[5] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1960.
[6] ____, "Low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.
[7] C. R. P. Hartmann and L. D. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 5, pp. 514–517, Sept. 1976.
[8] W. Meier and O. Staffelbach, "Fast correlation attacks on stream ciphers," in *Advances in Cryptology, Eurocrypt'88*, C. G. Günter, Ed. Berlin: Springer, 1988, pp. 300–314.
[9] W. W. Peterson and E. J. Weldon, *Error-Correction Codes*. Cambridge, MA: MIT Press, 1982.
[10] G. R. Redinbo, "Inequalities between the probability of a subspace and the probabilities of its cosets," *IEEE Trans. Inform. Theory*, vol. IT-19, no. 4, pp. 533–536, July 1973.
[11] D. D. Sullivan, "A fundamental inequality between the probabilities of binary subgroups and cosets," *IEEE Trans. Inform. Theory*, vol. IT-13, no. 1, pp. 91–94, Jan. 1967.

Decoding Cyclic and BCH Codes up to Actual Minimum Distance Using Nonrecurrent Syndrome Dependence Relations

Gui-Liang Feng and Kenneth K. Tzeng, *Senior Member, IEEE*

***Abstract*—The decoding capabilities of algebraic algorithms, mainly the Berlekamp–Massey algorithm, the Euclidean algorithm and our generalizations of these algorithms, are basically constrained by the minimum distance bounds of the codes. Thus, when the actual minimum distance of the codes is greater than that given by the bounds, these algorithms usually cannot fully utilize the error-correcting capability of the codes. The limitation is seen to be rooted in the original Peterson decoding procedure adhered to by these algorithms. Thus, these algorithms all require the determination of the error-locator polynomial from Newton's identities which in turn require that the syndromes be contiguous in forming a set or multiple sets of linear recurrences. A procedure is introduced that breaks away from this restriction and can determine the error locations from nonrecurrent syndrome dependence relations. This procedure employs an algorithm that has recently been introduced as a basis for the derivation of the Berlekamp–Massey algorithm and its generalization. It can decode many cyclic and BCH codes up to their actual minimum distance and is seen to be a generalization of Peterson's procedure.**

***Index Terms*—Cyclic coding, BCH coding, generalization of Peterson decoding procedure, decoding up to actual minimum distance.**

I. Introduction

Algorithms for algebraic decoding of cyclic and BCH codes, mainly the Berlekamp–Massey algorithm [1], [2], the Euclidean algorithm [3], as well as our generalizations of these algorithms [4], [5] basically suffer from a restriction imposed by the minimum distance bounds of the codes. For the Berlekamp–Massey algorithm and the Euclidean algorithm, this restriction comes from the BCH bound as these algorithms can normally decode only up to this bound. Likewise, our generalizations of these algorithms usually cannot decode beyond the Hartmann–Tzeng (HT) bound and the Roos bound [4], [6]–[8]. Several authors [9]–[15] have attempted to stretch the capability of the Berlekamp–Massey algorithm for decoding beyond the BCH bound and have succeeded in various degrees for particular cases. But, generally speaking, when the actual minimum distance of the codes is greater than that given by such bounds, these algebraic algorithms usually are not able to utilize the full error-correcting capability of the codes. The limitations are seen to be originated in the Peterson procedure for decoding BCH codes [16] adhered to by these algorithms. As such, these algorithms all require the determination of the error-locator polynomial from Newton's identities which in turn require that the syndromes be contiguous in forming a set or multiple sets of linear recurrence relations. This, of course, is a consequence of the contiguity required on the roots of the generator polynomial by these bounds.

In this correspondence, we introduce a more general procedure which breaks away from this restriction imposed by the minimum distance bounds and can determine the error locations from nonrecurrent dependence relations among the syndromes. This procedure employs an algorithm, referred to as the Fundamental Iterative Algorithm, which we have recently introduced as a basis for the derivation of the Berlekamp–Massey algorithm and its generalization [4]. The procedure can decode many cyclic and BCH codes up to their actual minimum distance and is seen to be a generalization of Peterson's procedure.

II. Preliminaries

In this section, we give a brief review of the Peterson decoding procedure and the Fundamental Iterative Algorithm for ease of later reference.

Let $g(x)$ be the generator polynomial of a cyclic code of length n over GF(q) and let d be the actual minimum distance of this code. The code is then capable of correcting up to $t = \lfloor (d-1)/2 \rfloor$ errors. Let $e(x) = \sum_{\mu=1}^{\nu} Y_\mu X^{a_\mu}$ with $\nu \le t$, $0 \le a_1 < a_2 < \cdots < a_\nu < n$ and $Y_\mu \ne 0$ for $\mu = 1, 2, \cdots, \nu$, be an error polynomial resulted from some transmitted code polynomial $v(x)$. Then the received polynomial is $r(x) = v(x) + e(x)$. Suppose, for some primitive nth root of unity $\beta \in \mathrm{GF}(q^m)$, $g(\beta^k) = 0$. Then $v(\beta^k) = 0$ and $r(\beta^k) = e(\beta^k)$. Thus the syndrome term $S_k = e(\beta^k)$ can be computed from the received polynomial. Furthermore, we have $S_{qk} = S_k^q$ and $S_{n+k} = S_k$.

For BCH codes, and cyclic codes in general, $d_0 - 1$ "consecutive" syndrome terms are known where d_0 denotes the BCH bound. Suppose $g(\beta^{b+ic}) = 0$ where b is any integer, c is an integer relatively prime to n and $i = 0, 1, \cdots, d_0 - 2$. Then $S_b, S_{b+c}, \cdots, S_{b+(d_0-2)c}$ are known, where

$$\begin{aligned} S_{b+ic} &= e\left(\beta^{b+ic}\right) \\ &= \sum_{\mu=1}^{\nu} Y_\mu \left(\beta^{a_\mu}\right)^{b+ic} \\ &= \sum_{\mu=1}^{\nu} Y_\mu X_\mu^{b+ic} \end{aligned} \tag{1}$$

with $X_\mu = \beta^{a_\mu}$ and $i = 0, 1, \cdots, d_0 - 2$.

Manuscript received July 24, 1990; revised May 7, 1991. This work was supported by the National Science Foundation under Grant NCR-8716953. This work was presented in part at the 1990 IEEE International Symposium on Information Theory, San Diego, CA, January 14–19, 1990.

G.-L. Feng was with the Department of Computer Science and Electrical Engineering, Lehigh University, Bethlehem, PA 18015. He is now with the Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, LA 70504.

K. K. Tzeng is with the Department of Computer Science and Electrical Engineering, Lehigh University, Bethlehem, PA 18015.

IEEE Log Number 9102344.

The problem of decoding BCH codes is to determine the error locations X_μ's and the error magnitudes Y_μ's from the d_0-1 syndromes. The procedure devised by Peterson and generalized to the nonbinary cases by Gorenstein and Zierler [16] is to separate the issues of determining the error locations and the error magnitudes by first defining the error-locator polynomial

$$\sigma(z)=\prod_{\mu=1}^{\nu}(z-X_\mu^c)$$
$$=z^\nu+\sigma_1 z^{\nu-1}+\cdots+\sigma_{\nu-1}z+\sigma_\nu.$$

Then,

$$S_{b+uc}+\sigma_1 S_{b+(u-1)c}+\cdots+\sigma_\nu S_{b+(u-\nu)c}=0,\qquad \text{for } u\geq\nu. \tag{2}$$

These recurrence relations have been referred to as the generalized Newton identities.

