ARMY RESEARCH LABORATORY

Toward a New Method of Decoding Algebraic Codes Using Gröbner Bases

A. Brinton Cooper III

ARL-TR-293

October 1993

NOTICES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
October 1993

3. REPORT TYPE AND DATES COVERED
Final, October 1991 to September 1992

4. TITLE AND SUBTITLE
Toward a New Method of Decoding Algebraic Codes Using Gröbner Bases

5. FUNDING NUMBERS
PR: 1L161102AH43

6. AUTHOR(S)
A. Brinton Cooper III

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-CI-CC
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-OP-CI-B (Tech Lib)
Aberdeen Proving Ground, MD 21005-5066

10. SPONSORING MONITORING AGENCY REPORT NUMBER
ARL-TR-293

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

A binary BCH error control code is a vector subspace of binary N-tuples. Algebraically, the code is generated by a polynomial having binary coefficients and roots in $GF(2^m)$. It is decoded by computing a set of syndrome equations which are multivariate polynomials over $GF(2^m)$ and which exhibit a certain symmetry. If the number of transmission errors in a received word does not exceed a bound t for the code, the roots of the syndromes are the locations, in the received word, of those errors. These multivariate polynomials are taken as the basis for an ideal in the ring of polynomials in t variables over $GF(2^m)$. A celebrated algorithm by Buchberger produces a reduced Gröbner basis of that ideal. It turns out that, since the common roots of all the polynomials in the ideal are a set of isolated points, this reduced Gröbner basis is in triangular form, and the univariate polynomial in that basis is the well known BCH error locator polynomial, the roots of which specify the error locations. Decoding is algorithmically complete when this polynomial is known.

14. SUBJECT TERMS
decoding, algebraic functions, polynomials

15. NUMBER OF PAGES
22

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

Acknowledgements

The author expresses special thanks to Professor Peter Stiller of Texas A&M University for many discussions on Gröbner bases, the theory of equations, and topics in the theory of finite fields; to Richard Kaste and Edwin Davisson of the US Army Research Laboratory for reading many preliminary versions of this paper and for providing enlightening discussions of the mathematics; to Professor Charles Retter, recently of Northeastern University, for expressing his skeptical views on this work; to Professor P.G. Farrell of Manchester University, UK, for his unwavering encouragement to follow through with these ideas; and to Professor Stephen Wicker of the Georgia Institute of Technology for reading this work and making many thoughtful suggestions for its improvement. Notwithstanding these contributions, the shortcomings remain mine.

INTENTIONALLY LEFT BLANK.

Contents

INTENTIONALLY LEFT BLANK.

1 Introduction

Modern algebraic techniques have been used to design and decode codes for error control as far back as the early 1960s when the binary BCH[1] codes [2-5] were discovered independently by Bose and Chaudhuri and by Hocquenghem. The BCH codes and their descendents are popular for several reasons, including their regular algebraic structure which permits easy encoding using simple shift registers and the existence of codes for a wide range of block lengths and error correction capabilities.

However, the asymptotic performance of BCH codes is not "good" [5] in that the error probability after decoding and the information rate of the code are not simultaneously bounded away from zero with increasing block length. Nevertheless, the BCH codes and their derivatives are widely used because they are easy to generate, well understood, and useful in the control of transmission errors over noisy channels. BCH decoders, however, are complex, and work continues to find simpler and more powerful decoders.

This work applies recent results from the algebra of multivariate polynomials to the direct solution of the syndrome equations of binary BCH codes. In this problem, some number t of nonlinear polynomial equations must be solved for the locations of the errors.

Following a review of the basic theory of linear block codes, Section 2 presents the polynomial model of cyclic codes and shows how a BCH code is specified solely by a set of roots of its generator polynomial. Section 3 reviews popular methods for decoding BCH codes. Although the Berlekamp-Massey Algorithm (BMA) [6,7] is probably the most widely discussed in the literature, we present Peterson's algorithm [2] because it is simpler than BMA and provides the paradigm for BMA as well as other decoders. Section 4 casts the problem into ideals in the ring of multivariate polynomials over GF(2^m). Such ideals are defined by the roots of the member polynomials. Modern methods are used to solve these equations directly.

Examples are included.

2 Linear Block Codes

2.1 Error Control

A common method for controlling errors in information transmitted over noisy channels is the use of *linear block codes* (LBC)[2]. Algebraically, a LBC is a k-dimensional vector subspace of a vector space of n-tuples over a finite field and, therefore, has a *basis* which spans the code. Methods from linear algebra can be used to express and manipulate the *generator matrix*, the rows of which are the basis of the code. The dimension k of the LBC is smaller than n, the number of elements or *symbols* in the n-tuple. This gives rise to the existence of $n-k$ *redundant* symbols in each codeword. This redundancy introduces *distance* between pairs of codewords.

The sense in which we define "nearness" is Hamming distance.

[1] Bose-Chaudhuri-Hocquenghem. McEliece [1] presents an interesting history of the naming of these codes.

[2] For a thorough coverage of this topic, the reader is referred to any of several excellent texts [1-6,8,9].

Definition: The Hamming distance d_H between two n-tuples is the number of places in which they differ.

Thus, if $w_1 = (1001110)$ and $w_2 = (1011010)$, $d_H(w_1, w_2) = 2$ as can be verified by inspection.

