7-1988

DC-free coset codes

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

M. A. Herro

DOI: https://doi.org/10.1109/18.9775

DC-Free Coset Codes

ROBERT H. DENG AND MARK A. HERRO, MEMBER, IEEE

***Abstract*—A class of block coset codes with disparity and run-length constraints are studied. They are particularly well suited for high-speed optical fiber links and similar channels, where dc-free pulse formats, channel error control, and low-complexity encoder–decoder implementations are required. The codes are derived by partitioning linear block codes. The encoder and decoder structures are the same as those of linear block codes with only slight modifications. A special class of dc-free coset codes are derived from BCH codes with specified bounds on minimum distance, disparity, and run length. The codes we derive have low disparity levels (a small running digital sum) and good error-correcting capabilities.**

I. Introduction

Recently, work on constructing line codes (or transmission codes) for systems such as fiber optic links and magnetic or optical recording has received much attention in the literature [1]–[11]. These codes are designed to impose some type of spectral constraint on the transmitted sequence, e.g., have a null at dc (for fiber optic links) or to limit the high-frequency components (magnetic recording). In some applications it is also desirable that these codes provide some error-correcting or error-detecting capability as well. The codes we construct in this paper simultaneously meet the dc constraints and the error-correcting requirements and, because of the simplicity of the encoder and decoder design, can be used for high-speed digital communications.

Linear block codes can be designed to have powerful error-correcting and error-detecting capabilities and can be encoded and decoded efficiently due to their elegant algebraic structures [12]. However, they usually do not possess desirable dc properties. Line codes, on the other hand, are designed to have a zero dc component and limited run length to aid in the receiver synchronization and detection processes but typically offer little or no error-control capabilities. The dc-free attribute can be achieved by strongly bounding the running disparity of the transmitted sequences [7]. The *disparity* of a codeword is the difference between the number of ones and the number of zeros. We let D denote the maximum *running* (cumulative) disparity after any bit position, while D' denotes the running disparity at the end of a codeword. The *run length* is defined as the number of consecutive 1's or 0's in a sequence of coded bits. It is often given in the form (l, L), where l is equal to one less than the shortest run length, and L is equal to one less than the longest run length. Codes designed for digital transmission, such as the ones presented in this paper, have $l = 0$. The preferred codes for magnetic recording, on the other hand, usually have $l \geq 1$, i.e., the minimum spacing between transitions is longer than a symbol interval.

We study a class of dc-free coset codes for use on high-speed optical links and similar channels, where dc-free binary symbol formats, channel error control, and high-speed (or low-complexity) encoding/decoding are demanded. The dc-free coset code is denoted by (n, k, D), where n is the codeword length and k is the information block length. In Section II we first present a class of dc-free coset codes that have particularly simple encoding and decoding algorithms. This section also serves as an introduction to the error-correcting dc-free coset codes studied in Section III. The dc-free coset code considered in Section II is defined by $\boldsymbol{v} = (0, \boldsymbol{u}) + a\mathbf{1}_n$, where $\boldsymbol{v}$ is an n-bit codeword, $\boldsymbol{u}$ is an $(n-1)$-bit information vector, $a \in \{0,1\}$, and $\mathbf{1}_n$ is the n-bit all-one vector. The coset code consists of the linear code

$$T_1 = \{\boldsymbol{v}_i | \boldsymbol{v}_i = (0, \boldsymbol{u})\}, \qquad a = 0$$

and its coset,

$$T_2 = \{\boldsymbol{v}_i' | \boldsymbol{v}_i' = \boldsymbol{v}_i + \mathbf{1}_n\}, \qquad a = 1$$

for $i = 0, 1, \cdots, 2^{n-1} - 1$. (In this paper, the "+" and "Σ" operators indicate modulo-2 addition when applied to binary vectors.) The construction of the dc-free coset code is based on the idea of "vector space partitioning." The linear space of 2^n vectors is partitioned into 2^{n-1} disjoint subsets $\{A_0, A_1, \cdots, A_1, \cdots, A_{2^{n-1}-1}\}$, where $A_i = \{\boldsymbol{v}_i, \boldsymbol{v}_i'\}$, for $i = 0, 1, \cdots, 2^{n-1} - 1$.

In Fig. 1 we show the general encoding scheme for the dc-free coset code. The parameter D_t' denotes the running disparity at the end of a codeword at time t (one time unit corresponding to the transmission period of a codeword). The encoding can be described as follows. Suppose an $(n-1)$-bit information vector $\boldsymbol{u}$ (corresponding to message $i = 0, 1, 2, \cdots, 2^{n-1} - 1$) is to be encoded at time t. The $n-1$ bits are fed into the subset selector to select the subset A_i. Then D_{t-1}' is used to select one of the codewords in A_i, i.e., either $\boldsymbol{v}_i$ or $\boldsymbol{v}_i'$, such that the output codeword sequence has the desired running disparity.