The problem is now transformed to the determination of $\sigma(z)$ from the d_0-1 syndromes through (2). After $\sigma(z)$ is determined, the error locations will be given by the roots of $\sigma(z)$. Then the error magnitudes can be determined easily. The procedure thus consists of the following steps:

1) calculate the syndromes S_k,
2) determine $\sigma(z)$ from (2),
3) determine the error locations X_μ,
4) determine the error magnitudes Y_μ.

The second step is now best accomplished by the Berlekamp–Massey algorithm. The third step can be handled efficiently by the Chien search and the last step is completed by using Forney's formula [16]. Alternatively, after the σ_μ's are determined, the unknown syndromes can be determined through (2). When $S_0, S_1, \cdots, S_{n-1}$ all become known, then, as shown by Blahut [19], an inverse Fourier transform will determine all the error locations and all the error magnitudes.

However, this procedure can only decode up to $t=\lfloor(d_0-1)/2\rfloor$ errors. Similarly, when the lower bound on the minimum distance of the code is given by the HT bound d_{HT} or the Roos bound d_{Roos}, the generalized algorithms can only decode up to $\lfloor d_{\text{HT}}-1/2\rfloor$ or $\lfloor d_{\text{Roos}}-1/2\rfloor$ errors [4], [8]. To make it clear, we shall also use d_{BCH} to denote the BCH bound.

In the next section, we shall present a more general procedure for decoding up to the actual minimum distance. The main feature of this procedure is the incorporation of the fundamental iterative algorithm in determining the error locator polynomial from a nonrecurrent syndrome dependence relation.

This algorithm is for finding the smallest initial set of dependent columns in an $M\times N$ matrix over any field F with rank less than N.

Let

$$A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N}\\ a_{21} & a_{22} & \cdots & a_{2N}\\ \vdots & \vdots & \cdots & \vdots\\ a_{M1} & a_{M2} & \cdots & a_{MN}\end{pmatrix}$$

be such a matrix.

For $0\leq l<N$, the first $l+1$ columns of A are said to be linearly dependent if there exist $c_1,\cdots,c_l$ in F, not all zero, such that,

$$a_{i,l+1}+c_1a_{i,l}+\cdots+c_la_{i,1}=0,\qquad \text{for } i=1,2,\cdots,M. \tag{3}$$

Given matrix A, the algorithm is to determine the minimum l and $c_1,\cdots,c_l$ such that (3) holds. This algorithm has been developed in [4]. A brief review of this algorithm is given in the following.

Let $C(x)=c_0+c_1x+\cdots+c_lx^l$ and $a^{(i)}(x)=a_{i,0}+a_{i,1}x+\cdots+a_{i,N}x^N$, where $c_0=1$ and $a_{i,0}=1$, for $i=1,2,\cdots,M$. For $l+1\leq n\leq N$, let $[C(x)a^{(i)}(x)]_n$ be the coefficient of x^n in $C(x)a^{(i)}(x)$, namely

$$\left[C(x)a^{(i)}(x)\right]_n=c_0a_{i,n}+c_1a_{i,n-1}+\cdots+c_la_{i,n-l}$$
$$=\sum_{j=0}^{l}c_ja_{i,n-j}. \tag{4}$$

Let

$$C^{(i-1,j)}(x)=c_0^{(i-1,j)}+c_1^{(i-1,j)}x+\cdots+c_{j-1}^{(i-1,j)}x^{j-1}$$
$$=\sum_{k=0}^{j-1}c_k^{(i-1,j)}x^k, \tag{5}$$

where $1\leq i\leq M$, be defined as the polynomial with the property that

$$\left[C^{(i-1,j)}(x)a^{(h)}(x)\right]_j=a_{h,j}+c_1^{(i-1,j)}a_{h,j-1}+\cdots$$
$$+c_{j-1}^{(i-1,j)}a_{h,1}=0,\qquad \text{for } h\leq i-1.$$

Let

$$d_{i,j}=\left[C^{(i-1,j)}(x)a^{(i)}(x)\right]_j$$
$$=a_{i,j}+c_1^{(i-1,j)}a_{i,j-1}+\cdots+c_{j-1}^{(i-1,j)}a_{i,1}. \tag{6}$$

Then we have the following.

Fundamental Iterative Algorithm:

Step 1) Empty Tables D and C, $1\Rightarrow s$, $1\Rightarrow r$, $1\Rightarrow C^{(0,s)}(x)$.

Step 2) Compute $d_{r,s}=[C^{(r-1,s)}(x)a^{(r)}(x)]_s$.

Step 3) If $d_{r,s}=0$, then

a) If $r=M$, then $s\Rightarrow l+1$, $C^{(r-1,s)}(x)\Rightarrow C(x)$, stop;

b) otherwise $C^{(r,s)}(x)=C^{(r-1,s)}(x)$ and $r+1\Rightarrow r$ and return to Step 2).

Step 4) If $d_{r,s}\neq 0$, then

a) If there exists $d_{r,u}\in D$ for some $1\leq u<s$, then

$$C^{(r,s)}(x)=C^{(r-1,s)}(x)-\frac{d_{r,s}}{d_{r,u}}C^{(u)}(x)x^{s-u}$$

and return to Step 3a);

b) otherwise, $d_{r,s}$ is stored in D, $C^{(0,s+1)}(x)=C^{(s)}(x)=C^{(r-1,s)}(x)$ and $C^{(s)}(x)$ is stored in C, then $s+1\Rightarrow s$, $1\Rightarrow r$ and return to Step 2).

The final s and $C^{(r-1,s)}(x)$ obtained from applying the Fundamental Iterative Algorithm is the solution of the general problem with minimum possible s.

III. Decoding Procedure Based on Nonrecurrent Syndrome Dependence Relations

We now proceed to derive the procedure that is capable of decoding many cyclic and BCH codes up to their actual minimum distance. The main departure in concept from the Peterson procedure is to examine the whole set of known syndromes and properly select, for full utilization in decoding, a set of syndromes that are not necessarily consecutive.

Let us consider a cyclic or BCH code for which $g(\beta^{b+ic_1+jc_2}) = 0$, where $(n, c_1) = 1$, $(n, c_2) = 1, i = 0, i_1, i_2, \cdots, i_t$, $j = 0, j_1, j_2, \cdots, j_p$ with $t \le p+1$, and $0 < i_1 < i_2 < \cdots < i_t$, $0 < j_1 < j_2 < \cdots < j_p$. Then, among the known syndromes, we have

$$S_{b+ic_1+jc_2} = r(\beta^{b+ic_1+jc_2}) = e(\beta^{b+ic_1+jc_2}) = \sum_{\mu=1}^{\nu} Y_\mu (\beta^{a_\mu})^{b+ic_1+jc_2} = \sum_{\mu=1}^{\nu} Y_\mu X_\mu^{b+ic_1+jc_2},$$

$$\text{for } i = 0, i_1, i_2, \cdots, i_t, \quad j = 0, j_1, j_2, \cdots, j_p,$$

where $X_\mu = \beta^{a_\mu}$. Let

$$S = \begin{pmatrix} S_b & S_{b+i_1c_1} & \cdots & S_{b+i_tc_1} \\ S_{b+j_1c_2} & S_{b+i_1c_1+j_1c_2} & \cdots & S_{b+i_tc_1+j_1c_2} \\ S_{b+j_2c_2} & S_{b+i_1c_1+j_2c_2} & \cdots & S_{b+i_tc_1+j_2c_2} \\ \vdots & \vdots & & \vdots \\ S_{b+j_pc_2} & S_{b+i_1c_1+j_pc_2} & \cdots & S_{b+i_tc_1+j_pc_2} \end{pmatrix}_{(p+1)\times(t+1)}.$$