Channel noise often increases the probability that the received word will be closer to a code word other than that which was transmitted. Sufficient code redundancy, however, can usually provide sufficient distance between all pairs of codewords that the codeword which was transmitted can be identified correctly in a large fraction of cases, even when noise has moved it closer to another codeword.

Let k be the number of bits of information represented by one codeword, and let n be the codeword length. An information block is represented as follows:

$$I = (i_1, \ldots, i_k), \; i_j \in \mathrm{GF}(2), \; j = 1, \ldots, k, \; k < n. \tag{1}$$

Let G be a $k \times n$ generator matrix, the rows of which span the LBC. Then every block I of k information bits generates a distinct codeword V:

$$V = IG. \tag{2}$$

A convenient model of the channel represents noise as a set of n Bernoulli trials [10] in which the "probability of success" is taken as the probability p of an error in any binary symbol. This means that the n-tuple V_R received at the noisy channel output can be modeled as the modulo 2 vector sum of the transmitted codeword V and an error vector $E = (e_1, \ldots, e_n)$ where $e_j = 1$ if an error occurred in the j^{th} position and 0 otherwise.

$$V_R = V + E \tag{3}$$

The decoding problem is: given V_R, find V.

2.2 Polynomials and Cyclic Codes

Using powers of an indeterminate x as placeholders permits writing a polynomial model of the LBC. This is more than formalism, however, as it permits code construction and decoding based upon well-known principles of algebra.

Information can be carried in the (binary) coefficients of a polynomial $i(x)$:

$$i(x) = i_0 + i_1 x + \cdots + i_{k-1} x^{k-1}, \; i_j \in \mathrm{GF}(2), \; j = 0, 1, \ldots, k-1. \tag{4}$$

Codeword polynomials are generated by multiplying $i(x)$ by a *generator polynomial* $g(x)$ of degree $n-k$:

$$g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k}, \; g_j \in \mathrm{GF}(2), \; j = 0, 1, \ldots, n-k. \tag{5}$$

Coefficients of the resulting polynomial $v(x)$ represent the binary symbols in the codeword:

$$\begin{aligned} v(x) &= i(x)g(x) \\ &= v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}, \; v_j \in \mathrm{GF}(2), \; j = 0, 1, \ldots, n-1 \end{aligned} \tag{6}$$

Here, the code redundancy is introduced in the process of multiplication by $g(x)$ which results in the representation of k binary information symbols by $n > k$ binary code symbols. Previous notions of distance and error correction, therefore, hold here as well.

A code is said to be *cyclic* if every cyclic shift of every codeword is also a codeword. Algebraically, a code is cyclic whenever $g(x)|x^n - 1$, and the codeword length n is the smallest integer for which $g(x)|x^n - 1$ [5].

2.3 BCH Codes

The BCH codes provide a convenient paradigm for several families of powerful LBC's including Reed-Solomon [1-6,8,9] and Goppa [1] codes. A binary, primitive BCH code is a cyclic code of length $n = 2^m - 1$. Its generator polynomial numbers among its roots $2t$ consecutive powers[3] of a primitive element α of the locator field GF(2^m). With correct decoding, this code can correct up to t channel errors in every codeword.[4]

Example: Let $m = 4$ and $t = 2$. Then $n = 15$ and the roots of $g(x)$ include α, α^2, α^3, and α^4. Because $\alpha^{15} = 1$, these must also be roots of $g(x)$: $\{\alpha^8, \alpha^6, \alpha^{12}, \alpha^9\}$. Hence, the degree of $g(x)$ is $n - k = 8$ so that the dimension k of the code is 7. (*i.e.*, the code has $2^7 = 128$ code words.) The code is capable of correcting at least $t = 2$ errors in every codeword, and the code rate, k/n is 0.47 information bits per binary symbol transmitted.

3 BCH Decoding

Of course correcting t errors in a codeword of length n implies a decoding procedure that achieves this error correcting potential. A trivial but completely correct decoding technique is to construct a table of every binary n-tuple and the codeword into which it is to be decoded. For a channel imposing independent errors on the symbols of a codeword, the rule for constructing this table is to decode an n-tuple into the nearest codeword[5].

However, table lookup decoding is feasible only for rather small codes. The power of modern computers is quickly exhausted for codeword lengths of several thousand bits and hundreds of errors per word. Therefore, we continue to search for algorithmic, algebraic decoders which are much faster and demand much less storage. Many algebraic decoders will correct every error pattern of t or fewer errors but no more, even though the code may correct some patterns of more than t errors. Nevertheless, the number of such error patterns is usually sufficiently small that it does not affect the overall decoding error probability significantly.

[3]The nonzero powers $\alpha^0, \alpha^1, \ldots, \alpha^{2^m-1}$ of a primitive element of GF(2^m) are the distinct nonzero elements of that field.

[4]In order that the codewords be binary, it is necessary, for every root β^i of $g(x)$, that all conjugates $\{\beta^{2^i}, \beta^{4^i}, \ldots\}$ be roots of $g(x)$ as well [6] .

[5]Because this is a *minimum distance decoding* technique, no other decoder can correct more errors on a memoryless channel.

3.1 Peterson's Decoder

Let $r(x)$ represent the received vector when t−error correcting BCH codeword $v(x)$ is transmitted over a channel corrupted by additive noise:

$$r(x) = v(x) + e(x). \tag{7}$$

$e(x)$ is the error polynomial: $e_j = 1$ if an error occurred in the j^{th} position and 0 otherwise. The paradigm for many useful decoders of this code is Peterson's decoder [2], which implements a four-step decoding procedure:

- calculate *syndromes*, functions of the coeffcients of $r(x)$;
- calculate coefficients of the *error locator polynomial*;
- solve the error locator polynomial for the locations, in the received word, of the errors; and
- (for nonbinary codes) calculate the error values.