Manuscript received November 24, 1986; revised September 24, 1987. This work was supported in part by NSF Grant ECS 84-14608 and NASA Grant NCA2-1R562-401. This paper was presented in part at the 1987 Conference on Information Sciences and Systems, Baltimore, MD, March 1987.

R. H. Deng was with the Department of Electrical and Computer Engineering, University of Notre Dame, Notre Dame, IN. He is now with the Institute of Systems Science, National University of Singapore, Heng Mui Keng Terrace, Kent Ridge, Singapore 0511.

M. A. Herro is with the Department of Electrical and Computer Engineering, University of Notre Dame, Notre Dame, IN 46556.

IEEE Log Number 8822483.

0018-9448/88/0700-0786$01.00

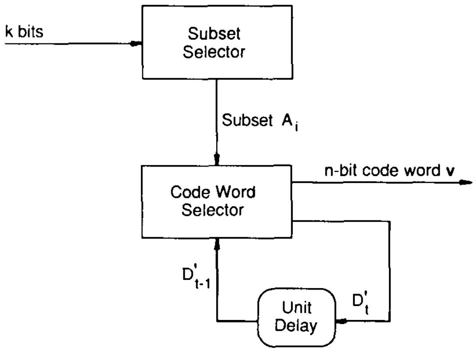

Fig. 1. General encoding scheme for dc-free coset code.

In Section III the idea of "vector space partitioning" is extended to "code vector space partitioning" to construct dc-free coset codes with error-correcting capability. The construction of these codes is motivated by the recent work of Herro and Hu [4]. The 2^{k+J} codewords generated by an $(n, k+J)$ linear block code are partitioned into 2^k disjoint subsets $\{B_0, B_1, \cdots, B_{2^k-1}\}$, and each k-bit information vector is associated with exactly one subset. The encoding can also be described by Fig. 1 if A_i is replaced by B_i. Because only codewords of the $(n, k+J)$ linear block code are transmitted, the error-correcting dc-free coset code has the same error-correcting capability as the linear block code, and the encoder/decoder structures of the linear block code can be used to encode and decode the error-correcting dc-free coset code with only slight modifications.

In Section IV we present a systematic method for constructing error-correcting dc-free coset codes from BCH codes. Codes of practical interest are also given and listed. Many of the codes we found have low disparity levels (a small running digital sum) and good error-correcting capabilities. Finally, in Section V we draw some conclusions from our results.

II. DC-Free Coset Codes—Without Error-Correcting Capability

The (n, k, D) dc-free coset code is defined by

$$\begin{aligned} \boldsymbol{v} &= (0, \boldsymbol{u}) + a\mathbf{1}_n \\ &= (a, \boldsymbol{u} + a\mathbf{1}_{n-1}), \qquad a = 0,1 \end{aligned} \tag{1}$$

where $\boldsymbol{v}$ is an n-bit codeword corresponding to a $k = (n-1)$-bit information vector $\boldsymbol{u}$, and $\mathbf{1}_n$ is the n-bit all-one vector. Equation (1) implies that an $(n-1)$-bit information vector can be encoded into either $(0, \boldsymbol{u})$ or $(1, \boldsymbol{u}+\mathbf{1}_{n-1})$, both of which have the same absolute disparity but opposite polarity. The encoding rule requires that the disparity polarity of consecutive codewords with nonzero disparity alternates.

Encoding: Suppose an $(n-1)$-bit information vector $\boldsymbol{u}$ is to be encoded at time t. Let D'_{t-1} be the running disparity at the end of a codeword at time $t-1$. We proceed as follows:

1) $\boldsymbol{v} \leftarrow (0, \boldsymbol{u})$, $D'_t \leftarrow D'_{t-1}$; let d_t be the disparity of $\boldsymbol{v}$ at time t;
2) if $D'_t \cdot d_t \le 0$, $D'_t \leftarrow D'_t + d_t$, and go to 4; else go to 3;
3) $\boldsymbol{v} \leftarrow \boldsymbol{v} + \mathbf{1}_n$, $D'_t \leftarrow D'_t - d_t$, go to 4;
4) encode the next $n-1$ information bits.

Decoding: Let $\hat{\boldsymbol{v}} = (\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_n)$ be the received version of $\boldsymbol{v}$:

1) if $\hat{v}_1 = 0$, $\hat{\boldsymbol{u}} = (\hat{v}_2, \hat{v}_3, \cdots, \hat{v}_n)$, and go to 2; else, $\hat{\boldsymbol{u}} = (\hat{v}_2, \hat{v}_3, \cdots, \hat{v}_n) + \mathbf{1}_{n-1} = (\hat{v}'_2, \hat{v}'_3, \cdots, \hat{v}'_n)$, and go to 2;
2) decode the next n-bit received word.