Then $S = XYZ$, where

$$X = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ X_1^{j_1c_2} & X_2^{j_1c_2} & \cdots & X_\nu^{j_1c_2} \\ X_1^{j_2c_2} & X_2^{j_2c_2} & \cdots & X_\nu^{j_2c_2} \\ \vdots & \vdots & & \vdots \\ X_1^{j_pc_2} & X_2^{j_pc_2} & \cdots & X_\nu^{j_pc_2} \end{pmatrix}_{(p+1)\times\nu}$$

$$Y = \begin{pmatrix} Y_1X_1^b & 0 & \cdots & 0 \\ 0 & Y_2X_2^b & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & Y_\nu X_\nu^b \end{pmatrix}_{\nu\times\nu}$$

and

$$Z = \begin{pmatrix} 1 & X_1^{i_1c_1} & \cdots & X_1^{i_tc_1} \\ 1 & X_2^{i_1c_1} & \cdots & X_2^{i_tc_1} \\ \vdots & \vdots & & \vdots \\ 1 & X_\nu^{i_1c_1} & \cdots & X_\nu^{i_tc_1} \end{pmatrix}_{\nu\times(t+1)}.$$

For the procedure described in this section, we consider codes for which S can be so chosen that X will be of full rank, i.e., $\text{rank}(X) = \nu$. As a consequence, the column rank of S will be the same as the column rank of Z. Furthermore, the same linear dependence relations will exist among the corresponding columns of S and among those of Z. Since $\nu \le t$, the column rank of Z will be at most ν. Then the column rank of S will also be at most ν. Suppose the column rank of S is $\lambda \le \nu$. Then the first $\lambda+1$ columns of S will be linearly dependent and there will exist $f_1, f_2, \cdots, f_\lambda$ such that

$$S_{b+i_\lambda c_1+jc_2} + f_1 S_{b+i_{\lambda-1}c_1+jc_2} + f_2 S_{b+i_{\lambda-2}c_1+jc_2} + \cdots + f_\lambda S_{b+jc_2} = 0, \quad \text{for } j = 0, j_1, j_2, \cdots, j_p \quad (7)$$

and

$$X_\mu^{i_\lambda c_1} + f_1 X_\mu^{i_{\lambda-1}c_1} + f_2 X_\mu^{i_{\lambda-2}c_1} + \cdots + f_{\lambda-1} X_\mu^{i_1c_1} + f_\lambda = 0, \quad \text{for } \mu = 1, 2, \cdots, \nu.$$

Let

$$f(z) = z^{i_\lambda} + f_1 z^{i_{\lambda-1}} + f_2 z^{i_{\lambda-2}} + \cdots + f_{\lambda-1} z^{i_1} + f_\lambda.$$

Then

$$f(X_\mu^{c_1}) = 0, \quad \text{for } \mu = 1, 2, \cdots, \nu. \quad (7')$$

Now let $\sigma(z) = \prod_{\mu=1}^{\nu}(z - X_\mu^{c_1})$. Apparently, we have $\nu \le i_\lambda$ and $\sigma(z)$ as a factor of $f(z)$, namely, $f(z) = h(z)\sigma(z)$ for some $h(z) \in \text{GF}(q^m)[z]$. Then, instead of attempting to determine the error-locator polynomial from Newton's identities, we now have a nonrecurrent dependence relation among the syndromes as given by (7) to solve for a polynomial $f(z)$ that contains the error-locator polynomial as a factor.

It is seen that this step can be accomplished by the Fundamental Iterative Algorithm presented in the previous section. Let $C(x) = 1 + c_1x + \cdots + c_\lambda x^\lambda$ be a solution obtained by the Fundamental Iterative Algorithm, then $f_i = c_i$ for $i = 1, 2, \cdots, \lambda$. This is illustrated by the following example.

Example 1: Let us consider the (39,15) binary BCH code generated by $g(x) = m_1(x)m_3(x)$. Let α be a primitive element of GF(2^{12}) and $\beta = \alpha^{105}$, namely β is a 39th root of unity in GF(2^{12}). Then $g(\beta^k) = 0$ for $k = 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15, 16, 18, 20, 21, 22, 24, 25, 27, 30, 32, 33, 36$. The BCH bound, the HT bound and the Roos bound all give $d \ge 7$, but the actual minimum distance of this code is 10. According to the BCH bound, the Berlekamp–Massey algorithm can decode up to three errors using the syndromes $S_1, S_2, S_3, S_4, S_5, S_6$. However, since $S_8, S_9, S_{10}, S_{11}, S_{12}$ are known, we have $S_{b+ic_1+jc_2}$ for $b = 1$, $c_1 = 1$, $c_2 = 1$, $i = 0, 1, 2, 7, 8$, $j = 0, 1, 2, 3$ and

$$S = \begin{pmatrix} S_1 & S_2 & S_3 & S_8 & S_9 \\ S_2 & S_3 & S_4 & S_9 & S_{10} \\ S_3 & S_4 & S_5 & S_{10} & S_{11} \\ S_4 & S_5 & S_6 & S_{11} & S_{12} \end{pmatrix}_{4\times5}.$$

Thus, $S = XYZ$ and for $\nu \le 4$

$$X = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_\nu \\ X_1^2 & X_2^2 & \cdots & X_\nu^2 \\ X_1^3 & X_2^3 & \cdots & X_\nu^3 \end{pmatrix}_{4\times\nu}$$

$$Y = \begin{pmatrix} Y_1X_1 & 0 & \cdots & 0 \\ 0 & Y_2X_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & Y_\nu X_\nu \end{pmatrix}_{\nu\times\nu}$$

$$Z = \begin{pmatrix} 1 & X_1 & X_1^2 & X_1^7 & X_1^8 \\ 1 & X_2 & X_2^2 & X_2^7 & X_2^8 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_\nu & X_\nu^2 & X_\nu^7 & X_\nu^8 \end{pmatrix}_{\nu\times5}.$$

Since X is a Vandermonde matrix, it is of rank ν, namely, X is of full rank. Then, the column rank of S will be the same as the column rank of Z. Applying the Fundamental Iterative Algorithm to S, we can determine $f(z)$.

For example, suppose $S_1 = \alpha^{3406}$ and $S_3 = \alpha^{1980}$. Then we have

$$S = \begin{pmatrix} \alpha^{3406} & \alpha^{2717} & \alpha^{1980} & \alpha^{2678} & \alpha^{3015} \\ \alpha^{2717} & \alpha^{1980} & \alpha^{1339} & \alpha^{3015} & \alpha^{2899} \\ \alpha^{1980} & \alpha^{1339} & \alpha^{3497} & \alpha^{2899} & \alpha^{1898} \\ \alpha^{1339} & \alpha^{3497} & \alpha^{3960} & \alpha^{1898} & \alpha^{3825} \end{pmatrix}_{4\times5}.$$

For $s = 1$, we have $C^{(0,1)}(x) = 1$.

Then $d_{11} = \alpha^{3406} \neq 0$. Since there is no $d_{1,u} \in D$, where $1 \leq u < 1$, $d_{11} \Rightarrow D$,

$$C^{(1)}(x) = C^{(0,1)}(x) = 1 \Rightarrow C.$$

For $s = 2$, we have $C^{(0,2)}(x) = C^{(1)}(x) = 1$.

Then

$$d_{12} = \alpha^{2717},\ C^{(1,2)}(x) = C^{(0,2)}(x) - \frac{d_{12}}{d_{11}} C^{(1)}(x) x^{2-1} = 1 + \alpha^{3406} x.$$

Next $d_{22} = \alpha^{2362} \neq 0$. Since there is no $d_{2,u} \in D$, where $1 \leq u < 2$, $d_{22} \Rightarrow D$,

$$C^{(2)}(x) = C^{(1,2)}(x) \Rightarrow C.$$

For $s = 3$, we have $C^{(0,3)}(x) = C^{(2)}(x) = 1 + \alpha^{3406} x$.

