbol{G}_s = \left[\boldsymbol{I}_k \mid \boldsymbol{P}\right], \qquad \textit{(Equation 13.10)}$$

where $\boldsymbol{I}_k$ is unity matrix of dimensions kxk, and $\boldsymbol{P}$ is the matrix of dimensions kx(n-k) with columns denoting the positions of parity checks

$$\boldsymbol{P} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ \cdots & & \cdots & \cdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}. \qquad \textit{(Equation 13.11)}$$

The codeword of a systematic code is obtained by

$$\boldsymbol{x} = [\boldsymbol{m} \mid \boldsymbol{b}] = \boldsymbol{m}\left[\boldsymbol{I}_k \mid \boldsymbol{P}\right] = \boldsymbol{mG}, \quad \boldsymbol{G} = \left[\boldsymbol{I}_k \mid \boldsymbol{P}\right], \qquad \textit{(Equation 13.12)}$$

and the structure of systematic codeword is shown in **Figure 13.3**.

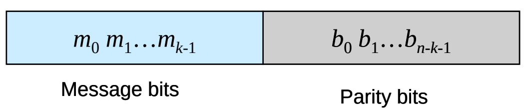

Figure 13.3 *Structure of systematic code word.*

Therefore, during encoding the message vector stays unchanged and the elements of vector of parity checks $\boldsymbol{b}$ are obtained by

$$b_i = p_{0i}m_0 + p_{1i}m_1 + \ldots + p_{k-1,i}m_{k-1}, \qquad \textit{(Equation 13.13)}$$

where

$$p_{ij} = \begin{cases} 1, & \text{if } b_i \text{ depends on } m_j \\ 0, & \text{otherwise} \end{cases}. \qquad \textit{(Equation 13.14)}$$

During transmission an optical channel introduces the errors so that the received vector $\boldsymbol{r}$ can be written as $\boldsymbol{r}$=$\boldsymbol{x}$+$\boldsymbol{e}$, where $\boldsymbol{e}$ is the error vector (pattern) with elements components determined by

$$e_i = \begin{cases} 1 & \text{if an error occured in the } i\text{th location} \\ 0 & \text{otherwise} \end{cases} \qquad \textit{(Equation 13.15)}$$

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

To determine whether the received vector ***r*** is a codeword vector, we are introducing the concept of a *parity check matrix*.

Parity-Check Matrix for Linear Block Code

Another useful matrix associated with the linear block codes is the parity-check matrix. Let us expand the matrix equation ***x***=***mG*** in scalar form as follows:

$$
\begin{aligned}
x_0 &= m_0 \\
x_1 &= m_1 \\
&\dots \\
x_{k-1} &= m_{k-1} \\
x_k &= m_0 p_{00} + m_1 p_{10} + \dots + m_{k-1} p_{k-1,0} \\
x_{k+1} &= m_0 p_{01} + m_1 p_{11} + \dots + m_{k-1} p_{k-1,1} \\
&\dots \\
x_{n-1} &= m_0 p_{0,n-k-1} + m_1 p_{1,n-k-1} + \dots + m_{k-1} p_{k-1,n-k-1}
\end{aligned}
\qquad \textit{(Equation 13.16)}
$$

By using the first k equalities, the last n-k equations can be rewritten as follows:

$$
\begin{aligned}
& x_0 p_{00} + x_1 p_{10} + \dots + x_{k-1} p_{k-1,0} + x_k = 0 \\
& x_0 p_{01} + x_1 p_{11} + \dots + x_{k-1} p_{k-1,0} + x_{k+1} = 0 \\
& \dots \\
& x_0 p_{0,n-k+1} + x_1 p_{1,n-k-1} + \dots + x_{k-1} p_{k-1,n-k+1} + x_{n-1} = 0
\end{aligned}
\qquad \textit{(Equation 13.17)}
$$

The matrix representation of (8) is given below:

$$
\begin{bmatrix} x_0 & x_1 & \dots & x_{n-1} \end{bmatrix}
\begin{bmatrix}
p_{00} & p_{10} & \dots & p_{k-1,0} & 1 & 0 & \dots & 0 \\
p_{01} & p_{11} & \dots & p_{k-1,1} & 0 & 1 & \dots & 0 \\
\dots & & \dots & & \dots & & & \dots \\
p_{0,n-k-1} & p_{1,n-k-1} & \dots & p_{k-1,n-k-1} & 0 & 0 & \dots & 1
\end{bmatrix}^T
= \boldsymbol{x}\begin{bmatrix} \boldsymbol{P}^T & \boldsymbol{I}_{n-k} \end{bmatrix} = \boldsymbol{x}\ {}^{T}\boldsymbol{H}\mathbf{0}, \quad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{P}^T & \boldsymbol{I}_{n-k} \end{bmatrix}_{(n-k)xn}
$$

(Equation 13.18)

The ***H***-matrix in (9) is known as the parity-check matrix. We can easily verify that:

$$
\boldsymbol{G}\boldsymbol{H}^T = \begin{bmatrix} \boldsymbol{I}_k & \boldsymbol{P} \end{bmatrix} \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{I}_{n-k} \end{bmatrix} = \boldsymbol{P} + \boldsymbol{P} = \mathbf{0}, \qquad \textit{(Equation 13.19)}
$$

meaning that the parity check matrix of an (n,k) linear block code ***H*** is a matrix of rank n-k and dimensions (n-k)xn whose null-space is k-dimensional vector with basis being the generator matrix ***G***.

<u>Example 6</u>: Parity-Check Matrices for (n,1) repetition code $\boldsymbol{H}_{\text{rep}}$ and (n,n-1) single-parity check code $\boldsymbol{H}_{\text{par}}$ are given respectively as:

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$\boldsymbol{H}_{\text{rep}} = \begin{bmatrix} 100...01 \\ 010...01 \\ ... \\ 000...11 \end{bmatrix} \qquad \boldsymbol{H}_{\text{par}} = [11...1]. \qquad \textit{(Equation 13.20)}$$

Example 7: For Hamming (7,4) code the generator ***G*** and parity check ***H*** matrices are given respectively as

$$\boldsymbol{G} = \begin{bmatrix} 1000|110 \\ 0100|011 \\ 0010|111 \\ 0001|101 \end{bmatrix} \qquad \boldsymbol{H} = \begin{bmatrix} 1011|100 \\ 1110|010 \\ 0111|001 \end{bmatrix}$$

(Equation 13.21)

Every (n,k) linear block code with generator matrix ***G*** and parity-check matrix ***H*** has a dual code with generator matrix ***H*** and parity check matrix ***G***. For example, (n,1) repetition and (n,n-1) single-parity check codes are dual.