From the encoding rule, we see that the running disparity at the end of any codeword is bounded by

$$|D'| \le n. \tag{2}$$

(We have dropped the subscript t from D' since these bounds hold for all codewords.) The maximum running disparity at any given bit position is given by

$$|D| = n + \left\lfloor \frac{n}{2} \right\rfloor \tag{3}$$

and the worst case occurs when the disparity at the end of a codeword is 0, followed by an all-one codeword, then followed by a codeword with $\lfloor n/2 \rfloor$ 1's in its first $\lfloor n/2 \rfloor$ positions. The maximum run length equals $2n + \lfloor n/2 \rfloor$, so that

$$L = \text{maximum run length} - 1 = 2n + \left\lfloor \frac{n}{2} \right\rfloor - 1 \tag{4}$$

and the worst case occurs when the disparity at the end of a codeword is n, followed by two all-zero codewords, then followed by a codeword with $\lfloor n/2 \rfloor$ 0's in its first $\lfloor n/2 \rfloor$ positions. Table I lists a set of rate $R = (n-1)/n$ dc-free coset codes. Also shown in Table I is the capacity $C(D)$ for the disparity constrained channel given by Chien [13].

TABLE I
Rate $R = (n-1)/n$ DC-Free Coset Codes

n	D	R	$C(D)$
4	6	0.75	0.963
8	12	0.875	0.989
16	24	0.938	0.997
32	48	0.969	0.999

III. DC-Free Coset Codes—With Error-Correcting Capability

In this section we will extend the idea of the previous section to construct dc-free coset codes with error-correcting capabilities. As we will see in the next section, error-correcting dc-free coset codes can be derived from BCH codes and will have almost the same dc properties as the codes constructed in the last section, with only a slight decrease in code rate. More importantly, these codes require only simple encoding/decoding operations, which is in contrast to the block line codes found in the existing literature. It is this feature that renders them very attractive for applications in high-speed transmission links.

A. Code Structure

An $(n, k+J)$ linear block code C_b is specified by its $(k+J)\times n$ generator matrix G_1. If the code has a minimum distance d_b, then it can correct t or fewer errors and simultaneously detect λ or fewer errors provided that [12]

$$2t+\lambda+1\leq d_b. \tag{5}$$

Although linear block codes can be designed for powerful error protection, they usually do not have good dc properties.

The (n, k, D) error-correcting dc-free coset code, denoted by C_e, is defined by

$$v = u[G_2G_1]+g \tag{6}$$

where u is a k-bit information vector, and G_2 is a $k\times(k+J)$ matrix, called the *transfer matrix.* The matrix G_2 transfers the k information bits into the last k bits of the $(k+J)$-bit vector uG_2. That is, uG_2 has the form $(0,0,\cdots,0,u_1,u_2,\cdots,u_k)$. The n-bit codeword

$$g=\sum_{j=1}^{J} a_j g_j, \qquad a_j=0,1 \tag{7}$$

is a linear combination of the first J rows of the generator matrix G_1 of C_b. The codeword g plays a similar role to the vector $\mathbf{1}_n$ in (1) in Section II, i.e., it is used to control the codeword disparity. (In (1), $J=1$ and $g=a_1\mathbf{1}_n$.) Since C_e is a subset of C_b, the minimum distance of C_e, denoted by d_e, is at least as large as d_b, i.e., $d_e\geq d_b$. Therefore, C_e is at least as powerful as C_b in error-correcting and error-detecting capability.

To be effective in controlling codeword disparity, the first J rows of G_1 are chosen to satisfy

$$\operatorname{supp}(g_i)\cap\operatorname{supp}(g_j)=\phi, \qquad \text{for } i\neq j \tag{8}$$

and

$$g_1+g_2+\cdots+g_J=\mathbf{1}_n \tag{9}$$

where the support of g_j, $\operatorname{supp}(g_j)$, is the set of coordinates at which the components of g_j are nonzero. Equation (8) implies that g_j only controls the disparity of v at coordinates $\operatorname{supp}(g_j)$, while (9) guarantees that the disparity at all the coordinates of v can be controlled. Let w_j denote the Hamming weight of g_j. Without loss of generality, we assume that[1]

$$\begin{aligned} g_1&=(\mathbf{1}_{w_1},\mathbf{0}_{w_2},\mathbf{0}_{w_3},\cdots,\mathbf{0}_{w_J})\\ g_2&=(\mathbf{0}_{w_1},\mathbf{1}_{w_2},\mathbf{0}_{w_3},\cdots,\mathbf{0}_{w_J})\\ &\ \vdots\\ g_J&=(\mathbf{0}_{w_1},\mathbf{0}_{w_2},\cdots,\mathbf{0}_{w_{J-1}},\mathbf{1}_{w_J}). \end{aligned} \tag{10}$$

[1]If g_j, $j=1,2,\cdots,J$ satisfy (8) and (9) but are not in the form of (10), the J vectors can be transformed into the form of (10) by reordering the coordinates of the original codewords since reordering produces an equivalent code.

