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 their generalizations, 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. This 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 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.

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 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 \( \text{GF}(q) \) and let \( d \) be the actual minimum distance of this code. The code is then capable of correcting up to \( \lfloor (d-1)/2 \rfloor \) errors. Let \( e(x) = \sum_{i \leq t, 0 \leq a_i < a, a_i < \cdots < a_n < n} Y_i X^{a_i} \) with \( \nu \geq 1 \). Then, \( e(x) = e(x) + e(x) \). Suppose, for some primitive \( n \)-th root of unity \( \beta \in \text{GF}(q^n) \), \( g(\beta^k) = 0 \). Then \( e(\beta^k) = 0 \) and \( r(\beta^k) = e(\beta^k) \). Thus the syndrome term \( S_i = e(\beta^k) \) can be computed from the received polynomial. Furthermore, we have \( S_{\nu} = S_{\nu} + S_{\nu} = S_{\nu} \).

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

\[
S_{a+b+c} = e(\beta^{a+b+c}) = \sum_{\nu=1}^{\nu=n} Y_\nu (\beta^s)^{a+b+c} = \sum_{\nu=1}^{\nu=n} X_\nu^{a+b+c} \quad (1)
\]

with \( \beta = \beta^s \) 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-8710953. 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 $X_{\nu}$'s and the error magnitudes $Y_{\nu}$'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

$$
\sigma(z) = \sum_{\mu=1}^{\nu} (z - X_{\mu}) = z^n + \sigma_1 z^{n-1} + \cdots + \sigma_{\nu-1} z + \sigma_{\nu},
$$

Then,

$$
S_{b_1} + \sigma_1 S_{b_2} + \cdots + \sigma_{\nu-1} S_{b_{\nu-1}} + \sigma_{\nu} = 0, \quad \text{for } \nu \geq 1.
$$

(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 $\sigma(z)$.

The error magnitudes can be determined easily. The procedure thus consists of the following steps:

1. calculate the syndromes $S_{b}$
2. determine $\sigma(z)$ from (2)
3. determine the error locations $X_{\nu}$
4. determine the error magnitudes $Y_{\nu}$

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 $\sigma_p$'s are determined, the unknown syndromes can be determined through (2). When $S_0, S_1, \ldots, S_{\nu-1}$ 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 $t = [(d_0 - 1)/2]$ 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 $[d_{HT} - 1/2]$ or $[d_{Roos} - 1/2]$ errors [4], [8].

To make it clear, we shall also use $d_{BCD}$ 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

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

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

$$
a_{1l+1} + c_1 a_{1l} + \cdots + c_l a_{1l} = 0, \quad \text{for } i = 1, 2, \cdots, M.
$$

(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 $C(x) = c_0 + c_1 x + \cdots + c_l x^l$ and $a_0(x) = a_{10} + a_{11} x + \cdots + a_{1l} x^l$, where $c_0 = 1$ and $a_{10} = 1$, for $i = 1, 2, \cdots, M$. For $1 \leq n \leq N$, let $[C(x)a_0(x)]_n$ be the coefficient of $x^n$ in $C(x)a_0(x)$, namely

$$
[C(x)a_0(x)]_n = c_0 a_{1n} + c_1 a_{1n-1} + \cdots + c_l a_{1n-l} = \sum_{j=0}^{l} c_j a_{1n-j}.
$$

(4)

Let

$$
C^{(r-1,i)}(x) = c_0^{(r-1,i)} + c_1^{(r-1,i)} x + \cdots + c_l^{(r-1,i)} x^{l-1}
$$

$$
= \sum_{k=0}^{l-1} c_k^{(r-1,i)} x^k,
$$

(5)

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

$$
[C^{(r-1,i)}(x)a_0(x)]_j = a_{r,j} + c_1^{(r-1,i)} a_{r,j-1} + \cdots + c_l^{(r-1,i)} a_{r,j-l}, \quad \text{for } h \leq j.
$$

(6)

Then we have the following.