3.1.1 The Syndromes

Consider the channel output, $r(x)$ as given by (7). The $2t$ syndrome values are obtained by substituting the $2t$ consecutive roots of the generator polynomial into the received polynomial:

$$S_j = r(\alpha^j) = g(\alpha^j) + e(\alpha^j) = e(\alpha^j),\ j = 1, \ldots, 2t. \tag{8}$$

Writing only those coefficients e_j which are not zero leads to the following form of the $2t$ syndrome equations:

$$\begin{aligned} e_{i_1}\alpha^{i_1} + e_{i_2}\alpha^{i_2} + \cdots + e_{i_t}\alpha^{i_t} &= S_1 \\ e_{i_1}\alpha^{2i_1} + e_{i_2}\alpha^{2i_2} + \cdots + e_{i_t}\alpha^{2i_t} &= S_2 \\ &\vdots \\ e_{i_1}\alpha^{2ti_1} + e_{i_2}\alpha^{2ti_2} + \cdots + e_{i_t}\alpha^{2ti_t} &= S_t. \end{aligned} \tag{9}$$

Note the following:

(a) The indices $\{i_1, i_2, \ldots\}$ in (9) are the coordinates of the nonzero elements (and hence, of the errors) in the error vector. It is convenient, therefore, to write $X_j = \alpha^{i_j}$. The values of the α^{i_j} are called the *error locators* of the received word.

(b) In any field GF(2^m) of characteristic two, $(a+b)^2 = a^2 + b^2$ [11]. Therefore, in (9), every syndrome computed from even powers of α is an even power of some syndrome computed from odd powers of α; *e.g.*, $S_2 = S_1^2$. These are redundant and do not contribute to solving for the error locators.

(c) In (9), $e_{i_j} = 1$, $j = 1, \ldots, 2t$ and need not be explicitly written. The syndromes $\{S_j,\ j = 1, \ldots, 2t\}$ are known (computed) elements of GF(2^m) and can be expressed as powers of α; *i.e.*, $S_\sigma = \alpha^{j_\sigma}$.

Considering (a), (b), and (c) with (9) gives a system of t polynomial equations, the solutions to which are the error locators of the received word:

$$\begin{aligned} S_1 &= \alpha^{j_1} = X_1 + X_2 + \cdots + X_t \\ S_3 &= \alpha^{j_3} = X_1{}^3 + X_2{}^3 + \cdots + X_t{}^3 \\ &\vdots \\ S_{2t-1} &= \alpha^{j_{2t-1}} = X_1{}^{2t-1} + X_2{}^{2t-1} + \cdots + X_t{}^{2t-1}. \end{aligned} \tag{10}$$

3.1.2 The Error Locator Polynomial

Derivation of (10) [6] assumed that no more than t errors occured in a block of length n. An *error locator polynomial* is derived from these functions.

Definition: The error locator polynomial $\sigma(x)$ is the (univariate) polynomial, all the roots of which indicate the locations of errors in a received word:

$$\begin{aligned} \sigma(x) &= \prod_{i=1}^{t}(x - X_i) \\ &= x^t + \sigma_1 x^{t-1} + \sigma_2 x^{t-2} + \cdots + \sigma_t. \end{aligned} \tag{11}$$

It is easy to see that the coefficients are functions of the elementary symmetric functions of the roots (ie error locators):

$$\sigma_1 = \sum_i X_i \tag{12}$$

$$\sigma_2 = \sum_{i<j} X_i X_j$$

$$\sigma_3 = \sum_{i<j<k} X_i X_j X_k \tag{13}$$

$$\vdots \tag{14}$$

$$\sigma_t = X_1 X_2 \ldots X_t.$$

Since $\sigma(x)$ is satisfied by the error locators, (11) becomes

$$X_i^t + \sigma_1 X_i^{t-1} + \sigma_2 X_i^{t-2} + \cdots + \sigma_t = 0. \tag{15}$$

Peterson's method uses the syndrome relations to construct a set of linear equations in the $\{\sigma_i\}$. This set can be solved for these coefficients. Multiplying (15) by X_i^j for any j gives

$$X_i^{t+j} + \sigma_1 X_i^{t+j-1} + \sigma_2 X_i^{t+j-2} + \cdots + \sigma_t X_i^j = 0. \tag{16}$$

[6] The reader should recognize these as a set of power-sum symmetric functions [11].

Summing over i and substituting $S_j = \sum_{i=1}^{t} X_i^j$ gives

$$S_{t+j} + \sigma_1 S_{t+j-1} + \sigma_2 S_{t+j-2} + \cdots + \sigma_t S_j = 0. \tag{17}$$

These *Newton's Identities* [11] generate linear systems of equations for the $\{\sigma_j\}$, one system for each value of t. For $t = 1$,

$$S_2 + \sigma_1 S_1 = 0,$$

and for $t = 2$,

$$S_3 + S_2\sigma_1 + S_1\sigma_2 + = 0.$$

These are recursively solved for the coefficients, yielding (15) explicitly.