That is, the $\{g_j\}$ in (10) consist of $J-1$ all-zero segments and one all-one segment. Obviously, the $\{g_j\}$ in (10) satisfy (8) and (9). Divide the n-bit codeword v in (6) into J segments, i.e., $v=(v_1,v_2,\cdots,v_J)$, where v_j is a w_j-bit vector. Then g_j only controls the disparity of the jth segment v_j.

The $k\times(k+J)$ matrix G_2 is given by

$$G_2=\begin{bmatrix} 0 & \cdots & 0 & \\ 0 & \cdots & 0 & \\ \vdots & & \vdots & I_k \\ 0 & \cdots & 0 & \end{bmatrix} \tag{11}$$

where I_k is the $k\times k$ identity matrix. From (6), (7), and (11) we obtain

$$\begin{aligned} v&=uG_2G_1+g=(\mathbf{0}_J,u)G_1+\sum_{j=1}^{J}a_jg_j\\ &=(a_1,a_2,\cdots a_J,u)+G_1. \end{aligned} \tag{12}$$

Note that $G_2G_2^T=I_k$. From (12) the k-bit information vector u can be recovered from

$$u=(a_1,a_2,\cdots,a_J,u)G_2^T. \tag{13}$$

Example 1: Let $n=7$, $k=2$, and $J=2$. Let the $d_b=3$ (7,4) Hamming code be generated by

$$G_1=\begin{bmatrix} 1&1&1&0&0&0&0\\ 0&0&0&1&1&1&1\\ 1&1&0&1&0&1&0\\ 1&0&0&1&0&0&1 \end{bmatrix}.$$

Note that the first two rows of G_1, $g_1=(1\ 1\ 1\ 0\ 0\ 0\ 0)$ and $g_2=(0\ 0\ 0\ 1\ 1\ 1\ 1)$ satisfy (10). The 2×4 matrix G_2, from (11) is given by

$$G_2=\begin{bmatrix} 0&0&1&0\\ 0&0&0&1 \end{bmatrix}.$$

The 2-bit information vector $u=(u_1,u_2)$ is then encoded into

$$v=uG_2G_1+\sum_{j=1}^{2}a_jg_j=(a_1,a_2,u_1,u_2)G_1.$$

At the decoder, suppose (a_1,a_2,u_1,u_2) is recovered correctly based on the (7,4) block code decoder, then

$$\hat{u}=(a_1,a_2,u_1,u_2)G_2^T=(u_1,u_2).$$

B. Encoding and Decoding

Let D_t' be the disparity at the end of a codeword at time t. Let d_j denote the disparity of the jth segment v_j of v. If g_j is added to v, the polarity of d_j in the new codeword will change, and the disparity d_i, $i\neq j$, $i=1,2,\cdots,J$, remains the same as in the old codeword. The encoding rule, as in Section II, also requires that the disparity polarity of consecutive nonzero disparity segments alternate.

Encoding: Suppose a k-bit information vector $\boldsymbol{u}$ is to be encoded at time t. We proceed as follows:

1) $\boldsymbol{v} \leftarrow \boldsymbol{u}\boldsymbol{G}_2\boldsymbol{G}_1$, $j \leftarrow 1$, $D_t' \leftarrow D_{t-1}'$;
2) if $D_t' \cdot d_j \le 0$, $D_t' \leftarrow D_t' + d_j$, $j \leftarrow j+1$, go to 4; else go to 3;
3) $\boldsymbol{v} \leftarrow \boldsymbol{v} + \boldsymbol{g}_j$, $D_t' \leftarrow D_t' - d_j$, $j \leftarrow j+1$, go to 4;
4) if $j \le J$, go to 2; else encode the next information block.

Decoding: Let $\hat{\boldsymbol{v}}$ be the received version of $\boldsymbol{v} = (a_1, a_2, \cdots, a_j, \boldsymbol{u})\boldsymbol{G}_1$:

1) find $(\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_j, \hat{\boldsymbol{u}})$ based on the block code C_b with generator matrix $\boldsymbol{G}_1$;
2) $\hat{\boldsymbol{u}} = (\hat{a}_1, \hat{a}_2 \cdots \hat{a}_j, \hat{\boldsymbol{u}})\boldsymbol{G}_2^T$;
3) decode the next received word.