Distance Properties of Linear Block Codes

To determine the *error correction capability* of the code we have to introduce the concept of Hamming distance and Hamming weight. *Hamming distance* between two codewords $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, $d(\boldsymbol{x}_1,\boldsymbol{x}_2)$, is defined as the number of locations in which their respective elements differ. *Hamming weight*, $w(\boldsymbol{x})$, of a codeword vector $\boldsymbol{x}$ is defined as the number of nonzero elements in the vectors. The *minimum distance*, d_{min}, of a linear block code is defined as the smallest Hamming distance between any pair of code vectors in the code. Since the zero-vector is a codeword, the minimum distance of a linear block code can be determined simply as the smallest Hamming weight of the nonzero code vectors in the code. Let the parity check matrix be written as $\boldsymbol{H}=[\boldsymbol{h}_1\ \boldsymbol{h}_2\ \ldots\ \boldsymbol{h}_n]$, where $\boldsymbol{h}_i$ is the ith column in ***H***. Since every codeword $\boldsymbol{x}$ must satisfy the syndrome equation, $\boldsymbol{x}\boldsymbol{H}^{\text{T}}=\boldsymbol{0}$ (see **Equation 13.18)**the minimum distance of a linear block code is determined by the minimum number of columns of the ***H***-matrix whose sum is equal to the zero vector.

The codewords can be represented as points in n-dimensional space, as shown in **Figure 13.4.** Decoding process can be visualized by creating the spheres of radius t around codeword points. The received word vector $\boldsymbol{r}$ in **Figure 13.4(a)** will be decoded as a codeword $\boldsymbol{x}_i$ because its Hamming distance $d(\boldsymbol{x}_i,\boldsymbol{r})\leq t$ is closest to the codeword $\boldsymbol{x}_i$. On the other hand, as shown in **Figure 13.4(b)** the Hamming distance $d(\boldsymbol{x}_i,\boldsymbol{x}_j)\leq 2t$ and the received vector $\boldsymbol{r}$ that falls in the intersecting area of two spheres cannot be uniquely decoded.

Therefore, an (n,k) linear block code of minimum distance d_{min} can correct up to t errors if, and only if, $t\leq\lfloor 1/2(d_{\text{min}}-1)\rfloor$ (where $\lfloor\ \rfloor$ denotes the largest integer less than or equal to the enclosed quantity) or equivalently $d_{\text{min}}\geq 2t+1$. If we are only interested in detecting e_{d} errors then $d_{\text{min}}\geq e_{\text{d}}+1$. Finally, if we are interested in detecting e_{d} errors and correcting e_{c} errors then

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$d_{min} \geq e_d + e_c + 1$. The Hamming (7,4) code is therefore a single error correcting and double error detecting code. More generally, a family of (n,k) linear block codes with following parameters:

- Block length: $n=2^m-1$
- Number of message bits: $k=2^m-m-1$
- Number of parity bits: $n-k=m$
- $d_{min}=3$

where $m \geq 3$, are known as *Hamming* codes. Hamming codes belong to the class of perfect codes, the codes that satisfy the Hamming inequality below with equality sign [30],[35]:

$$2^{n-k} \geq \sum_{i=0}^{t} \binom{n}{i}. \qquad \textbf{\textit{(Equation 13.22)}}$$

This bound gives how many errors t can be corrected with an (n,k) linear block code by using the syndrome decoding (described in section 4.2.6).

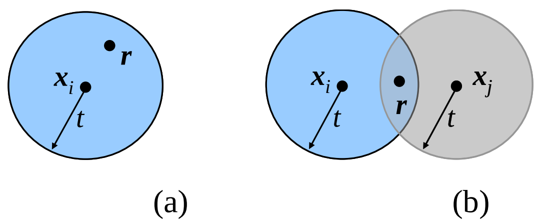

Figure 13.4 *Illustration of Hamming distance: (a) $d(\mathbf{x}_i,\mathbf{x}_j) \geq 2t+1$, and (b) $d(\mathbf{x}_i,\mathbf{x}_j) < 2t+1$.*

Example 8: The parity-check matrix of Hamming (7,4) code is given in Example 7. By adding the first, fifth and 6th column we obtain a zero vector:

$$\begin{bmatrix}0\\1\\1\end{bmatrix}+\begin{bmatrix}0\\1\\0\end{bmatrix}+\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}, \qquad \textbf{\textit{(Equation 13.23)}}$$

which means that minimum distance is $d_{min}=3$ and code can correct all single errors.

Coding Gain

A very important characteristic of an (n,k) linear block code is so called coding gain, which was introduced in introductory section of this chapter as being the savings attainable in the energy per information bit to noise spectral density ratio (E_b/N_0) required to achieve a given bit error probability when coding is used compared to that with no coding. Let E_c denote the transmitted bit energy, and E_b denote the information bit energy. Since the total information word energy kE_b must be the same as the total codeword energy nE_c, we obtain the following relationship between E_c and E_b:

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$E_c=(k/n)E_b=RE_b. \qquad \textit{(Equation 13.24)}$$

The probability of error for BPSK on an AWGN channel, when coherent hard decision (bit-by-bit) demodulator is used, can be obtained as follows:

$$p=\frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E_c}{N_0}}\right)=\frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{RE_b}{N_0}}\right), \qquad \textit{(Equation 13.25)}$$

where erfc(x) function is defined by

$$\operatorname{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{+\infty}e^{-z^2}dz\cdot \qquad \textit{(Equation 13.26)}$$

For high SNRs the word error probability (remained upon decoding) of a t-error correcting code is dominated by a t+1 error event

$$P_w(e)\approx\binom{n}{t+1}p^{t+1}(1-p)^{n-t+1}\approx\binom{n}{t+1}p^{t+1}. \qquad \textit{(Equation 13.27)}$$

The bit error probability P_b is related to the word error probability by

$$P_b\approx\frac{2t+1}{n}P_w(e)\approx c(n,t)\,p^{t+1}, \qquad \textit{(Equation 13.28)}$$

because 2t+1 and more errors per codeword cannot be corrected and they can be located anywhere on n codeword locations, and $c(n,t)$ is a parameter dependent on error correcting capability t and codeword length n. By using the upper bound on erfc(x) we obtain

$$P_b\approx\frac{c(n,t)}{2}\left[\exp\left(\frac{-RE_b}{N_0}\right)\right]^{t+1}. \qquad \textit{(Equation 13.29)}$$