Then

$$d_{13} = \alpha^{2362},\ C^{(1,3)}(x) = C^{(0,3)}(x) - \frac{d_{13}}{d_{11}} C^{(1)}(x) x^{3-1} = 1 + \alpha^{3406} x + \alpha^{3051} x^2.$$

Next

$$d_{23} = 0,\ C^{(2,3)}(x) = C^{(1,3)}(x) = 1 + \alpha^{3406} x + \alpha^{3051} x^2.$$

Then $d_{33} = \alpha^{1375} \neq 0$. Since there is no $d_{3,u} \in D$, where $1 \leq u < 3$, $d_{33} \Rightarrow D$,

$$C^{(3)}(x) = C^{(2,3)}(x) \Rightarrow C.$$

For $s = 4$, we have $C^{(0,4)}(x) = C^{(3)}(x) = 1 + \alpha^{3406} x + \alpha^{3051} x^2$.

Then

$$d_{14} = \alpha^{42},\ C^{(1,4)}(x) = C^{(0,4)}(x) - \frac{d_{14}}{d_{11}} C^{(1)}(x) x^{4-1} = 1 + \alpha^{3406} x + \alpha^{3051} x^2 + \alpha^{731} x^3.$$

Next

$$d_{24} = \alpha^{3021},\ C^{(2,4)}(x) = C^{(1,4)}(x) - \frac{d_{24}}{d_{22}} C^{(2)}(x) x^{4-2} = 1 + \alpha^{3406} x + \alpha^{2377} x^2 + \alpha^{1723} x^3.$$

Then

$$d_{34} = \alpha^{538},\ C^{(3,4)}(x) = C^{(2,4)}(x) - \frac{d_{34}}{d_{33}} C^{(3)}(x) x^{4-3} = 1 + \alpha^{1060} x + \alpha^{3905} x^2 + \alpha^{2211} x^3.$$

Next $d_{44} = \alpha^{533} \neq 0$. Since there is no $d_{4,u} \in D$, where $1 \leq u < 4$, $d_{44} \Rightarrow D$,

$$C^{(4)}(x) = C^{(3,4)}(x) \Rightarrow C.$$

For $s = 5$, we have $C^{(0,5)}(x) = C^{(4)}(x) = 1 + \alpha^{1060} x + \alpha^{3905} x^2 + \alpha^{2211} x^3$.

Then

$$d_{15} = \alpha^{1102},\ C^{(1,5)}(x) = C^{(0,5)}(x) - \frac{d_{15}}{d_{11}} C^{(1)}(x) x^{5-1} = 1 + \alpha^{1060} x + \alpha^{3905} x^2 + \alpha^{2211} x^3 + \alpha^{1791} x^4.$$

Next

$$d_{25} = \alpha^{77},\ C^{(2,5)}(x) = C^{(1,5)}(x) - \frac{d_{25}}{d_{22}} C^{(2)}(x) x^{5-2} = 1 + \alpha^{1060} x + \alpha^{3905} x^2 + \alpha^{3295} x^3 + \alpha^{1144} x^4.$$

Then

$$d_{35} = \alpha^{813},\ C^{(3,5)}(x) = C^{(2,5)}(x) - \frac{d_{35}}{d_{33}} C^{(3)}(x) x^{5-3} = 1 + \alpha^{1060} x + \alpha^{3413} x^2 + \alpha^{1699} x^3 + \alpha^{3358} x^4.$$

Next

$$d_{45} = \alpha^{2695},\ C^{(4,5)}(x) = C^{(3,5)}(x) - \frac{d_{45}}{d_{44}} C^{(4)}(x) x^{5-4} = 1 + \alpha^{321} x + \alpha^{3246} x^2 + \alpha^{2791} x^3 + \alpha^{1463} x^4.$$

Since $r = 4 = M$, the algorithm stops. Thus, the five columns of S are linearly dependent and $f_4 = c_4 = \alpha^{1463}$, $f_3 = c_3 = \alpha^{2791}$, $f_2 = c_2 = \alpha^{3246}$, $f_1 = c_1 = \alpha^{321}$ and we have

$$f(z) = \alpha^{1463} + \alpha^{2791} z + \alpha^{3246} z^2 + \alpha^{321} z^7 + z^8.$$

Once $f(z)$ is obtained, we may also use the Chien search to determine the set U of nth roots of unity that are roots of $f(z)$.

Suppose $U = \{U_1^{c_1}, U_2^{c_1}, \cdots, U_\delta^{c_1}\}$, where $\nu \leq \delta \leq i_\lambda$, then

$$\{X_1^{c_1}, X_2^{c_1}, \cdots, X_\nu^{c_1}\} \subseteq \{U_1^{c_1}, U_2^{c_1}, \cdots, U_\delta^{c_1}\}.$$

Since $S_{b+ic_1+jc_2} = \sum_{\mu=1}^{\nu} Y_\mu X_\mu^{b+ic_1+jc_2}$, then there exist $W_1, W_2, \cdots, W_\delta$ such that

$$S_{b+ic_1+jc_2} = \sum_{\eta=1}^{\delta} W_\eta U_\eta^{b+ic_1+jc_2}. \tag{8}$$

For $\eta = 1, 2, \cdots, \delta$, if $U_\eta = X_\mu$, for some $\mu = 1, 2, \cdots, \nu$, then $W_\eta = Y_\mu \neq 0$, otherwise $W_\eta = 0$.

The problem now becomes that of determining δ, instead of ν error magnitudes where $\delta = |U|$. Clearly this can still be accomplished through the Forney formula if δ consecutive syndromes are known. If $\delta < d_0$, then there are enough syndromes available to accomplish this. If not, then we may determine the required additional syndromes through (7) as well as using the fact that $S_{qk} = S_k^q$. This step then gives not only the error locations but also the corresponding error magnitudes.

Example 1 (continued): For the obtained $f(z) = \alpha^{1463} + \alpha^{2791} z + \alpha^{3246} z^2 + \alpha^{321} z^7 + z^8$, using the Chien search we find that $X_1 = \alpha^{105} = \beta$, $X_2 = \alpha^{315} = \beta^3$, $X_3 = \alpha^{420} = \beta^4$ and $X_4 = \alpha^{630} = \beta^6$ are all 39th roots of unity of $f(z)$. In this case $\delta = 4 < d_0$, we can apply the Forney formula to (8) and have $Y_1 = 1$, $Y_2 = 1$, $Y_3 = 1$, and $Y_4 = 1$. Thus this received vector has four errors and the errors are in 2nd, 4th, 5th, and 7th positions.

To summarize, the steps in our procedure are:

1) compute S_k for $k = b + ic_1 + jc_2$, $i = 0, i_1, i_2, \cdots, i_t$ and $j = 0, j_1, j_2, \cdots, j_p$,
2) determine $f(z)$ from (7) using the Fundamental Iterative Algorithm,
3) find the roots of $f(z)$ that are nth roots of unity through a Chien search,
4) determine the error magnitudes and the error locations from (8) using Forney's formula.

It should be noted that when the syndromes used are consecutive, namely, when $i_k = k$ and $j_l = l$ for $k = 1, 2, \cdots, t$ and $l = 1, 2, \cdots, p$, then (7) becomes a set of recursive equations, the generalized Newton identities. In this case, the Fundamental Iterative Algorithm will become the Berlekamp–Massey algorithm as shown in [4]. Thus this procedure is seen as a generalization of the Peterson procedure.

Another example to illustrate this procedure follows. In this example, we will introduce some notation, which will be used often later.