Example 2: Here we illustrate the encoding/decoding rules by using the code of Example 1. Suppose $D_{t-1}' = 1$, and $\boldsymbol{u} = (0\ 0)$ is to be sent at time t. From the encoding rule, we have the following:

1) $\boldsymbol{v} = \boldsymbol{u}G_2G_1 = (0\ 0\ 0\ 0\ 0\ 0\ 0)$, $j \leftarrow 1$, $D_t' \leftarrow D_{t-1}' = 1$;
2) $d_j = d_1 = -3$, $D_t' \cdot d_1 = -3 < 0$, $D_t' \leftarrow 1 - 3 = -2$, $j \leftarrow j+1 = 2$;
3) $d_j = d_2 = -4$, $D_t' \cdot d_2 = 8 > 0$, then
4) $\boldsymbol{v} = (0\ 0\ 0\ 0\ 0\ 0\ 0) + \boldsymbol{g}_2 = (0\ 0\ 0\ 1\ 1\ 1\ 1)$, $j \leftarrow j+1 = 3$;
5) since $j = 3 > J = 2$, encode the next information block.

Therefore, $\boldsymbol{u}$ is encoded as $\boldsymbol{v} = (0\ 0\ 0\ 1\ 1\ 1\ 1)$. Note that $\boldsymbol{v} = (0\ 1\ 0\ 0)\boldsymbol{G}_1$, (i.e., $a_1 = 0$ and $a_2 = 1$).

Suppose that at most one channel error occurred during the transmission, i.e., the received word is decoded correctly. Then at the Hamming code decoder output we have $(\hat{a}_1, \hat{a}_2, \hat{u}_1, \hat{u}_2) = (0\ 1\ 0\ 0)$. The estimated information vector is

$$\hat{\boldsymbol{u}} = (0\quad 1\quad 0\quad 0)\boldsymbol{G}_2^T = (0\quad 1\quad 0\quad 0)\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = (0\quad 0),$$

which is the transmitted information vector.

C. Code Properties

In this subsection we will study the dc properties of the (n, k, D) error-correcting coset code just presented. Specifically, we give bounds on D and L. The subsection ends with some general comments on the code construction.

From the encoding rule we easily see that the disparity at the end of any segment of a codeword is bounded by

$$|D'| \le \max_{1 \le j \le J} w_j \tag{14}$$

where w_j is the Hamming weight of $\boldsymbol{g}_j$. The precise derivations of the maximum running disparity (D) and the largest run length (L) require detailed knowledge of the algebraic structures of the specified linear block code. Hence it is impossible to give a unified derivation. However, by reasoning similar to that given in Section II, D and L can be upper-bounded by

$$|D| \le w_{\max} + \left\lfloor \frac{w_{\max}}{2} \right\rfloor \tag{15}$$

and

$$L \le 2w_{\max} + \left\lfloor \frac{w_{\max}}{2} \right\rfloor - 1 \tag{16}$$

respectively, where

$$w_{\max} = \max_{1 \le j \le J} w_j.$$

The actual values of $|D|$ and L may be significantly smaller than the bounds. This is because a codeword in C_e is subject to many constraints. For example, the blocks that yield the worst case disparity and run length may not even be codewords in C_e. Even if these blocks are codewords in C_e, their weight w must satisfy $w \le (n + w_{\max})/2$.

Before finishing this section we give some further comments on the code construction. If the J vectors $\boldsymbol{g}_j$, $j = 1, 2, \cdots, J$, all have equal weight, then $w_j = n/J$ for all j. If J equal weight vectors cannot be found, then the $\{\boldsymbol{g}_j\}$ should be chosen such that their weights are as equal as possible. In any case, a larger value of J will result in smaller values of $|D|$ and L. Therefore, the determination of the maximum possible value of J is very important. This is stated in the following theorem.

Theorem 1: For an $(n, k+J)$ linear block code of minimum distance d_b, the number of vectors satisfying (8) and (9) is bounded by

$$J \le \left\lfloor \frac{n}{d_b} \right\rfloor. \tag{17}$$

Proof: Because the J vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_J$ satisfy (8) and (9), we have

$$\sum_{j=1}^{J} w_j = n.$$

However, since for all j, $d_b \le w_j$,

$$J \cdot d_b \le \sum_{j=1}^{J} w_j = n$$

or

$$J \le \frac{n}{d_b}.$$

Since J is an integer, we must have $J \le \lfloor n/d_b \rfloor$. Q.E.D.

IV. Construction of Error-Correcting DC-Free Coset Codes from BCH Codes

A. BCH Code Properties

BCH codes are a powerful class of codes which have well-defined code structures. A large selection of block lengths, code rates, alphabet sizes, and code minimum distances are possible. The most interesting codes to us are the binary codes. For any positive integer b, $(b > 3)$, and

$t(t < 2^b - 1)$, there exists a binary BCH code with the following parameters [12]:

block length: $n = 2^b - 1$;
number of parity-check bits: $n-(k+J) \le bt$;
minimum distance: $d_b \ge 2t+1$.