The corresponding approximation for uncoded case is

$$P_{b,\text{uncoded}}\approx\frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right). \qquad \textit{(Equation 13.30)}$$

By equating the **Equations (13.29) and (13.30)** and ignoring the parameter $c(n,t)$ we obtain the following expression for hard decision decoding coding gain

$$\frac{(E_b/N_0)_{\text{uncoded}}}{(E_b/N_0)_{\text{coded}}}\approx R(t+1)\cdot \qquad \textit{(Equation 13.31)}$$

The corresponding soft decision asymptotic coding gain of convolutional codes is [21], [23]-[25], [32]

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$\frac{\left(E_b / N_0\right)_{\text{uncoded}}}{\left(E_b / N_0\right)_{\text{coded}}} \approx R d_{\min}, \qquad \textit{(Equation 13.32)}$$

and it is about 3 dB better than hard decision decoding (because $d_{\min} \geq 2t+1$).

In optical communications it is very common to use the Q-factor as the figure of merit instead of SNR, which is related to the BER on an AWGN, as shown in previous chapters, as follows

$$\text{BER} = \frac{1}{2}\operatorname{erfc}\left(\frac{Q}{\sqrt{2}}\right). \qquad \textit{(Equation 13.33)}$$

Let BER_{in} denote the BER at the input of FEC decoder, let BER_{out} denote the BER at the output of FEC decoder, and let BER_{ref} denote target BER (such as either 10^{-12} or 10^{-15}). The corresponding coding gain GC and net coding gain NCG are respectively defined as [10]

$$\text{CG} = 20\log_{10}\left[\operatorname{erfc}^{-1}\left(2\text{BER}_{\text{ref}}\right)\right] - 20\log_{10}\left[\operatorname{erfc}^{-1}\left(2\text{BER}_{\text{in}}\right)\right] \ \ [\text{dB}], \qquad \textit{(Equation 13.34)}$$

$$\text{NCG} = 20\log_{10}\left[\operatorname{erfc}^{-1}\left(2\text{BER}_{\text{ref}}\right)\right] - 20\log_{10}\left[\operatorname{erfc}^{-1}\left(2\text{BER}_{\text{in}}\right)\right] + 10\log_{10} R \ \ [\text{dB}]. \qquad \textit{(Equation 13.35)}$$

All coding gains reported in this chapter are in fact NCG, although they are sometimes called the coding gains only, because this is the common practice in coding theory literature [21], [23]-[25], [32].

Syndrome Decoding and Standard Array

The received vector $\boldsymbol{r}=\boldsymbol{x}+\boldsymbol{e}$ ($\boldsymbol{x}$ is the codeword and $\boldsymbol{e}$ is the error patter introduced above) is a codeword if the following *syndrome equation* is satisfied $\boldsymbol{s}=\boldsymbol{r}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$. The syndrome has the following important properties:

1. The syndrome is only function of the error pattern. This property can easily be proved from definition of syndrome as follows: $\boldsymbol{s}=\boldsymbol{r}\boldsymbol{H}^{\mathrm{T}}=(\boldsymbol{x}+\boldsymbol{e})\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{x}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}}$.
2. All error pattern that differ by a codeword have the same syndrome. This property can also be proved from syndrome definition. Let $\boldsymbol{x}_i$ be the ith ($i=0,1,\ldots,2^{k-1}$) codeword. The set of error patterns that differ by a codeword is known as cosset: $\{\boldsymbol{e}_i=\boldsymbol{e}+\boldsymbol{x}_i;\ i=0,1,\ldots,2^{k-1}\}$. The syndrome corresponding to ith error pattern from this set $\boldsymbol{s}_i=\boldsymbol{r}_i\boldsymbol{H}^{\mathrm{T}}=(\boldsymbol{x}_i+\boldsymbol{e})\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{x}_i\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}}$ is only function of the error pattern, and therefore all error patterns from the cosset have the same syndrome.
3. The syndrome is a function of only those columns of a parity-check matrix corresponding to the error locations. The parity-check matrix can be written into following form: $\boldsymbol{H}=[\boldsymbol{h}_1 \ \ldots \ \boldsymbol{h}_n]$, where the ith element $\boldsymbol{h}_i$ denotes the ith column of $\boldsymbol{H}$. Based on syndrome definition for an error pattern $\boldsymbol{e}=[e_1\ldots e_n]$ the following is valid:

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^T = \begin{bmatrix} e_1 & e_2 & \dots & e_n \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1^T \\ \boldsymbol{h}_2^T \\ \dots \\ \boldsymbol{h}_n^T \end{bmatrix} = \sum_{i=1}^{n} e_i \boldsymbol{h}_i^T, \qquad \text{(Equation 13.36)}$$

which proves the claim of property 3.

4. With syndrome decoding an (n,k) linear block code can correct up to t errors, providing that the Hamming bound (11) is satisfied.

By using the property 2, 2^k codewords partition the space of all received words into 2^k disjoint subsets. Any received word within a subset will be decoded as the unique codeword. A *standard array* is a technique by which this partition can be achieved, and can be constructed using the following two steps [21],[23]-[25],[28],[30],[31],[32]:

1. Write down 2^k code words as elements of the first row, with the all-zero codeword as the leading element.
2. Repeat the steps 2(a) and 2(b) until all 2^n words are exhausted.
   (a) Out of the remaining unused n-tuples, select one with the least weight for the leading element of the next row (the *current row*).
   (b) Complete the current row by adding the leading element to each nonzero code word appearing in the first row and writing down the resulting sum in the corresponding column.