Example 2: Consider the (33,11) binary cyclic code generated by $g(x)=m_1(x)m_3(x)m_{11}(x)$. For this code, $d_0=8$, but the actual minimum distance is 11. The decoding of this code was discussed in [15] and [13], but no method to decode five errors was given in [15] and the method in [13] has to test all the field elements of GF(2^5) for S_7 in order to decode five errors. On the other hand, $d_{\text{Roos}}=10$ and $d_{\text{HT}}=8$, thus, the decoding method given in [4], [8] cannot decode five errors either.

Let α be a primitive element of GF(2^{10}) and $\beta=\alpha^{31}$, namely β is a 33th root of unity. From the received vector we can calculate S_k for $k=1,2,4,8,16,32,31,29,25,17;3,6,12,24,15,30,27,21,9,18;11,22$. To correct up to five errors, we let

$$S=\begin{pmatrix} S_{24} & S_1 & S_{11} & S_{15} & S_{25} & S_2 \\ S_1 & S_{11} & S_{21} & S_{25} & S_2 & S_{12} \\ S_{11} & S_{21} & S_{31} & S_2 & S_{12} & S_{22} \\ S_{21} & S_{31} & S_8 & S_{12} & S_{22} & S_{32} \\ S_{31} & S_8 & D_{18} & D_{22} & S_{32} & S_9 \end{pmatrix}.$$

Then $b=24$, $c_1=c_2=10$, $i_1=1$, $i_2=2$, $i_3=9$, $i_4=10$, $i_5=11$, and $j_1=1$, $j_2=2$, $j_3=3$, $j_4=4$.

(In the following this syndrome pattern is expressed by "$b, b+i_1c_1,\cdots, b+i_tc_1$ and $0, j_1c_2,\cdots, j_{t-1}c_2$" = "24, 1, 11, 15, 25, 2 and 0, 10, 20, 30, 40".)

Suppose $S_1=\alpha^{638}$, $S_3=\alpha^{590}$ and $S_{11}=\alpha^{682}$. Then

$$S=\begin{pmatrix} \alpha^{628} & \alpha^{638} & \alpha^{682} & \alpha^{233} & \alpha^{671} & \alpha^{253} \\ \alpha^{638} & \alpha^{682} & \alpha^{841} & \alpha^{671} & \alpha^{253} & \alpha^{341} \\ \alpha^{682} & \alpha^{841} & \alpha^{935} & \alpha^{253} & \alpha^{341} & \alpha^{314} \\ \alpha^{841} & \alpha^{935} & \alpha^{1012} & \alpha^{341} & \alpha^{314} & \alpha^{979} \\ \alpha^{935} & \alpha^{1012} & \alpha^{259} & \alpha^{314} & \alpha^{979} & \alpha^{659} \end{pmatrix}.$$

Applying the Fundamental Iterative Algorithm, we have $f_1=\alpha^{1001}$, $f_2=\alpha^{529}$, $f_3=\alpha^{467}$, $f_4=\alpha^{226}$, and $f_5=\alpha^{930}$, namely $f(z)=z^{11}+\alpha^{1001}z^{10}+\alpha^{529}z^9+\alpha^{467}z^2+\alpha^{226}z+\alpha^{930}$. Through the Chien search, we found that $f(z)$ has only five roots which are 33rd roots of unity in GF(2^{10}), namely $U=\{1,\alpha^{31},\alpha^{62},\alpha^{124},\alpha^{186}\}=(1,\beta,\beta^2,\beta^4,\beta^6\}$. Since $\delta=5<d_0=8$, using the Forney formula and $S_{24}=\alpha^{628}$, $S_1=\alpha^{638}$, $S_{11}=\alpha^{682}$, $S_{21}=\alpha^{841}$, $S_{31}=\alpha^{935}$, we know that 1, $\alpha^{31}=\beta$, $\alpha^{62}=\beta^2$, $\alpha^{124}=\beta^4$, and $\alpha^{186}=\beta^6$ are the error locators and the error magnitudes are all equal to 1.

Generally speaking, δ may be greater than 5 (since deg $f(z)=11$). If $\delta=11$, from (7), we have

$$S_{26}+f_1S_{16}+f_2S_6+f_3S_2+f_4S_{25}+f_5S_{15}=0.$$

Since $S_{15},S_{25},S_2,S_6,S_{16}$ are known, from the last equation we can obtain S_{26} (we express it as "15,25,2,6,16 → 26"). Then we have $S_5=S_{26}^{2^2}$, $S_{10}=S_{26}^{2^3}$, and $S_7=S_{26}^{2^5}$ (we express these as "26 → 5 → 10 → 7"). Finally applying the Forney formula to S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, and S_{11} (we express it as "$F(1,2,3,4,5,6,7,8,9,10,11)$" or simply "$F(1$–$11)$"), we can determine the error locations and magnitudes.

In the following section, we give additional examples to illustrate the application of this procedure. In particular, we give a more detailed description on the application of this procedure to the decoding of the (23,12) Golay code.

IV. Additional Examples and Table

Example 3: We consider the (21,7) binary code with $g(x)=m_1(x)m_3(x)m_7(x)m_9(x)$. For this code $d=8$, $d_{\text{Roos}}=8$, and $d_{\text{HT}}=6$. But the method in [4], [8] cannot correct three errors. From the received vector, we can calculate S_k for $k=1,2,4,8,16,11;3,6,12;7,14;9,18,15$. We can decode up to three errors using the following syndrome pattern,

$$S=\begin{pmatrix} S_1 & S_2 & S_6 & S_7 \\ S_2 & S_3 & S_7 & S_8 \\ S_3 & S_4 & S_8 & S_9 \end{pmatrix},$$

namely, S is (1,2,6,7 and 0,1,2), where $b=1$, $c_1=c_2=1$, $i_1=1$, $i_2=5$, $i_3=6$, and $j_1=1$, $j_2=2$.

If f_1,f_2,f_3 are nonzero, then from $f_3S_7+f_2S_8+f_1S_{12}+S_{13}=0$, where only S_{13} is unknown, we can find S_{13} (7,8,12 → 13). Then $S_5=S_{13}^2$ (13 → 5). Since $\delta\le i_3=6$, using S_1,S_2,S_3,S_4, S_5,S_6 and the Forney formula we can determine the errors and the magnitudes from (8) (F(1-6)).

Example 4: Now we consider in detail the (23,12) Golay code with $n=23$, $g(x)=m_1(x)$, $d=7$, and $d_{\text{BCH}}=5$. Since the generator $g(x)$ has roots β^1, β^2, β^4, β^8, β^{16}, β^9, β^{18}, β^{13}, β^3, β^6 and β^{12}, where β is a primitive 23rd root of unity in GF(2^{11}), we can determine S_1, S_2, S_4, S_8, S_{16}, S_9, S_{18}, S_{13}, S_3, S_6, and S_{12}. Since $S_0=0$ when the number of errors is even and $S_0=1$ otherwise, we can assume that S_0 is known. From the syndromes we have

$$S=\begin{pmatrix} S_{16} & S_{12} & S_6 & S_4 \\ S_{13} & S_9 & S_3 & S_1 \\ S_{12} & S_8 & S_2 & S_0 \end{pmatrix}=XYZ,$$

where

$$X=\begin{pmatrix} 1 & \cdots & 1 \\ X_1^{-3} & \cdots & X_\nu^{-3} \\ X_1^{-4} & \cdots & X_\nu^{-4} \end{pmatrix},$$

$$Y=\begin{pmatrix} X_1^{16} & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & X_\nu^{16} \end{pmatrix},$$

$$Z=\begin{pmatrix} 1 & X_1^{-4} & X_1^{-10} & X_1^{-12} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_\nu^{-4} & X_\nu^{-10} & X_\nu^{-12} \end{pmatrix},$$

and $1\le\nu\le 3$.