**Fundamental Iterative Algorithm:**

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

**Step 2** Compute $d_{r,s} = [C^{(r-1,s)}(x)a_0(x)]_r$.

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

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

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

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

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

and return to Step 3a);
- b) otherwise, $d_{r,s}$ is stored in $D$, $C^{(r+1,s)}(x) = C^{(r-1,s)}(x)$ and $C^{(r+1,s)}(x)$ 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
\[ g(b^{i_1}c_1+b^{i_2}c_2) = 0, \]
where \( (n,c_1) = 1, (n,c_2) = 1, i_1 = 0, i_2 = 0, \ldots, i_n = 0, j_1 = 0, j_2 = 0, \ldots, j_l = 0, \]
with \( 1 \leq i_1 < i_2 < \cdots < i_n \leq \frac{n}{2} \) and \( 1 \leq j_1 < j_2 < \cdots < j_l \leq \frac{n}{2} \). Then, among the known syndromes, we have
\[ S_{b+i_1+j_1} = \sum_{\mu=1}^c Y_\mu (b^{i_1}c_1+b^{i_2}c_2)^{i_1+j_1}, \]
where \( X_\mu = b^{c_1} c_2 \). Let
\[ S = \begin{bmatrix} S_0 & S_{b+i_1+j_1} & \cdots & S_{b+j_1} \\ S_{b+i_2+j_2} & S_{b+i_1+j_1+j_2} & \cdots & S_{b+j_1+j_2} \\ \vdots & \vdots & \ddots & \vdots \\ S_{b+i_l+j_l} & S_{b+i_1+j_1+j_2+\cdots+j_l} & \cdots & S_{b+j_1+j_2+\cdots+j_l} \end{bmatrix}_{(p+1)\times (p+1)} \]

Then \( S = XYZ \), where
\[ X = \begin{bmatrix} X_1 & X_2 & \cdots & X_p \\ X_1^2 & X_2^2 & \cdots & X_p^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{p-1} & X_2^{p-1} & \cdots & X_p^{p-1} \end{bmatrix}_{(p+1)\times p} \]
and
\[ Y = \begin{bmatrix} Y_1 X_1^0 & Y_1 X_1^1 & \cdots & Y_1 X_1^{p-1} \\ 0 & Y_2 X_2^0 & \cdots & Y_2 X_2^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Y_p X_p^0 \\ 0 & 0 & \cdots & Y_p X_p^{p-1} \end{bmatrix}_{p+1 \times p} \]

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 \( \nu \leq i_1 \), the column rank of \( Z \) will be at most \( \nu \). Then the column rank of \( S \) will also be at most \( \nu \). Suppose the column rank of \( S \) is \( \lambda \leq \nu \). Then the first \( \lambda + 1 \) columns of \( S \) will be linearly dependent and there will exist \( f_1, f_2, \ldots, f_\lambda \) such that
\[ S_{b+i_1+j_1+j_2} = f_1 S_{b+i_1+j_1+j_2} + f_2 S_{b+i_1+j_1+j_2} + \cdots + f_\lambda S_{b+i_1+j_1+j_2}, \] (7)

and
\[ X_\mu^{i_1+j_1+j_2} + f_1 X_\mu^{i_1+j_1+j_2} + f_2 X_\mu^{i_1+j_1+j_2} + \cdots + f_\lambda X_\mu^{i_1+j_1+j_2} = 0, \]
for \( \mu = 1, 2, \ldots, p \).

Let
\[ f(z) = z^{i_1} + z^{i_1+1} + z^{i_2} + \cdots + z^{i_1+i_2} + f_{j_1} z^{i_1+j_1} + f_{j_2} z^{i_1+j_1+j_2} + \cdots + f_{j_l} z^{i_1+j_1+j_2+\cdots+j_l}. \]

Then
\[ f(X_\mu^\nu) = 0, \quad \text{for } \mu = 1, 2, \ldots, \nu. \] (7')

Now let \( s(z) = \prod_{\mu=1}^c (z - X_\mu) \). Apparently, we have \( \nu \leq i_1 \) and \( s(z) \) as a factor of \( f(z) \), namely, \( f(z) = h(z)s(z) \) for some \( h(z) \in \mathbb{GF}(q^m)[z] \). 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 \( C(x) = 1 + c_1 x + \cdots + c_k x^k \) be a solution obtained by the Fundamental Iterative Algorithm, then \( f_i = c_i \) for \( i = 1, 2, \ldots, \lambda \).