The standard array for an (n, k) block code obtained by this algorithm is illustrated in **Figure 13.5**. The columns represent 2^k disjoint sets, and every row represents the cosset of the code with leading elements being called the coset leaders.

$$\begin{array}{ccccccc}
\boldsymbol{x}_1 = 0 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \dots & \boldsymbol{x}_i & \dots & \boldsymbol{x}_{2^k} \\
\boldsymbol{e}_2 & \boldsymbol{x}_2 + \boldsymbol{e}_2 & \boldsymbol{x}_3 + \boldsymbol{e}_2 & \dots & \boldsymbol{x}_i + \boldsymbol{e}_2 & \dots & \boldsymbol{x}_{2^k} + \boldsymbol{e}_2 \\
\boldsymbol{e}_3 & \boldsymbol{x}_2 + \boldsymbol{e}_3 & \boldsymbol{x}_3 + \boldsymbol{e}_3 & \dots & \boldsymbol{x}_i + \boldsymbol{e}_2 & \dots & \boldsymbol{x}_{2^k} + \boldsymbol{e}_3 \\
\dots & \dots & & \dots & & \dots & \dots \\
\boldsymbol{e}_j & \boldsymbol{x}_2 + \boldsymbol{e}_j & \boldsymbol{x}_3 + \boldsymbol{e}_j & \dots & \boldsymbol{x}_i + \boldsymbol{e}_j & \dots & \boldsymbol{x}_{2^k} + \boldsymbol{e}_j \\
\dots & \dots & & \dots & & & \dots \\
\boldsymbol{e}_{2^{n-k}} & \boldsymbol{x}_2 + \boldsymbol{e}_{2^{n-k}} & \boldsymbol{x}_3 + \boldsymbol{e}_{2^{n-k}} & \dots & \boldsymbol{x}_i + \boldsymbol{e}_{2^{n-k}} & \dots & \boldsymbol{x}_{2^k} + \boldsymbol{e}_{2^{n-k}}
\end{array}$$

Figure 13.5 *The standard array architecture.*

<u>Example 9</u>: Standard array of (5,2) code C={(00000), (11010),(10101),(01111) } is given in Table 2. The parity-check matrix of this code is given by:

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$\boldsymbol{H}=\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}. \quad \textit{(Equation 13.37)}$$

Because the minimum distance of this code is 3 (first, second and fourth columns add to zero), this code is able to correct all single errors. For example, if the word 01010 is received it will be decoded to the top most codeword 11010 of column in which it lies. In the same Table corresponding syndromes are provided as well.

The syndrome decoding procedure is three step procedure [21],[23]-[25],[28],[30],[31],[32]:

1. For the received vector $\boldsymbol{r}$, compute the syndrome $\boldsymbol{s}=\boldsymbol{rH}^T$. From property 3 we can establish one-to-one correspondence the syndromes and error patterns (see Table 2), leading to the lookup table (LUT) containing the syndrome and corresponding error pattern (the cosset leader).
2. Within the cosset characterized by the syndrome $\boldsymbol{s}$, identify the cosset leader, say $\boldsymbol{e}_0$. The cosset leader corresponds to the error pattern with the largest probability of occurrence.
3. Decode the received vector as $\boldsymbol{x}=\boldsymbol{r}+\boldsymbol{e}_0$.

Table 2: Standard array of (5,2) code and corresponding decoding table.

	Codewords			
Coset leader	00000	11010	10101	01111
	00001	11011	10100	01110
	00010	11000	10111	01101
	00100	11110	10001	01011
	01000	10010	11101	00111
	10000	01010	00101	11111
	00011	11001	10110	01100
	00110	11100	10011	01001

Syndrome $\boldsymbol{s}$	Error pattern
000	00000
101	00001
110	00010
001	00100
010	01000
100	10000
011	00011
111	00110

<u>Example 10</u>: Let the received vector for (5,2) code example above be $\boldsymbol{r}$=(01010). The syndrome can be computed as $\boldsymbol{s}=\boldsymbol{rH}^T$=(100), and corresponding error pattern from LUT is found to be $\boldsymbol{e}_0$=(10000). The decoded word is obtained by adding the error pattern to received word $\boldsymbol{x}=\boldsymbol{r}+\boldsymbol{e}_0$=(11010), and the error on the first bit position is corrected.

<u>Example 11</u>: The decoding table for (7,4) Hamming code is given in Table 3. Let say that the message to be transmitted is $\boldsymbol{m}$=[1 0 0 1], the corresponding codeword is $\boldsymbol{x}=\boldsymbol{mG}^T$=[1 0 0 1 0 1 1], and let the channel introduce the error on 6th bit position. The received word is $\boldsymbol{r}$=[1 0 0 1 0 0 1]. The syndrome can be determined by

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$s = rH^{\mathrm{T}} = [1\ 0\ 0\ 1\ 0\ 1\ 1]\begin{bmatrix}110\\011\\111\\101\\100\\010\\001\end{bmatrix} = [010], \qquad \textit{(Equation 13.38)}$$

and from Table 3 we identify the error pattern ***e***=[0000010]. By adding this error pattern to the received word ***r***, we correct the error on 6th bit position.

Table 3: The decoding table of (7,4) Hamming code.

Syndrome ***s***	Error pattern
000	0000000
110	1000000
011	0100000
111	0010000
101	0001000
100	0000100
010	0000010
001	0000001

The standard array can be used to determine probability of word error as follows

$$P_w(e) = 1 - \sum_{i=0}^{n} \alpha_i p^i (1-p)^{n-i}, \qquad \textit{(Equation 13.39)}$$

where α_i is the number of cosset leaders of weight i (distribution of weights is also known as *weight distribution* of cosset leaders) and p is the probability of error (also known as crossover probability) of binary input binary output channel (binary symmetric channel, BSC). Any error pattern that is not a cosset leader will result in decoding error. For example, the weight distribution of cosset leaders in (5,2) code are α_0=1, α_1=5, α_2=2, α_i=0, i=3,4,5; which leads to the following word error probability:

$$P_w(e) = 1-(1-p)^5 - 5p(1-p)^4 - 2p^2(1-p)^3 |_{p=10^{-3}} = 7.986 \cdot 10^{-6}. \qquad \textit{(Equation 13.40)}$$

We can use the **Equation (13.29)** to estimate the coding gain of a given linear block code. For example, the word error probability for Hamming (7,4) code is

$$P_w(e) = 1-(1-p)^7 - 7p(1-p)^6 = \sum_{i=2}^{7} \binom{7}{i} p^i (1-p)^{7-i} \approx 21p^2. \qquad \textit{(Equation 13.41)}$$

In previous section we established the following relationship between bit and word error probabilities: $P_b \approx P_w(e)(2t+1)/n = (3/7)P_w(e) \approx (3/7)21p^2 = 9p^2$. Therefore, the crossover probability can be evaluated as

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$p=\sqrt{P_b}/3=(1/2)\operatorname{erfc}\left(\sqrt{\frac{RE_b}{N_0}}\right). \qquad \textit{(Equation 13.42)}$$

From this expression we can easily calculate the required SNR to achieve target P_b. By comparing such obtained SNR with corresponding SNR for uncoded BPSK we can evaluate the corresponding coding gain.