3.1.3 Solving the Error Locator Polynomial

Decoding is complete when the roots of $\sigma(x)$ are found and the necessary corrections made to $r(x)$. The *Chien search* [8] is a method for doing this without explicitly solving $\sigma(x)$. This method uses a digital circuit which evaluates $\sigma(x)$ at each member α^j of GF(2^m) and sets a *correction bit* to unity if $\sigma(x)$ is satisfied. The received polynomial $r(x)$ is clocked through the circuit and the correction bit is added modulo 2 at the appropriate location. Therefore, whenever a root of $\sigma(x)$ is found, the appropriate received symbol is complemented.

The Chien search will be required in implementing the direct solution methods discussed below.

3.2 Related Methods of Finding the Error Locator Polynomial

For more than approximately six errors per codeword, Peterson's method requires a number of finite field multiplications which grows with the square of t. Berlekamp [6] produced an iterative method for finding the coefficients that grows only linearly with t; Massey [7] improved Berlekamp's method (producing the BMA), showing that it is equivalent to synthesizing the shortest linear feedback shift register that can generate the sequence of syndrome values. The methods are similar and can be studied in the references.

4 Direct Solution Techniques

The objective is to find a solution set to (10):

$$\begin{aligned}
\alpha^{j_1} &= X_1 + X_2 + \cdots + X_t \\
\alpha^{j_3} &= {X_1}^3 + {X_2}^3 + \cdots + {X_t}^3 \\
&\vdots \\
\alpha^{j_{2t-1}} &= {X_1}^{2t-1} + {X_2}^{2t-1} + \cdots + {X_t}^{2t-1}
\end{aligned} \tag{18}$$

where α is a primitive element in GF(2^m). Assume that the number of errors in a received word does not exceed t[7]. Then (18) is a system F of t independent equations with at most t solutions. Hence, F is a system of t polynomials in t unknowns and has one unique solution, $\beta = (\beta_1, \ldots, \beta_t)$[8].

4.1 Rings and Ideals

Direct solution techniques of (18) attempt to exploit the rich algebraic structure of the *ring* $R = K[\mathbf{X}] = K[X_1, X_2, \ldots, X_t]$ of polynomials in t variables over K = GF(2^m) [11]. A subset $\mathcal{I}$ of a ring is called an *ideal* if it is a subgroup of the additive group of the ring and if, for every $i \in \mathcal{I}$ and every $r \in R$, both ir and ri belong to $\mathcal{I}$. Hilbert's Basis Theorem [12] requires that every ideal in $K[\mathbf{X}]$ have a finite basis.

Consider F to be a subset of the ring $K[\mathbf{X}]$. The set $\mathcal{I}(F)$ spanned by members of F (where coefficients are taken from K[**X**]) is an ideal in $K[\mathbf{X}]$:

$$\mathcal{I}(F) \triangleq (F) \subset K[\mathbf{X}]. \tag{19}$$

The common zeros of the polynomials of F are said to form an *algebraic manifold*, [12] which is "defined by" those polynomials. Thus, all points of the manifold satisfy every polynomial in $\mathcal{I}(F)$. Direct solution techniques involve searching $\mathcal{I}(F)$ for another set G of polynomials which span $\mathcal{I}(F)$ and which are simpler to solve than those in F. Hence, new methods for finding bases of ideals in $K[\mathbf{X}]$ bear on the decoding problem.

4.2 A Basis for $\mathcal{I}(F)$

The objective now is to find for $\mathcal{I}(F)$ a basis G which is "easily" solved for the underlying roots.

The basis G is obtained from the defining polynomial set F by applying transformations which do not eliminate any roots of the system. An example illustrates the transformations:

Example: Suppose set F is:

$$\begin{aligned} f_1 : \; X_1 + X_2 + \alpha^j &= 0 \\ f_2 : \; X_1^3 + X_2^3 + \alpha^k &= 0, \end{aligned} \tag{20}$$

and suppose that it is known that this system has the solution $(\beta_1, \beta_2) \in \text{GF}(2^m)^2$. Then

$$y(\mathbf{X}) = a_1(\mathbf{X}) f_1(\mathbf{X}) + a_2(\mathbf{X}) f_2(\mathbf{X}) \tag{21}$$

is satisfied by (β_1, β_2) as well[9].

[7] If the number of errors exceeds t, such a decoder is likely to exhibit *decoding failure*. That is, it may return an incorrect result.

[8] Actually, the rigorously correct statement is that all zeros of the system are "equivalent" and "mapped on one another by an isomorphism which leaves fixed the elements of the ground field..." [12]

[9] Of course, if $a_1(\mathbf{X})$ and $a_2(\mathbf{X})$ have a common factor, $y(\mathbf{X})$ may have an additional root that does not satisfy f_1 or f_2, but this case is of no interest.

If $a_2(\mathbf{X}) = 1$ and

$$a_1(\mathbf{X}) = X_1^2 + X_1(X_2 + \alpha^j) + (X_2 + \alpha^j)^2, \tag{22}$$

then

$$y(\mathbf{X}) = X_2^2\alpha^j + X_2\alpha^{2j} + \alpha^{2j} + \alpha^k, \tag{23}$$

and this system has been *reduced* from two equations (a cubic and a linear) to a single, univariate second degree equation having the same solution (β_1, β_2) as the original system. We say that the cubic has been *reduced modulo* F to $y(\mathbf{X})$.