This is illustrated by the following example.

**Example 1:** Let us consider the (39,15) binary BCH code generated by \( g(x) = m(x)l_m(x) \). Let \( \alpha \) be a primitive element of \( \mathbb{GF}(2^{15}) \) and \( \beta = \alpha^{105} \), namely \( \beta \) is a 39th root of unity in \( \mathbb{GF}(2^{15}) \). 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 \( d \geq 7 \), 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 \( S_1, S_3, S_4, S_5, S_6, S_7, S_8 \). However, since \( S_6, S_9, S_{30}, S_{11}, S_{12} \) are known, we have \( S_{b+i_1+j_1+j_2} \) for \( b = 1, c_1 = 1, c_2 = 1, i = 0, 1, 2, 7, 8, j = 0, 1, 2, 3 \) and

\[ S = \begin{bmatrix} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \\ S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{4 \times 5} \]

Thus, \( S = XYZ \) and for \( \nu \leq 4 \)

\[ X = \begin{bmatrix} X_1 & X_2 & \cdots & X_p \\ X_1^2 & X_2^2 & \cdots & X_p^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{p-1} & X_2^{p-1} & \cdots & X_p^{p-1} \end{bmatrix}_{p \times p} \]
and
\[ Y = \begin{bmatrix} Y_1 X_1^0 & Y_1 X_1^1 & \cdots & Y_1 X_1^{p-1} \\ 0 & Y_2 X_2^0 & \cdots & Y_2 X_2^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Y_p X_p^0 \\ 0 & 0 & \cdots & Y_p X_p^{p-1} \end{bmatrix}_{p \times p} \]

Since \( X \) is a Vandermonde matrix, it is of rank \( \nu \), 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 \( f(z) \).

**Example 2:** Suppose \( S_1 = \alpha^{160} \) and \( S_3 = \alpha^{1800} \). Then we have

\[ S = \begin{bmatrix} \alpha^{1408} & \alpha^{2717} & \alpha^{1900} & \alpha^{3627} & \alpha^{3015} \\ \alpha^{1900} & \alpha^{1339} & \alpha^{3045} & \alpha^{2899} & \alpha^{1389} \\ \alpha^{1339} & \alpha^{3497} & \alpha^{2899} & \alpha^{1389} & \alpha^{1900} \\ \alpha^{1900} & \alpha^{1339} & \alpha^{3497} & \alpha^{2899} & \alpha^{1389} \\ \alpha^{1339} & \alpha^{3497} & \alpha^{2899} & \alpha^{1389} & \alpha^{1900} \end{bmatrix}_{4 \times 5} \]

For \( s = 1 \), we have \( C(x) = 1 \).
Then \( d_1 = \alpha^{3406} \neq 0 \). Since there is no \( d_{1,u} \in D \), where \( 1 \leq u < 1 \), \( d_{1,1} \Rightarrow D \),

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

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

Then

\[
d_{12} = \alpha^{2317}, \quad C^{(1),2}(x) = C^{(0,2)}(x) - \frac{d_{12}}{d_{11}} C^{(1)}(x) x^{2-1} = 1 + \alpha^{3406} x.
\]

Next \( d_{22} = \alpha^{3262} \neq 0 \). Since there is no \( d_{2,u} \in D \), where \( 1 \leq u < 2 \), \( d_{2,2} \Rightarrow D \),

\[
C^{(2)}(x) = C^{(0,2)}(x) = 1 \Rightarrow C.
\]