13.3 Cyclic Codes

The most commonly used class of linear block codes is the class of cyclic codes. Examples of cyclic codes include BCH codes, Hamming codes and Golay codes. RS codes are also cyclic but nonbinary codes. Even LDPC codes can be designed in cyclic or quasi-cyclic fashion.

Let us observe the vector space of dimension n. The subspace of this space is *cyclic code* if for any codeword $\boldsymbol{c}(c_0,c_1,\ldots,c_{n-1})$ arbitrary cyclic shift $\boldsymbol{c}_j(c_{n-j},c_{n-j+1},\ldots,c_{n-1},c_0,c_1,..,c_{n-j-1})$ is another codeword. With every codeword $\boldsymbol{c}(c_0,c_1,\ldots,c_{n-1})$ from a cyclic code, we associate the *codeword polynomial*

$$c(x)=c_0+c_1x+c_2x^2+\ldots+c_{n-1}x^{n-1}. \qquad \textit{(Equation 13.43)}$$

The jth cyclic shift, observed mod(x^n-1), is also a codeword polynomial

$$c^{(j)}(x)=x^jc(x)\operatorname{mod}\left(x^n-1\right). \qquad \textit{(Equation 13.44)}$$

It is straightforward to show that observed subspace is cyclic if composed from polynomials divisible by a polynomial $g(x)=g_0+g_1x+\ldots+g_{n-k}x^{n-k}$ that divides x^n-1 at the same time. The polynomial $g(x)$, of degree n-k, is called the *generator polynomial* of the code. If $x^n-1=g(x)h(x)$ then the polynomial of degree k is called the *parity-check polynomial*. The generator polynomial has the following three important properties [21],[23]-[25],[28],[30],[31],[32]:

1. The generator polynomial of an (n,k) cyclic code is unique (usually proved by contradiction),
2. Any multiple of generator polynomial is a codeword polynomial, and
3. The generator polynomial and parity-check polynomial are factors of x^n-1.

The generator polynomial $g(x)$ and the parity-check polynomial $h(x)$ serve the same role as the generator matrix $\boldsymbol{G}$ and parity check matrix $\boldsymbol{H}$ of a linear block code. n-tuples related to the k polynomials $g(x)$, $xg(x)$,…,$x^{k-1}g(x)$ may be used in rows of the kxn generator matrix $\boldsymbol{G}$, while n-tuples related to the (n-k) polynomials $x^kh(x^{-1})$, $x^{k+1}h(x^{-1})$,…,$x^{n-1}h(x^{-1})$ may be used in rows of the (n-k)xn parity-check matrix $\boldsymbol{H}$.

To encode we have simply to multiply the message polynomial $m(x)=m_0+m_1x+\ldots+m_{k-1}x^{k-1}$ with the generator polynomial $g(x)$, i.e. $c(x)=m(x)g(x)\text{mod}(x^n-1)$, where $c(x)$ is the codeword polynomial. To encode in *systematic* form we have to find the reminder of $x^{n-k}m(x)/g(x)$ and add

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

it to the shifted version of message polynomial $x^{n-k}m(x)$, i.e. $c(x)= x^{n-k}m(x)+\text{rem}[x^{n-k}m(x)/g(x)]$, where with rem[ ] is denoted the remainder of a given entity. The general circuit for generating the codeword polynomial in systematic form is given in **Figure 13.6**. The encoder operates as follows. When the switch S is in position 1 and Gate is closed (on), the information bits are shifted into the shift register and at the same time transmitted onto the channel. Once all information bits are shifted into register in k shifts, with Gate being open (off), the switch S is moved in position 2, and the content of (n-k)-shift register is transmitted onto the channel.

To check if the received word polynomial is the codeword polynomial $r(x)=r_0+r_1x+\ldots+r_{n-1}x^{n-1}$ we have simply to determine the *syndrome polynomial* $s(x)=\text{rem}[r(x)/g(x)]$. If $s(x)$ is zero then there is no error introduced during transmission. Corresponding circuit is given in **Figure 13.7**.

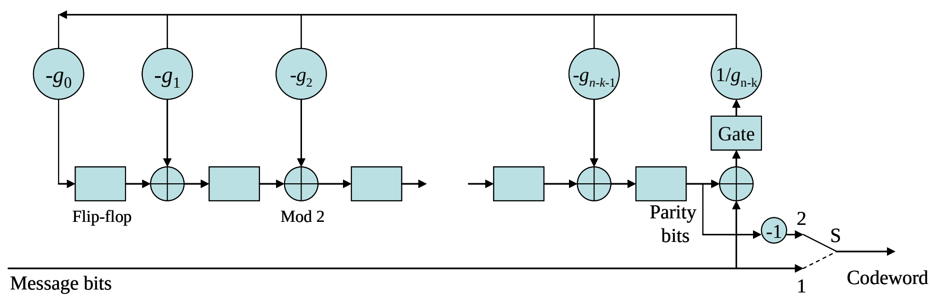

Figure 13.6 *Systematic cyclic encoder*

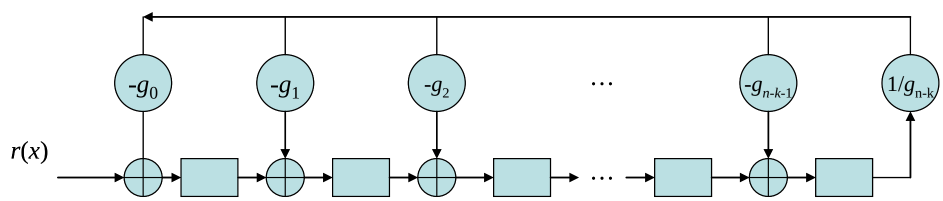

Figure 13.7 *Syndrome calculator*

13.4 Reed-Solomon (RS) Codes, Concatenated Codes, And Product Codes

The Reed-Solomon codes were discovered in 1960, and represent a special class of nonbinary BCH codes [37],[38]. RS codes represent the most commonly used nonbinary codes. Both the code symbols and the roots of generating polynomial are from the locator field. In other words, the symbol field and locator field are the same (m=1) for RS codes. The codeword length of RS codes is determined by $n=q^m-1=q-1$, so that RS codes are relatively short codes. The minimum polynomial for some element β is $P_\beta(x)=x-\beta$. If α is the primitive element of GF(q) (q is a prime or prime power), the generator polynomial for t-error correcting Reed-Solomon code is given by [21],[42]-[25]