This code, denoted C_{BCH}, is called a t-error-correcting BCH code. Since BCH codes are a class of cyclic codes, we treat the components of a codeword $\boldsymbol{v} = (v_1, v_2, v_3, \cdots, v_n)$ as the coefficients of a polynomial over GF(2) as follows:

$$v(X) = v_1 + v_2 X + v_3 X^2 + \cdots + v_n X_{n-1}.$$

The correspondence between the codeword $\boldsymbol{v}$ and the polynomial $v(X)$ is one to one. The polynomial $v(X)$ is called the code polynomial of $\boldsymbol{v}$. Hereafter, we use the terms "codeword" and "code polynomial" interchangeably. The t-error-correcting BCH code is generated by the generator polynomial

$$\begin{aligned} g(X) &= \mathrm{LCM}\{\phi_1(X), \phi_2(X), \cdots, \phi_{2t}(X)\} \\ &= \mathrm{LCM}\{\phi_1(X), \phi_3(X), \cdots, \phi_{2t-1}(X)\} \\ &= 1 + g_1 X + g_2 X^2 + \cdots + g_{n-k-J-1} X^{n-k-J-1} \\ &\quad + X^{n-k-J} \end{aligned} \tag{18}$$

where $\phi_i(X)$ is the minimal polynomial of α^i, and α is a primitive element in GF(2^b). Therefore, the generator polynomial $g(X)$ of the t-error-correcting BCH code of length $2^b - 1$ is the lowest degree polynomial over GF(2) which has

$$\alpha, \alpha^2, \alpha^3, \cdots, \alpha^{2t} \tag{19}$$

as its roots (i.e., $g(\alpha^i) = 0$ for $1 \le i \le 2t$). The code polynomial $v(X)$ of the $(n, k+J)$ BCH code is generated from

$$\begin{aligned} v(X) &= u(X) g(X) \\ &= (u_1 + u_2 X + u_3 X^2 + \cdots + u_{k+J} X^{k+J-1}) g(X) \end{aligned} \tag{20}$$

where the coefficients in $u(X)$, $(u_1, u_2, \cdots, u_{k+J})$, are the $k+J$ information bits to be encoded. A polynomial $v(X)$ is a code polynomial if and only if it has $\alpha, \alpha^2, \alpha^3, \cdots, \alpha^{2t}$ as roots [12].

The construction of the (n, k, D) error-correcting dc-free coset code from a BCH code is based on the following theorem.

Theorem 2: Let m and J be two odd integers such that

$$m \cdot J = 2^b - 1 \tag{21a}$$

and

$$m \ge 2t + 1. \tag{21b}$$

Then

$$\begin{aligned} z_1(X) &= 1 + X^J + X^{2J} + \cdots + X^{(m-1)J} \\ z_2(X) &= X z_1(x) \\ &\vdots \\ z_J(X) &= X^{J-1} z_1(X) \end{aligned} \tag{22}$$

are J code polynomials of the t-error-correcting BCH code C_{BCH} generated by (18) and their corresponding codewords satisfy

$$\mathrm{supp}(z_i) \cap \mathrm{supp}(z_j) = \phi, \qquad i \ne j \tag{23a}$$

and

$$z_1 + z_2 + \cdots + z_J = \mathbf{1}_n. \tag{23b}$$

Proof: Because $m \cdot J = 2^b - 1 = n$, the polynomial $X^{2^b-1} + 1$ can be factored as

$$X^{2^b-1} + 1 = (1 + X^J) z_1(X)$$

where

$$z_1(X) = 1 + X^J + X^{2J} + \cdots + X^{(m-1)J}$$

Let α be a primitive element of GF(2^b). Since $(\alpha^m)^J = \alpha^{2^b-1} = 1$, the polynomial $X^J + 1$ has $\alpha^0 = 1$, α^m, $\alpha^{2m}, \cdots, \alpha^{(J-1)m}$ as all its roots. Since the $2^b - 1$ nonzero elements of GF(2^b) form all the roots of $X^{2^b-1} + 1$, $z_1(X)$ has α^i as a root if and only if i is not a multiple of m. From (21b) we have $m \ge 2t+1$, so $\alpha, \alpha^2, \alpha^3, \cdots, \alpha^{2t}$ are the roots of $z_1(X)$. From the definition of $g(X)$ given in (18) and (19), we see that $z_1(X)$ is a multiple of $g(X)$, and from (20) it is a code polynomial of the t-error-correcting BCH code, C_{BCH}, generated by $g(X)$. Clearly, $z_2(X) = X z_1(X), \cdots, z_J(X) = X^{J-1} z_1(X)$ are also code polynomials. From (22) we observe that for $i \ne j$, and $i, j = 1, 2, \cdots, J$, $z_i(X)$ and $z_j(X)$ do not have any common nonzero coefficients, and therefore their corresponding codewords $z_1, z_2, \cdots, z_J$ satisfy (23a). Moreover, since each z_i has a weight m, and $m \cdot J = 2^b - 1$, their modulo two sum must have a weight $n = 2^b - 1$, which proves (23b).