Then $b=16$, $c_1=-2$, $c_2=-1$, $i_1=2$, $i_2=5$, $i_3=6$, and $j_1=3$, $j_2=4$. Since rank$(X)=\nu$ (See Lemma 3 in [17]), then the column rank of S is the same as the column rank of Z and the same linear dependence relations exist among the corresponding columns of S and Z. We consider the following cases.

a) If $\nu=1$, then the first two columns of Z are linearly dependent and those of S are also linearly dependent. Thus from the linear dependence relation of the first two columns of S we can find X_1.

b) If $\nu=2$, then the first three columns of Z are linearly dependent and those of S are also linearly dependent. Suppose that the coefficients for this linear dependence are f_2, f_1, and 1. Thus from the linear dependence relation we know that X_1^{-2} and X_2^{-2} must be roots of $f_2+f_1z^2+z^5=0$. This equation has at most five 23rd roots of unity in GF(2^{11}). From these roots we can find X_1 and X_2 using the Forney formula and S_0, S_1, S_2, S_3, and S_4, with $S_0=0$.

c) If $\nu=3$, then we consider two cases.

1) The first three columns of Z are linearly dependent and those of S are linearly dependent. Then let $S_0=1$, following the same procedure as in Case b).

TABLE I

n	k	d	t	d_0	G	Syndrome Patterns	Decoding Procedure
17	9	5	2	4	1**	1,8,15 and 0,1	A $F(1,8)$
21	9	8	3	6	0,1,3,7	0,1,2,6 and 0,1,2	A 6,7,11 → 5; F(1-6)
	7	8	3	5	1,3,7,9	1,2,6,7 and 0,1,2	A 7,8,12 → 13 → 5; F(1-6); Exa. 3
	6	8	3	6	0,1,3,7,9	1,2,6,7 and 0,1,2	A 7,8,12 → 13 → 5; F(1-6)
23	12	7	3	5	1**	12,8,2,0 and 4,3,0	A' 1,8,16 → 5; F(1-6); Exa. 4
	11	8	3	6	0,1**	12,8,2,0 and 4,3,0	A 1,8,16 → 5; F(1-6)
31	21	5	2	4	1,5	4,8,9 and 0,1	A 16,20 → 21 → 22; 18,22 → 23 → 29 → 30; F(29-2)
	21	5	2	4	1,7	4,16,1 and 0,3	A 2,14 → 30 → 29 → 27; F(27-2)
	20	6	2	4	0,1,5	0,1,2 and 0,8	A $F(0,1)$
	20	6	2	4	0,1,7	1,4,7 and 0,24	A $F(1,4)$
	16	7	3	5	1,5,7	0,7,8,18 and 0,1,2	A' 7,14,25 → 15; 9,16,27 → 17; 5,12,23 → 11; F(0-17)
	15	8	3	5	0,1,5,7	0,7,8,18 and 0,1,2	A 7,14,25 → 15; 9,16,27 → 17; 5,12,23 → 11; F(0-17)
33	11	11	5	8	1,3,11	24,1,11,15,25,2 and 0,10,20,30,40	A 15,25,2,6,16 → 26 → 5 → 10 → 7; F(1-11). Exa. 2
	10	12	5	10	0,1,3,11	12,17,22,24,29,1 and 0,5,10,15,20	A 15,25,27,32,4 → 20; F(1-9)
35	16	7	3	6	1,5,7	1,2,4,5 and 0,3,6	A F(7-10)
	15	8	3	6	0,1,5,7	1,2,4,5 and 0,3,6	A F(7-10)
	13	8	3	6	1,5,7,15	1,2,4,5 and 0,3,6	A F(7-10)
	12	8	3	6	0,1,5,7,15	1,2,4,5 and 0,3,6	A F(7-10)
	7	14	6	12	0,1,3,5**	31,32,33,34,0,1,8 and 0,1,2,3,4,5	A 8,9,10,11,12,19 → 7; F(1-12) Exa. 5
39	26	6	2	4	0,1**	0,1,4 and 0,1	A 10,11 → 14 → 37 → 38; F(37-2)
	24	6	2	4	0,1,13	0,1,4 and 0,1	A 10,11 → 14 → 37 → 38; F(37-2)
	15	10	4	7	1,3**	1,2,3,8,9 and 0,1,2,3	A 9,10,11,16 → 17 → 7; F(1-7). Exa. 1
	14	10	4	8	0,1,3**	1,2,3,8,9 and 0,1,2,3	A 9,10,11,16 → 17; F(1-7)
43	29	6	2	4	1**	1,21,41 and 0,1	A $F(1,21)$
45	23	7	3	6	1,5,21	31,32,33,38 and 0,1,2	A 19,20,21 → 26 → 41 → 43 → 44; F(38-44)
	22	8	3	6	0,1,5,21	0,19,38,76 and 0,1,2	A 19,20,21 → 26 → 41 → 43 → 44; F(38-44)
	16	10	4	8	0,1,3,7	43,44,0,1,11 and 0,1,2,3	A 3,4,6,16 → 5 → 10; F(43-8)
	15	9	4	8	1,3,7,15	11,12,13,14,28 and 0,1,2,3	A 1,2,3,4 → 18 → 9; 2,3,4,19 → 5; F(1-17)
	15	10	4	8	1,7,9,15	13,14,15,16,17 and 0,1,2,13,14,15	A F(13-16). Exa. 7
	14	10	4	8	0,1,7,9,15	13,14,15,16,17 and 0,1,2,13,14,15	A F(13-16)
	14	10	4	8	0,1,3,7,15	13,14,15,16,17 and 0,1,2,13,14,15	A F(13-16)
	12	10	4	8	0,1,3,7,9	43,44,0,1,6 and 0,1,2,3	A 23,24,26,31 → 25; F(0-8)
	11	9	4	8	1,3,7,15,21	11,12,13,14,15 and 0,1,2,17,18,19	A F(11-14)
	9	12	5	9	1,5,7,9,15	13,14,15,16,25,26 and 0,1,2,3,4	A 22,23,25,34,35 → 24 → 3 → 6 → 12; F(1-12)
45	8	12	5	9	0,1,5,7,9,15	13,14,15,16,25,26 and 0,1,2,3,4	A 22,23,25,34,35 → 24 → 3 → 6 → 12; F(1-12)
51	35	5	2	4	1,9	1,8,15 and 0,1	A $F(1,8)$
	34	6	2	4	0,1,9	1,8,15 and 0,1	A $F(1,8)$
	34	6	2	4	0,1,5	0,1,4 and 0,1	A 4,7 → 3; F(1-4)
	32	6	2	4	0,1,5,17	0,1,4 and 0,1	A 4,7 → 3; F(1-4)
	27	5	2	4	1,9,19	1,8,15 and 0,1	A $F(1,8)$
	27	8	3	5	1,3,9	0,2,8,12 and 0,1,4	A' 16,18,24 → 28 → 5; F(1-6)
	26	8	3	6	0,1,3,9	0,2,8,12 and 0,1,4	A 16,18,24 → 28 → 5; F(1-6)
	25	8	3	5	1,3,9,17	1,2,15,16 and 0,1,2	A 6,7,21 → 20; F(1-15)
	24	8	3	6	0,1,3,9,17	0,1,2,15 and 0,1,2	A 3,4,18 → 5; F(1-15)
	19	10	4	6	1,3,9,19	33,38,42,45,49 and 0,5,10,15	A 47,1,8,12 → 5 → 10; 2,7,14,18 → 11; F(1-16)
	17	12	5	6	1,3,9,17,19	32,47,48,49,50,0 and 0,1,2,3,4	A' 1,6,17,18,19 → 20; 3,18,19,20,21 → 22; F(1-19)
	16	12	5	10	0,1,3,9,17,19	32,47,48,49,50,0 and 0,1,2,3,4	A 1,6,17,18,19 → 20; 3,18,19,20,21 → 22; F(1-19)
55	35	5	2	4	1**	7,8,9 and 0,9	A $F(7,8)$

** Indicates that the code is a BCH code.
d Actual minimum distance.
t Maximum number of errors correctable.
d_0 BCH bound.
A Indicates that the code can be decoded by the indicated syndrome pattern following the procedure in Section III.
A' Indicates that the decoding syndrome pattern contains unknown S_0 ($S_0 = 0$ or 1).