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

$$g(x)=(x-\alpha)\left(x-\alpha^2\right)...\left(x-\alpha^{2t}\right)$$
(Equation 13.45)

The generator polynomial degree is $2t$ and it is the same as the number of parity symbols n-k=$2t$, while the block length of the code is n=q-1. Since the minimum distance of BCH codes is $2t$+1, the minimum distance of RS codes is $d_{\min}$=n-k+1, satisfying therefore the Singleton bound ($d_{\min}\leq n$-k+1) with equality and belonging to the class of *maximum-distance separable* (MDS) *codes*. When q=2^m, the RS codes parameters are: n=$m(2^m$-1), n-k=$2mt$, and $d_{\min}$=$2mt$+1. Therefore, the minimum distance of RS codes, when observed as binary codes, is large. The RS codes may be considered as burst error correcting codes, and as such are suitable for high speed optical transmission at 40 Gb/s or higher, since the fiber-optics channel at 40 Gb/s is burst-errors prone due to intrachannel nonlinearities, especially intrachannel four-wave mixing and nonlinear phase noise. This binary code is able to correct up to t bursts of length m. Equivalently, this binary code is able to correct a single burst of length $(t$-1$)m$+1.

Example. Let the GF(4) be generated by $1+x+x^2$ as we explained in section 4.4.1. The symbols of GF(4) are 0, 1, α and α^2. The generator polynomial for RS(3,2) code is given by $g(x)$=x-α. The corresponding codewords are: 000, 101, $\alpha 0\alpha$, $\alpha^2 0\alpha^2$, 011, 110, $\alpha 1\alpha^2$, $\alpha^2 1\alpha$, $0\alpha\alpha$, $1\alpha\alpha^2$, $\alpha\alpha 0$, $\alpha^2\alpha 1$, $0\alpha^2\alpha^2$, $1\alpha^2\alpha$, $\alpha\alpha^2 1$, and $\alpha^2\alpha^2 0$. This code is essentially the even parity check code ($\alpha^2+\alpha+1=0$). The generator polynomial for RS(3,1) is $g(x)$=$(x$-$\alpha)(x$-$\alpha^2)$=x^2+x+1, while the corresponding codewords are: 000, 111, $\alpha\alpha\alpha$, and $\alpha^2\alpha^2\alpha^2$. Therefore, this code is in fact the repetition code.

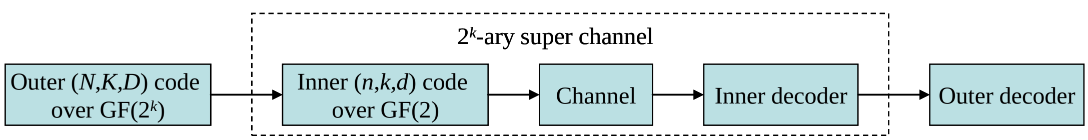

Figure 13.8 *The concatenated (Nn,Kk,≥Dd) code*

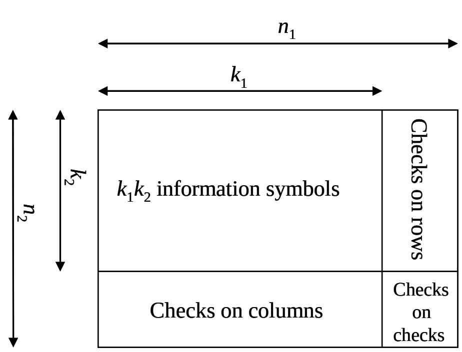

Figure 13.9 *The structure of a codeword of a turbo product code.*

To improve the bust error correction capability of RS codes, RS code can be combined with an inner binary block code in a *concatenation* scheme as shown in **Figure 13.8**. The key idea

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

behind the concatenation scheme can be explained as follows [23]. Consider the codeword generated by inner (n,k,d) code (with d being the minimum distance of the code), and transmitted over the bursty channel. The decoder processes the erroneously received codeword and decodes it correctly. However, occasionally the received codeword is decoded incorrectly. Therefore, the inner encoder, the channel and the inner decoder may be considered as a super channel whose input and output alphabets belong to GF(2^k). The outer encoder (N,K,D) (D-the minimum distance of outer code) encodes input K symbols and generates N output symbols transmitted over the super channel. The length of each symbol is k information digits. The resulting scheme, known as concatenated code and proposed initially by Forney [45], is an $(Nn,Kk,\geq Dd)$ code with the minimum distance of at least Dd. For example, RS(255,239,8) code can be combined with the (12,8,3) single parity check code in the concatenation scheme $(12\cdot 255,239\cdot 8,\geq 24)$. The outer RS decoder can be implemented using the Massey-Berlekamp algorithm [38], while the inner decoder can be implemented using the maximum a posteriori decoding based on BCJR algorithm [27]. The concatenated scheme from **Figure 13.8** can be generalized to q-ary channels, the inner code operating over GF(q), and outer over GF(q^k).

Two RS codes can be combined in a concatenated scheme by interleaving. An *interleaved code* is obtained by taking L codewords (of length N) of a given code $\boldsymbol{x}_j=(x_{j1},x_{j2},\ldots,x_{jN})$ $(j=1,2,\ldots,L)$, and forming the new codeword by interleaving the L codewords as follows $\boldsymbol{y}_i=(x_{11},x_{21},..,x_{L1},x_{12},x_{22},..,x_{L2},\ldots,x_{1N},x_{2N},..,x_{LN})$. The process of interleaving can be visualized as the process of forming an LxN matrix of L codewords written row by row and transmitting the matrix column by column, as given below

$$\begin{matrix} x_{11}\ x_{12}\ \ldots\ x_{1N} \\ x_{21}\ x_{22}\ \ldots\ x_{2N} \\ \ldots\ldots\ldots\ldots\ldots \\ x_{L1}\ x_{L2}\ \ldots\ x_{LN} \end{matrix}$$

(Equation 13.46)

The parameter L is known as the interleaving *degree*. The transmission must be postponed until L codewords are collected. To be able to transmit a column whenever a new codeword becomes available, the codewords should be arranged down diagonals as given below, and the interleaving scheme is known as the *delayed interleaving* (1-frame delayed interleaving):

$$\begin{matrix} x_{i-(N-1),1} & \ldots & x_{i-2,1} & x_{i-1,1} & x_{i,1} & & \\ & x_{i-(N-1),2} & \ldots & x_{i-2,2} & x_{i-1,2} & x_{i,2} & \\ & & \ldots\ldots & \ldots\ldots & \ldots\ldots & & \\ & & & x_{i-(N-1),N-1} & x_{i-(N-2),N-1} & & \\ & & & & x_{i-(N-1),N} & x_{i-(N-2),N} & \end{matrix}$$

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

(Equation 13.47)

Each new codeword completes one column of this array. In the example above the codeword $\boldsymbol{x}_i$ completes the column (frame) $x_{i,1}, x_{i-1,2}, \ldots, x_{i-(N-1),N}$. A generalization of this scheme, in which the components of i-th codeword $\boldsymbol{x}_i$ say $x_{i,j}$ and $x_{i,j+1}$ are spaced λ frames apart, is known as λ-frame delayed interleaved.