The algorithm for deriving the desired ideal basis G is based upon such reduction operations and produces a *reduced Gröbner basis* [13] of the ideal spanned by F. A reduced Gröbner G basis is a basis of the ideal, each member of which has coefficient of highest order term = 1 and no element of which can be reduced modulo G. It is known [13] that a reduced Gröbner basis for $\mathcal{I}(F)$ can be written in *triangularized* form:

$$\begin{aligned} g_1 &= g_1(X_1) \\ g_2 &= g_2(X_1, X_2) \\ &\vdots \\ g_t &= g_t(X_1, X_2, \ldots, X_t). \end{aligned} \tag{24}$$

This form suggests a recursive root finding technique. However, the following lemma forms the bases for our direct method of finding the BCH error locator polynomial [14].

Lemma 1 *$g_1(x_1)$ is, within a multiplicative constant, the error locator polynomial $\sigma(x)$ of the BCH code.*

Proof: Every element of I has among its roots the set $\{\beta_i\}$ of roots which defines the original spanning set F. Reducing F to G neither adds nor subtracts roots to/from any polynomial. Therefore, $\sigma(x)$ and $g_1(x)$ are products of the same factors $\{(x_i - \beta_i)\}$ and, hence, differ by no more than a n.ultiplicative constant. *q.e.d.*

4.3 Gröbner Bases as a Basis for Decoding

Descriptions of the general form of Buchberger's algorithm for finding the Gröbner basis of an ideal $\mathcal{I}(F)$ run for many pages [13]. We include a succinct tutorial exposition of the algorithm in the Appendix. The example below illustrates the use of the algorithm:

Example: This is a general form of the problem. Taking K to be GF(2^4) and $t = 3$ results in a 3-error correcting code with block length $n = 2^4 - 1$, dimension $k = 5$, and 32 code words. In general, the decoder produces these non-redundant syndromes:

$$\begin{aligned} X_1 + X_2 + X_3 + \alpha^i &= 0 \\ X_1^3 + X_2^3 + X_3^3 + \alpha^j &= 0 \\ X_1^5 + X_2^5 + X_3^5 + \alpha^k &= 0. \end{aligned} \tag{25}$$

Define three intermediate polynomials,

$$\begin{aligned}
p_1(\mathbf{X}) &= \sum_{j=0}^{2} X_3^j (X_2 + X_1 + a^i)^{2-j} \\
p_2(\mathbf{X}) &= \sum_{j=0}^{4} X_3^j (X_2 + X_1 + a^i)^{4-j} \\
p_3(\mathbf{X}) &= X_2^2 + X_2 X_1 + X_2 a^i + X_1^2 + X_1 a^i + a^{2i},
\end{aligned} \tag{26}$$

and from these produce three "coefficient" polynomials:

$$\begin{aligned}
a_1(\mathbf{X}) &= p_1 p_3 (X_1 + a^i) + p_2 (X_1 + a^i) + p_1 (a^j + a^{3i}) \\
a_2(\mathbf{X}) &= p_3 (X_1 + a^i) + a^j + a^{3i} \\
a_3(\mathbf{X}) &= X_1 + a^i.
\end{aligned} \tag{27}$$

Substitute the p_i into the a_j to get

$$\begin{aligned}
a_1(\mathbf{X}) = {} & X_1 X_3^4 + a^i X_3^4 + X_1 X_2 X_3^3 + a^i X_2 X_3^3 + X_1^2 X_3^3 + a^{2i} X_3^3 + X_1^2 X_2 X_3^2 \\
& + a^{2i} X_2 X_3^2 + a^i X_1^2 X_3^2 + a^{2i} X_1 X_3^2 + a^j X_3^2 + a^{3i} X_3^2 + X_1^2 X_2^2 X_3 + a^{2i} X_2^2 X_3 \\
& + X_1^3 X_2 X_3 + a^j X_2 X_3 + a^i X_1^3 X_3 + a^j X_1 X_3 + a^{j+i} X_3 + a^{4i} X_3 + X_1^2 X_2^3 \\
& + a^{2i} X_2^3 + a^i X_1^2 X_2^2 + a^{2i} X_1 X_2^2 + a^j X_2^2 + a^{3i} X_2^2 + X_1^4 X_2 + a^{4i} X_2 \\
& + a^i X_1^4 + a^{2i} X_1^3 + a^j X_1^2 + a^{4i} X_1 + a^{j+2i} + a^{5i} \\
a_2(\mathbf{X}) = {} & X_1 X_2^2 + a^i X_2^2 + X_1^2 X_2 + a^{2i} X_2 + X_1^3 + a^j.
\end{aligned}$$

This yields a univariate polynomial which we recognize as the error locator polynomial:

$$\begin{aligned}
\sigma(X_3) &= \sum_{\nu=1}^{3} a_\nu(\mathbf{X}) f_\nu(\mathbf{X}) \\
&= X_3^3(\alpha^j + \alpha^{3i}) + X_3^2(\alpha^{i+j} + \alpha^{4i}) + X_3(\alpha^k + \alpha^{2i+j}) \\
&\quad + \alpha^{i+k} + \alpha^{2j} + \alpha^{3i+j} + \alpha^{6i}.
\end{aligned} \tag{28}$$

Finding $\sigma(X)$ solves the decoding problem.

5 Conclusion

Mathematically, we have shown a decoder that computes a set of syndrome values which are functions of the roots of the code's generator polynomial and of the error locations. These syndromes are the constant terms of a system of nonlinear polynomials. We have presented a method for extracting from that system the error locator polynomial, one which is satisfied by the error locations expressed as elements of $GF(2^m)$. The coefficients of the error locator polynomial are functions of the syndrome values only. Thus, the decoder need do only two things: compute syndromes and coefficients.