2) The first three columns of Z are linearly independent and those of S are linearly independent. However, the four columns of Z must be linearly dependent and so are those of S. Suppose that the coefficients for this linear dependence are f_3, f_2, f_1, and 1. From the linear dependence relation we know that X_1^{-2}, X_2^{-2} and X_3^{-2} must be roots of $f_3 + f_2 z^2 + f_1 z^5 + z^6 = 0$. This equation has at most six 23rd roots of unity in GF(2^{11}). If the number of roots is 5 or less, then follow the same procedure as in Case c1). If the number of roots is 6, then $f_3 \neq 0$, and we can find S_5 using the relation $f_3 S_5 + f_{21} S_1 + f_1 S_{18} + S_{16} = 0$. Using the Forney formula and S_1, S_2, S_3, S_4, S_5, and S_6, we can find X_1, X_2, and X_3.

Example 5: We consider the (35,7) binary code, where $n = 35$, $g(x) = m_0(x)m_1(x)m_3(x)m_5(x)$, $d = 14$ and $d_{\text{BCH}} = 12$ and $d_{\text{Roos}} = 12$[18]. From the received vector we can calculate S_k for $k = 0$; 1, 2, 4, 8, 16, 32, 29, 23, 11, 22, 9, 18; 3, 6, 12, 24, 13, 26, 17, 34, 33, 31, 27, 19; 5, 10, 20. Using the following syndrome pattern and the previous decoding procedure, we can decode up to six errors. Let

$$S = \begin{pmatrix} S_{31} & S_{32} & S_{33} & S_{34} & S_0 & S_1 & S_8 \\ S_{32} & S_{33} & S_{34} & S_0 & S_1 & S_2 & S_9 \\ S_{33} & S_{34} & S_0 & S_1 & S_2 & S_3 & S_{10} \\ S_{34} & S_0 & S_1 & S_2 & S_3 & S_4 & S_{11} \\ S_0 & S_1 & S_2 & S_3 & S_4 & S_5 & S_{12} \\ S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_{13} \end{pmatrix},$$

namely, S is (31,32,33,34,0,1,8 and 0,1,2,3,4,5). Then $b = 31, c_1 = c_2 = 1, i_1 = 1, i_2 = 2, i_3 = 3, i_4 = 4, i_5 = 5, i_6 = 12$; $j_1 = 1, j_2 = 2, j_3 = 3, j_4 = 4, j_5 = 5$. This is a BCH code. The Peterson decoding procedure and the decoding method in [4], [8] can only correct up to five errors. The method in [13] has to test 2^3 times to correct six errors, and [15] did not discuss the decoding of this code. Since $i_6 = 12$, δ may be 12. If $\delta = 12$, we first determine S_7 from

$$f_6 S_7 + f_5 S_8 + f_4 S_9 + f_3 S_{10} + f_2 S_{11} + f_1 S_{12} + S_{19} = 0,$$

where $f_6 \neq 0$ (8,9,10,11,12,19 $\rightarrow$ 7). Then we can determine the six error locations and magnitudes (F(1-12)).

Example 6: We consider a longer binary code, the (99,43) cyclic code with $n = 99$, ($g(x) = m_3(x)m_5(x)m_9(x)m_{11}(x)$, $d = 11$, $d_{\text{Roos}} = 9$, $d_{\text{HT}} = 8$, and $d_{\text{BCH}} = 7$. We have the syndromes S_k for $k = 3, 6, 12, 24, 48, 96, 93, 87, 75, 51$; 5, 10, 20, 40, 80, 61, 23, 46, 92, 85, 71, 43, 86, 73, 47, 94, 89, 79, 59, 19, 38, 76 53, 7, 14, 28, 56, 13, 26, 52; 9, 18, 36, 72, 45, 90, 81, 63, 27, 54; 11, 22, 44, 88, 77, 55.

In order to decode this code up to five errors we use the following syndrome pattern:

$$(9, 10, 43, 44, 51, 52 \text{ and } 0, 1, 2, 3, 4),$$

namely,

$$S = \begin{pmatrix} S_9 & S_{10} & S_{43} & S_{44} & S_{51} & S_{52} \\ S_{10} & S_{11} & S_{44} & S_{45} & S_{52} & S_{53} \\ S_{11} & S_{12} & S_{45} & S_{46} & S_{53} & S_{54} \\ S_{12} & S_{13} & S_{46} & S_{47} & S_{54} & S_{55} \\ S_{13} & S_{14} & S_{47} & S_{48} & S_{55} & S_{56} \end{pmatrix}.$$

Since $i_5 = 43$, δ may be 43. When $\delta = 43$, we have 38, 72, 73, 80, 81 $\rightarrow$ 39 and 38, 71, 72, 79, 80 $\rightarrow$ 37. Then from the properties about conjugates and S_{39}, S_{37}, we can find S_k for $k = 39, 78, 57, 15, 30, 60, 21, 42, 84, 69$ and 37, 74, 49, 98, 97, 95, 91, 83, 67, 35, 70, 41, 82, 65, 31, 62, 25, 50, 1, 2, 4, 8, 16, 32, 64, 29, 58, 17, 34, 68. Using the Forney formula on S_1–S_{43} (F(1–43)), we can determine the error locations and the magnitudes. Relevant information for decoding many cyclic and BCH codes of length up to 55 are presented in Table I.

V. Conclusion

In this correspondence, we have derived a procedure that can determine the error locations from nonrecurrent syndrome dependence relations. This procedure employs an algorithm that we have recently introduced as a basis for the derivation of the Berlekamp–Massey algorithm and its generalization. It can decode many cyclic and BCH codes up to their actual minimum distance and is seen to be a generalization of Peterson's procedure. It should be noted that not every cyclic code can be decoded up to its actual minimum distance by this procedure. However, Table I clearly indicates that a large percentage of cyclic codes can be so decoded. For a code with actual minimum distance d, the computation complexity of this procedure is $O(d^3)$. When matrix S consists of recurrent rows, the complexity reduces to $O(d^2)$ as the Fundamental Iterative Algorithm can be refined to become the Berlekamp–Massey algorithm or its generalization.

References

[1] E. R. Berlekamp, *Algebraic Coding Theory*. McGraw-Hill, New York, 1968.

[2] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122–127, Jan. 1969.

[3] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equation for decoding Goppa codes," *Inform. Contr.*, vol. 27, no. 1, pp. 87–99, Jan. 1975.

[4] G. L. Feng and K. K. Tzeng, "A generalization of the Berlekamp–Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1274–1287, Sept. 1991.

[5] ___, "A generalized Euclidean algorithm for multisequence shift-register synthesis," *IEEE Trans. Inform. Theory*, vol. 35, pp. 584–594, May 1989.

[6] C. R. P. Hartmann and K. K. Tzeng, "Generalizations of the BCH bound," *Inform. Contr.*, vol. 20, no. 5, pp. 489–498, June 1972.

[7] C. Roos, "A new lower bound for the minimum distance of a cyclic code," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 330–332, May 1983.