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

Then

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

Next

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

Then \( d_{33} = \alpha^{3375} \neq 0 \). Since there is no \( d_{3,u} \in D \), where \( 1 \leq u < 3 \), \( d_{3,3} \Rightarrow D \),

\[
C^{(3)}(x) = C^{(2,3)}(x) = 1 \Rightarrow C.
\]

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

Then

\[
d_{14} = \alpha^{42}, \quad C^{(1),4}(x) = C^{(0,4)}(x) - \frac{d_{14}}{d_{11}} C^{(1)}(x) x^{4-1} = 1 + \alpha^{3406} x + \alpha^{3051} x^2 + \alpha^{3723} x^3.
\]

Next

\[
d_{24} = \alpha^{3321}, \quad C^{(2),4}(x) = C^{(1,4)}(x) - \frac{d_{24}}{d_{22}} C^{(2)}(x) x^{4-2} = 1 + \alpha^{3406} x + \alpha^{3237} x^2 + \alpha^{3723} x^3.
\]

Then

\[
d_{34} = \alpha^{238}, \quad C^{(3),4}(x) = C^{(2,4)}(x) - \frac{d_{34}}{d_{33}} C^{(3)}(x) x^{3-1} = 1 + \alpha^{1060} x + \alpha^{3051} x^2 + \alpha^{2213} x^3.
\]

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

\[
C^{(4)}(x) = C^{(3,4)}(x) = 1 \Rightarrow C.
\]

For \( s = 5 \), we have \( C^{(0),5}(x) = C^{(4)}(x) = 1 + \alpha^{1060} x + \alpha^{3051} x^2 + \alpha^{2213} x^3 \).

Then

\[
d_{15} = \alpha^{1102}, \quad C^{(1),5}(x) = C^{(0,5)}(x) - \frac{d_{15}}{d_{11}} C^{(1)}(x) x^{5-1} = 1 + \alpha^{1060} x + \alpha^{3051} x^2 + \alpha^{2213} x^3 + \alpha^{1901} x^4.
\]

Next

\[
d_{25} = \alpha^{77}, \quad C^{(2),5}(x) = C^{(1,5)}(x) - \frac{d_{25}}{d_{22}} C^{(2)}(x) x^{5-2} = 1 + \alpha^{1060} x + \alpha^{3051} x^2 + \alpha^{3298} x^3 + \alpha^{1144} x^4.
\]
Example 2: Consider the (33,11) binary code generated by \( g(x) = m_1(x)m_2(x)m_3(x)m_4(x) \). For this code, \( d_q = 8 \), 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 \( GF(2^7) \) for \( S \), in order to decode five errors. On the other hand, \( d_{\text{root}} = 10 \) and \( d_{\text{red}} = 8 \), thus, the decoding method given in [4], [8] cannot decode five errors either.

Let \( a \) be a primitive element of \( GF(2^7) \) and \( \beta = a^{31} \), namely \( \beta \) is a 33rd root of unity. From the received vector we can calculate \( S_k \) for \( k = 1, 2, 4, 8, 16, 32, 31, 29, 25, 17, 3, 6, 12, 24, 15, 30, 27, 9, 18, 11, 22 \). To correct up to five errors, we let

\[
S = \begin{bmatrix}
S_{24} & S_1 & S_{11} & S_{15} & S_{25} & S_2 \\
S_1 & S_{11} & S_{21} & S_{25} & S_2 & S_{12} \\
S_{11} & S_{21} & S_{21} & S_8 & S_{12} & S_{32} \\
S_{21} & S_3 & S_8 & S_{12} & S_{22} & S_{32} \\
S_{31} & S_8 & D_{18} & D_{22} & S_{32} & S_9
\end{bmatrix}
\]