This class of decoder is interesting because of the promise of noniterative decoding of BCH and BCH-like codes.[10] Of course, an efficient version of Buchberger's algorithm, tailored to systems

[10] Of special, near-term importance to system designers is the possibility of improved decoders for Reed-Solomon codes, powerful codes already used in many high performance systems.

of equations such as (10), is required but not yet in hand. Once this hurdle is overcome, we envision several possibilities. One is to incorporate a version of Buchberger's algorithm into a decoder. Another is to solve Buchberger's algorithm for a large family of codes, expressing the error locator polynomials in terms of the syndrome values alone. These could easily be programmed into hardware to produce a fast decoder.

6 References

[1] McEliece, R.J., "The Theory of Information and Coding," in *Encyclopedia of Mathematics and its Appliations, Volume 3*, Cambridge University Press, Cambridge, 1977.

[2] Peterson, W.W. and E.J. Weldon, Jr, *Error-Correcting Codes*, MIT Press, Cambridge, 1972.

[3] Lin, S. & D.J. Costello Jr., *Error Control Coding*, Prentice-Hall, Englewood Cliffs, 1983.

[4] Blahut, R.E., *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, 1983.

[5] MacWilliams, F.J. & N.J.A. Sloane, *The Theory of Error Correcting Codes*, North-Holland, Amsterdam, 1977.

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

[7] Massey, J.L., "Shift-register Synthesis and BCH Decoding," *IEEE Trans. Inf. Theory*, IT-18 (1969), 122–127.

[8] Michelson, A.M. & A.H. Levesque, *Error-Control Techniques for Digital Communication*, Wiley, New York, 1985.

[9] Pless, V., *Introduction to the Theory of Error-Correcting Codes*, Wiley-Interscience, New York, 1982.

[10] Feller, W., *An Introduction to Probability Theory and Its Applications*, Wiley, New York, 1968.

[11] van der Waerden, B.L., *Modern Algebra – Volume I*, Frederick Ungar, New York, 1953.

[12] van der Waerden, B.L., *Modern Algebra – Volume II*, Frederick Ungar, New York, 1950.

[13] Buchberger, B., "An Algorithmic Method in Polynomial Ideal Theory," in *Multidimensional Systems Theory*, N.K. Bose, ed., Mathematics and Its Applications, D. Reidel, Boston, 1985.

[14] Cooper, A.B., III, "Finding BCH error locator polynomials in one step," *Electronics Letters* 27 (24 October 1991), 2090–2091.

[15] Poli, A. & L. Huguet, *Error Correcting Codes: Theory and Applications*, Masson and Prentice Hall, Paris, 1992.

[16] Buchberger, B., *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, PhD Dissertation, University of Innsbruck, Math Institute, Austria, 1965.

INTENTIONALLY LEFT BLANK.

Appendix

Buchberger's Gröbner Basis Algorithm

What follows is a tutorial exposition of Buchberger's algorithm for finding a Gröbner basis of an ideal in the ring of multivariate polynomials [13,15]. Such an ideal is of interest here when it is defined by an algebraic manifold of roots of every member.

A Preliminaries

A.1 Notation

$$\begin{aligned} (i) &= (i_1, i_2, \ldots, i_m) \\ X^{(i)} &= X_1^{i_1}, X_2^{i_2}, \ldots, X_m^{i_m} \end{aligned} \quad (29)$$

A.2 Ordering

In what follows, we shall require that an *ordering* be defined on the multivariate monomials. The order of a multivariate polynomial is the analog of the degree of a univariate polynomial.

Let $\mathcal{R}$ be a *transitive* ordering on the monomials.

$$a \,\mathcal{R}\, b \cap b \,\mathcal{R}\, c \Rightarrow a \,\mathcal{R}\, c. \quad (30)$$

(Read $\mathcal{R}$ as "precedes" or as "is less than.") We require the following of $\mathcal{R}$:

- $1 \,\mathcal{R}\, X^{(i)}, \forall\, (i)$
- $X^{(i)} \,\mathcal{R}\, X^{(j)} \;\Rightarrow\; X^{(i)+(k)} \,\mathcal{R}\, X^{(j)+(k)}$

Any *admissible* ordering can be used.[11] Two examples follow.

1. The *lexicographic* ordering defines an order on the individual symbols, so that, *e.g.*, $X_1 \,\mathcal{R}\, X_2 \,\mathcal{R}\, \cdots$. (Some authors write $X_1 \leq X_2 \leq \cdots$). In this case, $X_1^2 X_2^3 \,\mathcal{R}\, X_1 X_2^6$.
2. For the *product ordering* P, monomials are ordered according to the exponents of *every* symbol, X_j. Therefore, $X_1^{i_1} X_2^{i_2} \,\mathcal{R}\, X_1^{j_1} X_2^{j_2}$ *iff* $i_1 \leq j_1$ and $i_2 \leq j_2$.