[8] G. L. Feng and K. K. Tzeng, "Decoding cyclic and BCH codes up to the Hartmann–Tzeng and Roos bounds," presented at the *IEEE Int. Symp. Inform. Theory*, San Diego, CA, Jan. 14–19, 1990.

[9] C. R. P. Hartmann, "Decoding beyond the BCH bound," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 441–444, May 1972.

[10] K. K. Tzeng and C. R. P. Hartmann, "Generalized BCH decoding," presented at the *IEEE Int. Symp. Inform. Theory*, Ashkelon, Israel, June 1973.

[11] C. R. P. Hartmann and K. K. Tzeng, "Decoding beyond the BCH bound using multiple sets of syndrome sequences," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 292–295, Mar. 1974.

[12] P. H. Chen, "Multisequence linear shift-register synthesis and its application to BCH decoding," *IEEE Trans. Commun.*, vol. 24, pp. 438–440, Apr. 1976.

[13] P. Stevens, "Extension of the BCH decoding algorithm to decode binary cyclic codes up to their maximum error-correction capacities," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1332–1340, Sept. 1988.

[14] M. Elia, "Algebraic decoding of the (23,12,7) Golay codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 150–151, Jan. 1987.

[15] P. Bours, J. C. M. Janssen, M. van Asperdt, and H. C. A. van Tilborg, "Algebraic decoding beyond e_{BCH} of some binary cyclic codes, when $e > e_{\text{BCH}}$," *IEEE Trans. Inform. Theory*, vol. 36, pp. 214–222, Jan. 1990.

[16] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed. Cambridge, MA: MIT Press, 1972.
[17] J. H. van Lint and R. M. Wilson, "On the minimum distance of cyclic codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 23–40, Jan. 1986.
[18] D. A. Bader and P. J. Stinson, "Table of lower bounds on the minimum distance of cyclic and BCH codes," NSF Res. Experiences for Undergraduates (REU) Program Summer Project Rep., Dept. of Comput. Sci. and Elect. Eng., Lehigh Univ., Bethlehem, PA, 1989.
[19] R. E. Blahut, *Fast Algorithm for Digital Processing*. Reading, MA: Addison-Wesley, 1985.

On Indecomposable Abelian Codes and Their Vertices

Karl-Heinz Zimmermann

Abstract—**Indecomposable nonsemisimple Abelian codes are investigated. It is illustrated that the minimal distance of every indecomposable Abelian code depends upon its associated vertex.**

Index Terms—**Indecomposable codes, Abelian group codes, minimum distance, relative projectivity, vertex.**

I. Preliminaries

In this article, we study linear codes with a given Abelian automorphism group. Codes of this kind were anticipated by Camion [3] in a more general point of view. We shall describe all indecomposable Abelian group codes and show that the minimal distance of such a code M is the product of the minimal distance of a semisimple Abelian group code and the minimal distance of the source module of M.

We first recall some basic facts about linear codes and particularly group codes. For this let $F = GF(q)$ be a finite field with $q = p^n$ elements (p prime). A *linear code* M of block length n is a subspace of F^n. A linear code with F-dimension k and block length n is denoted as (n, k)-code.

We shall show that all indecomposable Abelian group codes are product codes [11, pp. 568–570]. A (two-dimensional) *product code* M of two linear codes M_1 and M_2 is the code whose codewords are all the two-dimensional arrays for which columns are codewords in M_1 and rows are codewords in M_2. If M_i is a linear (n_i, k_i)-code with minimal distance d_i $(i = 1, 2)$, then M is a linear $(n_1 n_2, k_1 k_2)$-code with minimal distance $d_1 d_2$.

Throughout the article, let F denote a field of characteristic $p > 0$. All groups under consideration are assumed to be finite. A central role plays in the following the notion of group algebra [7, pp. 43–44].

For this let G be a finite group. The *group algebra* FG is the free F-module over G where G is regarded as an F-basis for FG. More explicitly, FG consists of all linear combinations

$$\sum_{g \in G} k_g g,$$

where $k_g \in F$ for all $g \in G$. Addition and multiplication with scalars $k \in F$ are defined as usual:

$$\sum_{g \in G} k_g g + \sum_{g \in G} l_g g = \sum_{g \in G} (k_g + l_g) g,$$

$$k\left(\sum_{g \in G} l_g g \right) = \sum_{g \in G} (k l_g) g.$$

Moreover, multiplication in G induces multiplication in FG as follows:

$$\left(\sum_{g \in G} k_g g \right)\left(\sum_{h \in G} l_h h \right) = \sum_{g \in G} \left(\sum_{uv = g} k_u l_v \right) g.$$

Hence, FG is an associative F-algebra with identity $1 = 1_F 1_G$ where 1_G and 1_F are the identity elements of G and F, respectively.

It is well known that every cyclic code of length n over a field F may be viewed as an ideal of the group algebra FG of the cyclic group G of order n [11, pp. 188–200]. This observation suggests the following generalization [4]:

Definition 1: Let G be a finite group of order n. Each right ideal M of the group algebra FG is called a code of length n over F. The right ideal M is also simply referred to as FG-code. If G is cyclic or Abelian, then every ideal M of FG is denoted as *cyclic* or *Abelian code*, respectively. An FG-code M is called indecomposable if M is an indecomposable right FG-module.

For each element $a = \sum_{g \in G} a_g g \in FG$ let $\operatorname{supp}(a) := \{g \in G | a_g \neq 0\}$ denote the *support* of a. The number $|\operatorname{supp}(a)|$ is called the *weight* of a (w.r.t. F-basis G). The *minimal distance* of an FG-code M (w.r.t. F-basis G) is given by

$$\operatorname{dist}(M) := \min\{|\operatorname{supp}(a)| | a \in M \setminus \{0\}\}.$$

We will often refer to the following trivial result [13, p. 6].

Proposition 1: Let H be a subgroup of G, and let ρ be a complete set of representatives of the right cosets of H in G. Then every element $a \in FG$ can be uniquely written as a finite sum of the form

$$a = \sum_{g \in \rho} a_g g,$$

with $a_g \in FH$ for all $g \in \rho$. Thus, FG is a left FH-module with F-basis ρ.

An FG-code M is called *semisimple* if the radical $\operatorname{Rad} M$ of M is the zero-module (0). The *radical* of M is the intersection of all maximal right FG-submodules of M. If G is a cyclic group of order n, then an FG-code M is semisimple iff $p \nmid n$. The class of semisimple cyclic codes has been exhaustively studied by many authors. See [11] for a list of references. We first make a few remarks about cyclic codes of length n where $p \nmid n$ or n is a power of p. For this let $G = \langle g | g^n = 1 \rangle$ be a cyclic group of order n.

1) If $p \nmid n$, then FG is a semisimple group algebra by Maschke's Theorem [7, p. 41], and, therefore, decomposes into a direct sum

$$FG = \bigoplus_{i=1}^{s} e_i FG$$

of minimal (simple) ideals $e_i FG$. $\{e_1, \cdots, e_s\}$ is a complete set of primitive idempotents of FG, which can be constructed in the following way [6, p. 56].

a) Determine a splitting field $\tilde{F}$ for G, i.e., a finite extension field of F containing the nth roots of unity.

Manuscript received August 29, 1989; revised March 20, 1991. This work was supported in part by the Fulbright Commission, while the author was working at Princeton University, Princeton, NJ. This work was presented in part at the 80th Birthday Marshall Hall Conference on Groups, Designs, and Codes, Burlington, VT, 1990.

The author is with the Mathematical Institute, University of Bayreuth, Postfach 10 12 51, D-8580 Bayreuth, Germany.

IEEE Log Number 9101889.