Another way to deal with burst errors is to arrange two RS codes in *turbo product* manner as shown in **Figure 13.9**. A product code [3-6] is an (n_1n_2,k_1k_2,d_1d_2) code in which codewords form an n_1xn_2 array such that each row is a codeword from an (n_1,k_1,d_1) code C_1, and each column is a codeword from an (n_2,k_2,d_2) code C_2; with n_i, k_i and d_i (i=1,2) being the codeword length, dimension and minimum distance, respectively, of i^{th} component code. Turbo product codes were proposed by Elias [28]. Both binary (such as binary BCH codes) and nonbinary codes (such as RS codes) may be arranged in the turbo product manner. It is possible to show [42] that the minimum distance of a product codes is the product of minimum distances of component codes. It is straightforwardly to show that the product code is able to correct the burst error of length b=max(n_1b_2,n_2b_1), where b_i is the burst error capability of component code i=1,2.

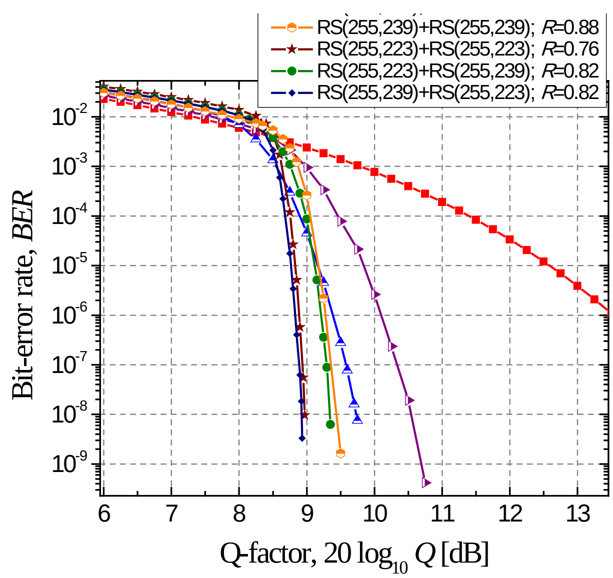

***Figure 13.10** BER performance of concatenated RS codes.*

The results of Monte Carlo simulations for different RS concatenation schemes and an AWGN channel are shown in **Figure 13.10**. Interestingly, the concatenation scheme RS(255,239)+RS(255,223) of code rate R=0.82 outperforms the concatenation scheme RS(255,223)+RS(255,223) of lower code rate R=0.76, as well as the concatenation scheme RS(255,223)+RS(255,239) of the same code rate.

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

13.5 Concluding Remarks

In this Chapter, which was based on our two recent books [47],[48], we described different FEC schemes suitable for use in optical channels. Those schemes were classified into two categories, standard block codes and iteratively decodable codes. Two important classes of standard block codes are cyclic codes and BCH codes. The cyclic codes are suitable for use in error detection, while the BCH codes for error correction. The important sub-class of BCH codes, RS codes, is standardized for use in optical communications [1]. To deal with simultaneous random errors and burst errors we described the concatenated codes, and interleaved codes.

The standard FEC schemes that belong to the class of hard-decision codes were described in this chapter. More powerful FEC schemes belong to the class of soft iteratively decodable codes, but their description is out of scope of this chapter. In our recent books [47],[48], we described several classes of iteratively decodable codes, such as turbo codes, turbo-product codes, LDPC codes, GLDPC codes, and nonbinary LDPC codes. We also discuss an FPGA implementation of decoders for binary LDPC codes. We then explain how to combine multilevel modulation and channel coding optimally by using coded modulation. Also, we describe an LDPC-coded turbo-equalizer as a candidate for dealing simultaneously with fiber nonlinearities, PMD, and residual chromatic dispersion.

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

References

[1] ITU, Telecommunication Standardization Sector, "Forward error correction for submarine systems," Tech. Recommendation G.975/ G709.

[2] "Forward error correction for high bit rate DWDM submarine systems," Telecommunication Standardization Sector, International Telecommunication Union, G. 975.1, Feb. 2004.

[3] T. Mizuochi *et al.*, "Forward error correction based on block turbo code with 3-bit soft decision for 10 Gb/s optical communication systems," *IEEE J. Selected Topics Quantum Electronics*, vol. 10, no. 2, pp. 376-386, Mar./Apr. 2004.

[4] T. Mizuochi *et al*., "Next generation FEC for optical transmission systems," in Proc. *Opt. Fib. Comm. Conf. (OFC 2003),* vol. 2, pp. 527-528, 2003.

[5] O. A. Sab, "FEC techniques in submarine transmission systems," in Proc. *Optical Fiber Communication Conference*, vol. 2, pp. TuF1-1-TuF1-3, 2001.

[6] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. 1993 Int. Conf. Comm.* (*ICC 1993*), pp. 1064-1070.

[7] C. Berrou, A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Trans. Comm.*, pp. 1261-1271, Oct. 1996.

[8] R. M. Pyndiah, "Near optimum decoding of product codes," *IEEE Trans. Comm.*, vol. 46, pp. 1003–1010, 1998.

[9] O. A. Sab, V. Lemarie, "Block turbo code performances for long-haul DWDM optical transmission systems," *OFC,* vol. 3, pp. 280 –282, 2001.

[10] T. Mizuochi, "Recent progress in forward error correction and its interplay with transmission impairments," *IEEE J. Sel. Top. Quantum Electron.*, vol. 12, no. 4, pp. 544 - 554, Jul./Aug. 2006.