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

In the following this syndrome pattern is expressed by \( "b,b + i_1 c_1 ,\cdots ,b + i_4 c_4 ,0,j_1 c_1 ,\cdots ,j_4 c_4 "\). From the received vector, we can calculate \( S_k \) for \( k = 1, 2, 4, 8, 16, 32, 31, 29, 25, 17, 3, 6, 12, 24, 15, 30, 27, 9, 18, 11, 22 \). We can decode up to three errors using the following syndrome pattern,

\[
S = \begin{bmatrix}
S_{10} & S_{12} & S_6 & S_4 \\
S_{13} & S_9 & S_3 & S_1
\end{bmatrix} = \text{XYZ}
\]

where

\[
X = \begin{bmatrix}
1 & \cdots & 1 \\
X^{-3} & \cdots & X^{-4}
\end{bmatrix},
\]

\[
Y = \begin{bmatrix}
X_{16} & \cdots & 0 \\
0 & \cdots & 0 \\
0 & \cdots & X_{16}
\end{bmatrix},
\]

\[
Z = \begin{bmatrix}
X_{1}^{-4} X_{10}^{-10} X_{12}^{-12} \\
\vdots & \vdots & \vdots \\
1 & X_{1}^{-4} X_{10}^{-10} X_{12}^{-12}
\end{bmatrix}
\]

and \( 1 \leq \nu \leq 3 \).

Then \( b = 24 \), \( c_1 = c_2 = 1 \), \( i_1 = 2 \), \( i_2 = 5 \), \( i_3 = 6 \), and \( j_1 = 3 \), \( j_2 = 4 \). Since \( r(X) = 5 \) (See Lemma 3 in [17]), the columns 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 \( f_2, f_1, f_0 \). Then from the linear dependence relation we know that \( X_1 \) and \( X_2 \) must be roots of \( f_2 + f_1 X + f_0 = 0 \). 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 \( S_0, S_1, S_2, S_3, S_4 \), where \( S_0 = 0 \).