Example 3: Let α be a primitive element of GF(2^4). The (15,11) $d_b = 3$, $t = 1$ BCH code is generated by

$$g(X) = \mathrm{LCM}\{\phi_1(X), \phi_2(X)\} = \phi_1(X) = X^4 + X + 1$$

where $\phi_1(X)$ and $\phi_2(X)$ are minimal polynomials of α and α^2 and therefore $\phi_1(X) = \phi_2(X)$. Since $3 \cdot 5 = 15$ and $2t + 1 = 3$, m can be either 3 or 5.

1) Let $m = 3$ and $J = 5$. From (22) we have

$$\begin{aligned} z_1(X) &= 1 + X^5 + X^{10} \\ z_2(X) &= X + X^6 + X^{11} \\ z_3(X) &= X^2 + X^7 + X^{12} \\ z_4(X) &= X^3 + X^8 + X^{13} \\ z_5(X) &= X^4 + X^9 + X^{14}. \end{aligned}$$

We can easily see that $g(X)$ divides $z_1(X)$, in fact,

$$z_1(X) = (1 + X + X^2 + X^3 + X^6) g(X).$$

2) Let $m = 5$ and $J = 3$. From (22) we have

$$\begin{aligned} z_1(X) &= 1 + X^3 + X^6 + X^9 + X^{12} \\ z_2(X) &= X + X^4 + X^7 + X^{10} + X^{13} \\ z_3(X) &= X^2 + X^5 + X^8 + X^{11} + X^{14}. \end{aligned}$$

It can be seen that $z_1(X) = (1 + X + X^2 + X^4 + X^8) g(X)$.

B. Code Construction

The encoding equation for the (n, k, D) error-correcting dc-free coset code is given by (6). The code can be constructed from a BCH code by first finding the $(k+J) \times n$ generator matrix G_1 whose first J rows satisfy (10).

For an $(n, k+J)$ t-error-correcting BCH code with two odd integers m and J such that $m \cdot J = n$ and $m \geq 2t-1$, the generator matrix can take the following form:

$$G_{BCH} = \begin{bmatrix} z_1 \\ \vdots \\ z_J \\ z_{J+1} \\ \vdots \\ z_{J+k} \end{bmatrix} \tag{24}$$

where z_j, $j = 1, 2, \cdots, J$, are the code vectors corresponding to the code polynominals $z_j(X)$ in (22), and z_j, $j = J+1, J+2, \cdots, J+k$, are the code vectors corresponding to the code polynomials

$$z_j(X) = X^{j-1} g(X). \tag{25}$$

It can be seen that the $k+J$ code vectors in (24) are linearly independent. The desired generator matrix G_1 can then be obtained by performing a permutation σ on the columns of G_{BCH} such that

$$\begin{aligned} \sigma(z_1) &= (\mathbf{1}_m, \mathbf{0}_m, \mathbf{0}_m, \cdots, \mathbf{0}_m) \\ \sigma(z_2) &= (\mathbf{0}_m, \mathbf{1}_m, \mathbf{0}_m, \cdots, \mathbf{0}_m) \\ &\vdots \\ \sigma(z_J) &= (\mathbf{0}_m, \mathbf{0}_m, \mathbf{0}_m, \cdots, \mathbf{1}_m). \end{aligned} \tag{26}$$

The new $(n, k+J)$ block code generated by G_1 is equivalent to the BCH code C_{BCH}, i.e., they have the same weight distribution.

Example 4: In this example, we construct an $n = 15$, $k = 8$ error-correcting dc-free coset code from the (15,11) $t = 1$ BCH code considered in Example 3. That is, with $J = 3$ and $m = 5$, we find the 11×15 matrix G_1. The generator matrix G_{BCH}, from (24), (25), and Example 3, is given by

$$G_{BCH} = \begin{bmatrix} 1&0&0&1&0&0&1&0&0&1&0&0&1&0&0 \\ 0&1&0&0&1&0&0&1&0&0&1&0&0&1&0 \\ 0&0&1&0&0&1&0&0&1&0&0&1&0&0&1 \\ 0&0&0&1&1&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&1 \end{bmatrix}$$

By performing the permutation σ on the columns of G_{BCH}, we obtain

$$G_1 = \begin{bmatrix} 1&1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1 \\ 0&1&0&0&0&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&1&1&0&0 \\ 0&0&1&1&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&1&1&0 \\ 0&0&0&1&1&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&1&1 \end{bmatrix}$$

Since $w_{max} = m$ for these codes, from (15) and (16) we have

$$|D| \leq m + \left\lfloor \frac{m}{2} \right\rfloor \tag{27}$$

and

$$L \leq 2m + \left\lfloor \frac{m}{2} \right\rfloor - 1 \tag{28}$$

respectively. In Table II we give the code parameters for some (n, k, D) codes constructed from BCH codes of length n.