We shall need the concept of *supremum (sup)* or *least upper bound.* With an ordering defined on the monomials, *sup* is defined exactly as in mathematical analysis. It is the maximum over a set

[11] An ordering is said to be admissible if $1 \,\mathcal{R}\, X$ and $X \,\mathcal{R}\, Y$ imply $XU \,\mathcal{R}\, YU$.

with respect to the ordering defined on that set. If there is no maximum (*e.g.*, if two or more elements in the set are tied for largest according to the ordering), the smallest monomial larger than either is the *sup* over the set. For each of the example orderings, we give the supremum. For the lexicographic ordering,

$$sup(X_1^2X_2^3, X_1X_2^6) = X_1{}^2X_2^3; \tag{31}$$

for the product ordering,

$$sup(X_1^2X_2^3, X_1X_2^6) = X_1{}^2X_2^6. \tag{32}$$

A.3 Derivation of $Spol(f_i, f_j)$

Definition: $Hterm(f) \triangleq$ the *maximal monomial* or *head term* of $f(\mathbf{X})$ with respect to $\mathcal{R}$. If $f = f_{(i)}X^{(i)} + \bar{f}$ then $Hterm(f) = X^{(i)}$.

For two polynomials $f,\ h\ \in\ F[\mathbf{X}]$ define

$$SUP(f,h) = sup(Hterm(f), Hterm(h)). \tag{33}$$

Then express each polynomial explicitly as the sum of its head term (multiplied by the appropriate scalar coefficient) and the rest of the polynomial.

$$\begin{aligned} f &= f_{(i)}Hterm(f) + \bar{f} \\ h &= h_{(i)}Hterm(h) + \bar{h} \end{aligned} \tag{34}$$

and define

$$Spol(f,h) = h_{(i)}\frac{SUP(f,h)}{Hterm(f)}f \ - \ f_{(i)}\frac{SUP(f,h)}{Hterm(h)}h. \tag{35}$$

It is easy to see that $Spol(f,h)$ has order less than that of either f or h.

A.4 Reduction Modulo F

Let $F = \{f_j, \in\ K[X_1, \ldots, X_m],\ j = 1, \ldots, r\}$. For each f_j, write

$$f_j = \mu_j Hterm(f_j)\ +\ \bar{f}_j. \tag{36}$$

Select some $h \in K[\mathbf{X}]$ such that at least one $Hterm(F_j)$ appears in h; *i.e.*,

$$h = \cdots + f_{(i)}Hterm(f_i) + \cdots \tag{37}$$

Now form

$$\begin{aligned} f_{(i)}\mu_{(j)}^{-1}X^{(i)-(j)}f_j &= f_{(i)}\mu_{(j)}^{-1}X^{(i)-(j)}(\mu_{(j)}X^{(j)} + \bar{f}_j) \\ &= f_{(i)}X^{(i)}\ +\ \text{other terms.} \end{aligned} \tag{38}$$

Finally, write

$$h' = h - f_{(i)}\mu_{(j)}^{-1}X^{(i)-(j)}f_j. \tag{39}$$

Now, h' no longer contains a monomial in $X^{(i)}$, and we say that h is *reduced to* h' modulo F[13]or that h' is an F − *derivative* of h[15].

Repeated application of reduction or derivation to h eventually, and in a finite number of steps [16], produces a polynomial which cannot be reduced further modulo F.

B The Gröbner Basis Algorithm

Let $F = \{f_1, f_2, \ldots, f_m\}$ be any basis of an ideal I in the ring of multivariate polynomials over GF(q). The following algorithm produces a Gröbner basis.

1. From F, select a pair (f_i, f_j) of polynomials not previously chosen.

2. Compute $Spol(f_i, f_j)$. By the process defined above, reduce $Spol$ to a polynomial f_{ij} which is F-irreducible.

3. If $f_{ij} = 0$ go to 1. Otherwise, add f_{ij} to the basis, then go to 1.

4. The algorithm, when it terminates (which it has been shown to do [13,15]), will have produced a GB = $(g_1, \ldots, g_m)$ for the ideal spanned by F. By construction $Spol(g_i, g_j) = 0 \ \forall g_i, g_j \in$ GB. It is well-known [13,15] that the existence of GB can always infer a *reduced* GB of the ideal.

INTENTIONALLY LEFT BLANK.

No. of Copies	Organization
2	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	Commander U.S. Army Materiel Command ATTN: AMCAM 5001 Eisenhower Ave. Alexandria, VA 22333-0001
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-CI-AD, Tech Publishing 2800 Powder Mill Rd. Adelphi, MD 20783-1145
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-CI-AD, Records Management 2800 Powder Mill Rd. Adelphi, MD 20783-1145
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-IMI-I Picatinny Arsenal, NJ 07806-5000
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-TDC Picatinny Arsenal, NJ 07806-5000
1	Director Benet Weapons Laboratory U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050
1	Director U.S. Army Advanced Systems Research and Analysis Office (ATCOM) ATTN: AMSAT-R-NR, M/S 219-1 Ames Research Center Moffett Field, CA 94035-1000

No. of Copies	Organization
1	Commander U.S. Army Missile Command ATTN: AMSMI-RD-CS-R (DOC) Redstone Arsenal, AL 35898-5010
1	Commander U.S. Army Tank-Automotive Command ATTN: AMSTA-JSK (Armor Eng. Br.) Warren, MI 48397-5000
1	Director U.S. Army TRADOC Analysis Command ATTN: ATRC-WSR White Sands Missile Range, NM 88002-5502
(Class. only) 1	Commandant U.S. Army Infantry School ATTN: ATSH-CD (Security Mgr.) Fort Benning, GA 31905-5660
(Unclass. only) 1	Commandant U.S. Army Infantry School ATTN: ATSH-WCB-O Fort Benning, GA 31905-5000
1	WL/MNOI Eglin AFB, FL 32542-5000
	Aberdeen Proving Ground
2	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen
1	Cdr, USATECOM ATTN: AMSTE-TC
1	Dir, ERDEC ATTN: SCBRD-RT
1	Cdr, CBDA ATTN: AMSCB-CII
1	Dir, USARL ATTN: AMSRL-SL-I
5	Dir, USARL ATTN: AMSRL-OP-CI-B (Tech Lib)