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 Case b).
<table>
<thead>
<tr>
<th>$n$</th>
<th>$k$</th>
<th>$d$</th>
<th>$t$</th>
<th>$d_0$</th>
<th>Syndrome Patterns</th>
<th>Decoding Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>1**</td>
<td>$A(F(1,8))$</td>
</tr>
<tr>
<td>21</td>
<td>9</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>0,1,3,7</td>
<td>$A(1,2,6,8)$</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1,3,7,9</td>
<td>1,2,6,7 and 0,1,2</td>
<td>$A(7,11)$ → 15; $F(1-6)$, Exa. 3</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>0,1,3,7,9</td>
<td>1,2,6,7 and 0,1,2</td>
<td>$A(7,12)$ → 13; $F(1-6)$</td>
</tr>
<tr>
<td>23</td>
<td>12</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>1**</td>
<td>$A(1,8,16)$ → 5; $F(1-6)$, Exa. 4</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>0,1**</td>
<td>12,8,20 and 4,5,0</td>
<td>$A(1,16)$ → 5; $F(1-6)$</td>
</tr>
<tr>
<td>31</td>
<td>21</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>1,5</td>
<td>4,8,9 and 0,1</td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>1,7</td>
<td>4,16,1 and 0,3</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0,1,5</td>
<td>0,1,2 and 0,8</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1,7</td>
<td>2,7,10</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>1,5,7</td>
<td>0,7,18 and 0,1,2</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>0,1,5,7</td>
<td>0,7,8,18 and 0,1,2</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>11</td>
<td>5</td>
<td>8</td>
<td>1,3,11</td>
<td>1,2,7,9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>5</td>
<td>10</td>
<td>0,1,3,11</td>
<td>12,17,22,29,1</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>16</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>1,5</td>
<td>1,2,4,5 and 0,3,6</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>0,1,5,7</td>
<td>1,2,4,5 and 0,3,6</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1,5,7,15</td>
<td>1,2,4,5 and 0,3,6</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>0,1,5,7,15</td>
<td>1,2,4,5 and 0,3,6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>6</td>
<td>12</td>
<td>0,1,5,5**</td>
<td>31,32,33,34,0,1,8</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>26</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0,1**</td>
<td>$A(10,11)$ → 28; $F(37-2)$</td>
</tr>
<tr>
<td>24</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0,1</td>
<td>1,1,2 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>4</td>
<td>7</td>
<td>1,3**</td>
<td>2,3,8,9 and 0,1,2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>4</td>
<td>8</td>
<td>0,1**</td>
<td>2,3,8,9 and 0,1,2</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>29</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>1**</td>
<td>$A(F(1,21))$</td>
</tr>
<tr>
<td>45</td>
<td>23</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>1,5,21</td>
<td>31,32,33,38 and 0,1,2</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>0,1,5,21</td>
<td>1,9,19,21 → 26 → 41 → 43 → 44; $F(38-44)$</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>0,1,5,7</td>
<td>43,44,0,1,11 and 0,1,2,3</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>1,5,7,15</td>
<td>5,12,33,14,18,28 and 0,1,2,3</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>1,7,9,15</td>
<td>13,14,15,16,17 and 10,12,13,14,28 and 0,1,2,3</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>0,1,7,9,15</td>
<td>$A(F(13-16))$</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>0,1,7,9,15</td>
<td>$A(F(13-16))$</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>0,1,7,9,15</td>
<td>$A(F(11-14))$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>5</td>
<td>9</td>
<td>1,5,7,9,15</td>
<td>$A(F(13-16))$</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>8</td>
<td>12</td>
<td>5</td>
<td>9</td>
<td>0,1,5,7,9,15</td>
<td>$A(F(13-16))$</td>
</tr>
<tr>
<td>35</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>0,1</td>
<td>1,8,15 and 0,1</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0,1</td>
<td>1,8,15 and 0,1</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0,1</td>
<td>1,8,15 and 0,1</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0,1,17</td>
<td>1,8,15 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>0,1,17</td>
<td>1,8,15 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1,3,9</td>
<td>1,8,15 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1,3,9</td>
<td>1,8,15 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1,3,9</td>
<td>1,8,15 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1,3,9</td>
<td>1,8,15 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>1,3,9</td>
<td>1,8,15 and 0,1,4</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>12</td>
<td>5</td>
<td>6</td>
<td>1,3,9,17</td>
<td>32,47,48,49,50 and 0,1,2,3,4</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>12</td>
<td>5</td>
<td>10</td>
<td>0,1,3,9,17,19</td>
<td>$A(F(13-16))$</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>35</td>
<td>2</td>
<td>4</td>
<td>1**</td>
<td>7,8,9 and 0,9</td>
<td>$A(F(7,8))$</td>
</tr>
</tbody>
</table>

** Indicates that the code is a BCH code.

$A$ Indicates the code is a BCH code.

$A'$ Indicates that the code can be decoded by the indicated syndrome pattern following the procedure in Section III.

$d$ Actual minimum distance.

$d_0$ Maximum number of errors correctable.

$d_0$ BCH bound.
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 relation are $f_1, f_2, f_3, f_4$ and $f_5$. From the linear dependence relation we know that $X_{k+1}^{-1}X_k^{-1}X_k^{-1}$ must be roots of $f_1 + f_2 z^{-1} + f_3 z = 0$. This equation has at most six 23rd roots of unity in $\mathbb{GF}(2^3)$. If the number of roots is 5, then follow the same procedure as in Case c). If the number of roots is 6, then $f_5 \neq 0$, and we can find $S_6$ using the relation $f_6 S_6 + f_5 S_5 + f_4 S_4 + f_3 S_3 + f_2 S_2 + f_1 S_1 + S_0 = 0$. Using the Forney formula and $S_1, S_2, S_3, S_4, S_5, S_6$ we can find $S_7, S_8, S_9, S_{10}$, and $S_{11}$. Using the following syndrome pattern and the previous decoding procedure, we can decode up to $6$ errors. Let $S = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \\ S_{11} \end{bmatrix}$, namely, $S$ is $(31, 32, 33, 34, 0, 1, 0, 1, 2, 3, 4, 5)$. Then $b = 31, c_1 = c_2 = 1, i_1 = 1, i_2 = 2, j_1 = 3, i_3 = 4, i_4 = 5, i_{12} = 12, j_1 = 1, j_2 = 2, j_3 = 3, j_4 = 4, j_{12} = 5$. 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 $2^5$ times to correct six errors, and [15] did not discuss the decoding of this code. Since $i_{12} = 12, \delta$ may be 12. If $\delta = 12$, we first determine $S_7$ from $f_0 S_7 + f_6 S_6 + f_5 S_5 + f_4 S_4 + f_3 S_3 + f_2 S_2 + f_1 S_1 + S_0 = 0$, where $f_0 \neq 0 (8, 9, 10, 11, 12, 19 \rightarrow 7)$. Then we can determine the six error locations and magnitudes ($F(1-123)$).