[11] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[12] I. B. Djordjevic, S. Sankaranarayanan , S. K. Chilappagari, and B. Vasic, "Low-density parity-check codes for 40 Gb/s optical transmission systems," *IEEE/LEOS J. Sel. Top. Quantum Electron.*, vol. 12, no. 4, pp. 555-562, July/Aug. 2006.

[13] I. B. Djordjevic, O. Milenkovic, B. Vasic, "Generalized low-density parity-check codes for optical communication systems," *IEEE/OSA J. Lightwave Technol.*, vol. 23, pp. 1939-1946, May 2005.

[14] B. Vasic, I. B. Djordjevic, R. Kostuk, "Low-density parity check codes and iterative decoding for long haul optical communication systems," *IEEE/OSA Journal of Lightwave Technology*, vol. 21, pp. 438-446, Feb. 2003.

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

[15] I. B. Djordjevic *et al*., "Projective plane iteratively decodable block codes for WDM high-speed long-haul transmission systems", *IEEE/OSA J. Lightwave Technol.*, vol. 22, pp. 695-702, Mar. 2004.

[16] O. Milenkovic, I. B. Djordjevic, B. Vasic, "Block-circulant low-density parity-check codes for optical communication systems," *IEEE/LEOS Journal of Selected Topics in Quantum Electronics*, vol. 10, pp. 294-299, Mar./Apr. 2004.

[17] B. Vasic, I. B. Djordjevic, "Low-density parity check codes for long haul optical communications systems," *IEEE Photonics Technology Letters,* vol. 14, pp. 1208-1210, Aug. 2002.

[18] I. B. Djordjevic, M. Arabaci, and L. Minkov, "Next generation FEC for high-capacity communication in optical transport networks," *IEEE/OSA J. Lightw. Technol.*, accepted for publication. (Invited Paper.)

[19] S. Chung *et al.*, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon Limit," *IEEE Comm. Lett.*, vol. 5, pp. 58–60, Feb. 2001.

[20] I. B. Djordjevic, L. L. Minkov, and H. G. Batshon, "Mitigation of linear and nonlinear impairments in high-speed optical networks by using LDPC-coded turbo equalization," *IEEE J. Sel. Areas Comm., Optical Comm. and Netw.*, vol. 26, no. 6, pp. 73-83, Aug. 2008.

[21] S. Lin, D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Inc., USA, 1983.

[22] P. Elias, "Error-free coding'" *IRE Trans. Inform. Theory*, vol. IT-4, pp. 29-37, Sep. 1954.

[23] J. B. Anderson, S. Mohan, *Source and Channel Coding: An Algorithmic Approach*. Boston, MA: Kluwer Academic Publishers, 1991.

[24] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, the Netherlands: North Holland, 1977.

[25] S. B. Wicker, *Error control systems for digital communication and storage*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1995.

[26] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[27] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory,* vol. IT-20, no. 2, pp. 284-287, Mar. 1974.

[28] D. B. Drajic, *An Introduction to Information Theory and Coding* (2nd Edition). Belgrade, Serbia: Akademska Misao, 2004. (in Serbian.)

[29] W. E. Ryan, "Concatenated convolutional codes and iterative decoding," in *Wiley Encyclopedia in Telecommunications* (J. G. Proakis, ed.), John Wiley and Sons, 2003.

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

[30] S. Haykin, *Communication Systems*. John Wiley & Sons, Inc., 2004.

[31] J. G. Proakis, *Digital Communications*. Boston, MA: McGaw-Hill, 2001.

[32] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. Boston, MA: John Wiley & Sons, 2002.

[33] I. Anderson, *Combinatorial Designs and Tournaments*. Oxford University Press, 1997.

[34] D. Raghavarao, *Constructions and Combinatorial Problems in Design of Experiments*. Dover Publications, Inc., New York 1988. (reprint)

[35] T. M. Cover, J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., New York 1991.

[36] F. M. Ingels, *Information and Coding Theory*. Intext Educational Publishers, Scranton 1971.

[37] I. S. Reed, G. Solomon, “Polynomial codes over certain finite fields,” *SIAM J. Appl. Math.*, vol. 8, pp. 300-304, 1960.

[38] S. B. Wicker, V. K. Bhargva (Eds.), *Reed-Solomon Codes and Their Applications*. IEEE Press, New York 1994.

[39] J. K. Wolf, “Efficient maximum likelihood decoding of linear block codes using a trellis,” *IEEE Trans. Inform. Theory,* vol. IT-24, no. 1, pp. 76-80, Jan. 1978.

[40] B. Vucetic, J. Yuan, *Turbo Codes-Principles and Applications*. Kluwer Academic Publishers, Boston 2000.

[41] M. Ivkovic, I. B. Djordjevic, and B. Vasic, “Calculation of achievable information rates of long-haul optical transmission systems using instanton approach,” *IEEE/OSA J. Lightwave Technol.*, vol. 25, pp. 1163-1168, May 2007.

[42] G. Bosco, and P. Poggiolini, “Long-distance effectiveness of MLSE IMDD receivers,” *IEEE Photon. Tehcnol. Lett.*, vol. 18, no. 9, pp. 1037-1039, May 1, 2006.

[43] M. Ivkovic, I. Djordjevic, P. Rajkovic, and B. Vasic, “Pulse energy probability density functions for long-haul optical fiber transmission systems by using instantons and edgeworth expansion,” *IEEE Photon. Technol. Lett.*, vol. 19, no. 20, pp. 1604 - 1606, Oct.15, 2007.

[44] D. Divsalar and F. Pollara, “Turbo Codes for deep-space communications,” TDA Progress Report 42-120, pp. 29-39, February 15, 1995.

[45] M. E. van Valkenburg, *Network Analysis*. 3rd Edition, Prentice-Hall, Englewood Cliffs, 1974.

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.

[46] I. B. Djordjevic, M. Arabaci, and L. Minkov, “Next generation FEC for high-capacity communication in optical transport networks,” *IEEE/OSA J. Lightw. Technol.*, vol. 27, no. 16, pp. 3518-3530, August 15, 2009. (Invited Paper.)

[47] W. Shieh and I. Djordjevic, *OFDM for Optical Communications*. Elsevier, Oct. 2009.

[48] I. B. Djordjevic, William Ryan, and Bane Vasic, *Coding for Optical Channels*. Springer, March 2010.

The authors would like to acknowledge support from the National Science Foundation through CIAN NSF ERC under grant #EEC-0812072.