No. of Copies	Organization
4	Commander U.S. Army CECOM ATTN: AMSEL-RD-C3-AC, Paul Sass ST-LA-L, Charles Graff ST-C, Dr. Felix Schwering ST-MC-M, Dr. Haim Soicher Fort Monmouth, NJ 07703
2	Director U.S. Army Research Office ATTN: AMXRO-EL, Dr. W. Sander AMXRO-MA, Dr. G. Anderson P.O. Box 12211 Research Triangle Park, NC 22709-2211
1	Commander U.S. Army Center for Signals Research ATTN: AMSEL-SW, S. Rhodes Vint Hill Farm Station Warrenton, VA 22186-5100
1	Commander U.S. Army Research, Development, and Standardization Group United Kingdom USARDSG (UK) Box 65 FPO New York 09510
1	Commander U.S. Army Research, Development, and Standardization Group Australia APO, SF 96404
1	Commandant U.S. Military Academy West Point, NY 10996
1	Commandant U.S. Naval Academy Annapolis, MD 21404
1	Commandant U.S. Air Force Academy Colorado Springs, CO 80840
1	Director Naval Research Laboratory ATTN: Code 7521, Dr. J. E. Wieselthier Washington, DC 20375-5000

No. of Copies	Organization
1	California Institute of Technology Department of Electrical Engineering ATTN: Prof. Rod Goodman Pasadena, CA 91125
1	Georgia Institute of Technology Coding and Information Theory Laboratory School of Electrical Engineering ATTN: Prof. Stephen B. Wicker Atlanta, GA 30332-0250
3	The Johns Hopkins University Department of Electrical and Computer Engineering ATTN: Prof. Brian Hughes Prof. Oliver Collins Prof. Willis C. Gore Baltimore, MD 21218
2	Cornell University Department of Mathematics ATTN: Prof. Mike Stillman Prof. Moss Sweedler Ithaca, NY 14853
1	Cornell University School of Electrical Engineering ATTN: Prof. Chris Heegard Ithaca, NY 14853
3	University of Delaware Department of Computer and Information Sciences ATTN: Prof. B. Caviness Prof. D. Saunders Prof. D. Wood Newark, DE 19716
1	Prof. Charles T. Retter 18 Ridgefield Drive Framingham, MA 01701
1	Texas A & M University Department of Mathematics ATTN: Prof. Peter Stiller College Station, TX 77843-3368

No. of Copies	Organization
1	Director Army Research Office ATTN: AMXRO-MCS, Mr. K. Clark P.O. Box 12211 Research Triangle Park, NC 27709-2211
1	Director Army Research Office ATTN: AMXRO-RT-IP, Library Services P.O. Box 12211 Research Triangle Park, NC 27709-2211
3	University of Virginia Department of Electrical Engineering ATTN: Prof. Stephen G. Wilson Prof. Demetrios Kazakos Prof. P. Papantoni-Kazakos Charlottesville, VA 22901

No. of Copies	Organization
	Aberdeen Proving Ground
20	Dir, USARL ATTN: AMSRL-CI, Mr. W. Mermagen AMSRL-CI-C, Dr. W. Sturek AMSRL-CI-CB, Mr. Richard Kaste AMSRL-CI-CC, Mr. B. Reichard Dr. P. Broome Dr. S. Chamberlain Mr. G. Hartwig Ms. B. Broome Mr. H. Caton Mr. A. Downs Ms. A. Brodeen Mr. D. Gwyn Ms. M. Lopez Mr. F. Brundick AMSRL-CI-S, Ms. V. Kaste AMSRL-CI-AC, Mr. P. Dykstra AMSRL-SS-SC, Ms. S. Stratton AMSRL-SS-IC, Dr. P. Emmerman AMSRL-SL-BV, Mr. E. Davisson AMSRL-WT-WB, Mr. R. McGee

No. of Copies	Organization
1	University of Wales at Bangor School of Electronic Engineering Science ATTN: Prof John O'Reilly Dean Street, Bangor, Gwynedd LL57 1UT UK
1	The University Department of Electrical Engineering ATTN: Prof P. G. Farrell Manchester ML3 9PL UK
1	IFI Institute of Advanced Microelectronics ATTN: Prof Patrick Fitzpatrick NMRC, Cork Ireland
1	University of Bristol Centre for Communication Research Faculty of Engineering ATTN: Prof Graham H. Norton UK
1	Universtity of Waterloo Department of Electrical Engineering ATTN: Prof Ian F. Blake Waterloo, Ontario, N2L 3G1 Canada
1	Waseda University Department of Industrial Engineering ATTN: Prof Shigeichi Hirasawa 3-4-1 Ohkubo, Shinjuku-ku Tokyo 160 Japan
1	Toyoashi University of Technology Department of Knowledge-Based Information Engineering ATTN: Prof Shojiro Sakata Tempaku, Toyohashi 440 Japan