Example 6: We consider a longer binary code, the (99, 43) cyclic code with $n = 99$, $(g(x) = m_2(x)m_2(x)m_3(x)m_3(x)m_1(x), d = 11, d_{\text{Roos}} = 9, d_{\text{BCH}} = 8, d_{\text{BCH}} = 7$. We have the syndromes $S_k$ for $k = 3, 5, 6, 12, 24, 48, 96, 93, 87, 75, 51, 5, 10, 20, 40, 60, 23, 46, 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: $(9, 10, 43, 44, 51, 52, 0, 1, 2, 3, 4)$, namely, $S = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \end{bmatrix}$. Since $i_3 = 43, \delta$ may be 43. When $\delta = 43$, we have $38, 72, 73, 80, 81 \rightarrow 39$ and $38, 71, 72, 79, 80 \rightarrow 37$. Then, by using the properties about conjugates and $S_{30}, S_{31}$, we can find $S_3$ for $k = 39, 78, 57, 15, 30, 21, 42, 84, 69$ 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, 94, 29, 58, 17, 34, 68. Using the Forney formula on $S_i = S_8$ ($F(1-43)$), we can determine the error locations and the magnitudes. Relevant information for decoding many cyclic and BCH codes of length up to 55 is 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

On Indecomposable Abelian Codes and Their Vertices

Karl-Heinz Zimmermann

Abstract—Indecomposable non-semisimple 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 \( q = p^e \) 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 \((n_1n_2,k_1k_2)\)-code with minimal distance \( d = d_1d_2 \).

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:

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

\[
k \left( \sum_{g \in G} l_g g \right) = \sum_{g \in G} (kl_g) g.
\]

Moreover, multiplication in \( G \) induces multiplication in \( FG \) as follows:

\[
\left( \sum_{g \in G} k_g g \right) \left( \sum_{h \in G} l_h h \right) = \sum_{w \in G} \left( \sum_{g \in G} \sum_{h \in G} (k_g l_h) g \right) h.
\]

Hence, \( FG \) is an associative \( F \)-algebra with identity \( 1 = 1_F \) 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 \( M \) be a finite group of order \( n \). Each right ideal \( M \) of the group algebra \( FG \) is 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 called cyclic or Abelian, 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 supp \((a) = \{ g \in G | a_g \neq 0 \} \) denote the support of \( a \). The number \( \text{supp}(a) = \sum_{g \in G} \text{supp}(a) \) is called the weight (w.r.t. \( F \)-basis \( G \)). The minimal distance of an \( FG \)-code \( M \) (w.r.t. \( F \)-basis \( G \)) is given by

\[
\text{dist}(M) = \frac{\text{dist}(M)}{\text{dist}(M)} = \frac{\text{dist}(M)}{\text{dist}(M)} = \frac{\text{dist}(M)}{\text{dist}(M)}.
\]

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

Proposition 1: Let \( H \) be a subgroup of \( G \), and let \( \rho \) 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 \( \rho \).

An \( FG \)-code \( M \) is called semisimple if the radical \( \text{Rad}M \) 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 \( G = \langle g \rangle \) be a cyclic group of order \( n \).

1) If \( p \mid n \), 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 \( \bar{F} \) for \( G \), i.e., a finite extension field of \( F \) containing the \( n \)th roots of unity.