TABLE II
(n, k, D) ERROR-CORRECTING DC-FREE COSET CODES CONSTRUCTED FROM BCH CODES OF LENGTH n

n	k	R	Upper bound on D	t
15	8	0.533	7	1
15	4	0.267	7	2
15	6	0.4	4	1
63	48	0.762	10	1
63	42	0.667	10	2
63	36	0.571	10	3
63	36	0.571	4	1
255	237	0.929	22	1
255	222	0.871	22	2
255	214	0.839	22	3
255	206	0.808	22	4
255	198	0.776	22	5
255	190	0.745	22	6
255	182	0.714	22	7
255	196	0.769	7	1
255	188	0.737	7	2
255	162	0.635	4	1

V. CONCLUSION

We introduced a class of error-correcting DC-free coset codes for high-speed fiber optic communication links and similar channels. The codes were derived from partitioning linear block codes as coset codes so that high-speed encoding and decoding could be achieved.

In Section III we gave the general description of the (n, k, D) error-correcting dc-free coset codes. The key problem in code design requires the finding of J basis vectors $g_1, g_2, \cdots, g_J$ which sum to $\mathbf{1}_n$ and have disjoint supports. The solution to this problem was presented in Theorem 2 in Section IV for a large class of primitive binary BCH codes. Codes of practical interest constructed from BCH codes were listed in Table II. The (15,8,7) and the (15,6,4) codes in Table II can be compared with the (16,8,5) code given in [2]. They all have an error-correcting capability of $t=1$, and their code rates are 0.53, 0.4, and 0.5, respectively. However, the decoding of the (15,8,7) and the (15,6,4) codes are far simpler than the decoding of the (16,8,5) code. The decoders for our codes can be implemented with modified Hamming decoders, while the code in [2] requires a $64K \times 8$ bit table.

The advantage of increasing the block size is apparent from Table II. The single-error-correcting (63,36,4) and the (255,162,4) codes have lower disparity bounds than the rate 0.5 code presented in [2] but have rates 0.57 and 0.63, respectively. The double-error-correcting (255,188,7) code has the same disparity bound as the (15,8,7) rate 0.53 code but has a rate of 0.73. If even longer block lengths are acceptable, higher rates and/or better dc performance is achievable with these codes.

References

[1] A. R. Calderbank, C. Heegard, and T. A. Lee, "Binary convolutional codes with application to magnetic recording," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 797–815, Nov. 1986.

[2] H. C. Ferreira, "Lower bounds on the minimum Hamming distance achievable with runlength constrained or dc free block codes and the synthesis of a (16,8) $d_{\min}=4$ dc free block code," *IEEE Trans. Magn.*, vol. MAG-20, pp. 881–883, Sept. 1984.

[3] ____, "The synthesis of magnetic recording trellis codes with good Hamming distance properties," *IEEE Trans. Magn.*, vol. MAG-21, pp. 1356–1358, Sept. 1985.

[4] M. A. Herro and L. Hu, "Error-correcting line codes," in *Proc. 23rd Ann. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 1985, pp. 450–451.

[5] J. K. Wolf and G. Ungerboeck, "Trellis coding for partial response channels," *IEEE Trans. Commun.*, vol. COM-34, pp. 765–773, Aug. 1986.

[6] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 51–53, Jan. 1986.

[7] A. Fromageot, P. De Laage, and M. Maitre, "Transformation d'une suite binaire en une suite de somme bornée," *Câbles Trans.*, pp. 183–229, no. 26, 1972.

[8] J. M. Griffiths, "Binary codes suitable for line transmission," *Electron. Lett.*, vol. 5, pp. 79–81, 1969.

[9] K. W. Cattermole, "Codes for digital line transmission," *Electron. Lett.*, vol. 11, pp. 548–550, 1975.

[10] A. X. Widmer and P. A. Franaszek, "A dc-balanced partitioned-block, 8B/10B transmission code," *IBM J. Res. Develop.*, vol. 27, pp. 440–451, Sept. 1983.

[11] Y. Takasaki, M. Tanaka, N. Maeda, L. Yamashita, and K. Nagano, "Optical pulse formats for fiber optic digital communications," *IEEE Trans. Commun.*, vol. COM-24, pp. 404–413, Apr. 1976.

[12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[13] T. M. Chien, "Upper bound on the efficiency of dc-constrained codes," *Bell Syst. Tech. J.*, vol. 49, pp. 2267–2287, 1970.