International Journal of Electronics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713599654

Transform decoding of BCH codes over Z_m

B. Sundar Rajan [a]; M. U. Siddiqi [b]
[a] Electrical Engineering Department, Indian Institute of Technology, Delhi, India
[b] Electrical Engineering Department, Indian Institute of Technology, Kanpur, India

Online Publication Date: 01 December 1993
To cite this Article: Rajan, B. Sundar and Siddiqi, M. U. (1993) 'Transform decoding of BCH codes over Z_m', International Journal of Electronics, 75:6, 1043 - 1054
To link to this article: DOI: 10.1080/00207219308907181
URL: http://dx.doi.org/10.1080/00207219308907181

INT. J. ELECTRONICS, 1993, VOL. 75, NO. 6, 1043-1054

Transform decoding of BCH codes over Z_m

B. SUNDAR RAJAN† and M. U. SIDDIQI‡

For BCH codes with symbols from rings of residue class integers modulo m, denoted by Z_m, we introduce the analogue of Blahut's frequency domain approach for codes over finite fields and show that the problem of decoding these codes is equivalent to the minimal shift register synthesis problem over Galois rings. A minimal shift register synthesis algorithm over Galois rings is obtained by straightforward extention of the Reeds-Sloane algorithm which is for shift register synthesis over Z_m.

1. Introduction

The transform domain description of BCH codes over $GF(q)$ using discrete Fourier transform (DFT) defined over an extension field $GF(q^m)$ is well known. Specifically, a BCH code over $GF(q)$ that corrects t errors is defined as the cyclic code over $GF(q)$ for which $2t$ consecutive DFT coefficients of all codewords are equal to zero (Blahut 1979, 1983). These codes have a simple decoding algorithm, the Berlekamp-Massey algorithm, which is equivalent to synthesizing minimal feedback shift register over the extension field $GF(q^m)$ (Massey 1969).

The important class of BCH codes over finite fields has been generalized to cover symbols from an arbitrary residue class integer ring Z_m (Prithi Shankar 1979). Prithi Shankar (1979) derived BCH codes over Z_m in terms of their generator polynomials as follows. By the Chinese remainder theorem, it is sufficient to consider the case $m=p^k$. For $m=p^k$, a cyclic code of length n over Z_{p^k} is defined as an ideal in the ring of polynomials with coefficients from Z_{p^k} modulo (x^n-1) that is generated by any monic polynomial $g(x)$ that divides (x^n-1). To characterize these cyclic codes in terms of roots of $g(x)$, an extension ring of Z_{p^k}, called the Galois ring (McDonald 1974), is used. The polynomial (x^n-1) factors uniquely in the Galois ring, and since $g(x)$ is a factor of (x^n-1) a subset of roots of (x^n-1) uniquely specifies the cyclic code generated by $g(x)$. If the roots that specify a cyclic code are consecutive powers of an element (one of the roots of (x^n-1)) then the cyclic code is called a BCH code over Z_{p^k}. In this correspondence, it is shown that the decoding problem for these BCH codes is equivalent to the minimal shift register synthesis problem over Galois rings and an algorithm for which is obtained by observing that the shift register synthesis algorithm of Reeds and Sloane (Reeds and Sloane 1985) for Z_m is also valid for Galois rings. This result is the counterpart of the Berlekamp-Massey algorithm for decoding BCH codes over finite rings.

2. Transform description of BCH codes over Z_{p^k}

A finite commutative ring with identity, denoted by R, is said to support a DFT of length n if there exists a transform of the form

Received 24 May 1993; accepted 20 July 1993.

† Electrical Engineering Department, Indian Institute of Technology, Delhi 110016, India.

‡ Electrical Engineering Department, Indian Institute of Technology, Kanpur 208016, India.

$$A_j=\sum_{i=0}^{n-1}\alpha^{ij}a_i, \qquad j=0,1,\ldots,n-1$$

where $a_i, A_j \in R$, and α is a unit of order n in the group of units of R and n is invertible in R (Dubios and Venetsanopoulos 1978a). The element α is called the transform factor of the DFT.

This DFT defines an isomorphism between convolution algebra $R[x]/(x^n-1)$ and pointwise product algebra of n-tuples of R, denoted by R^n. In other words, if $(A_0, A_1, \ldots, A_{n-1})$ and $(B_0, B_1, \ldots, B_{n-1})$ are the transform vectors of $(a_0, a_1, \ldots, a_{n-1})$ and $(b_0, b_1, \ldots, b_{n-1})$ then the cyclic convolution

$$a_k=\sum_{i=0}^{n-1}a_i b_{(k-i)\bmod n}, \quad k=0,1,\ldots,n-1$$

has the transform vector $(C_0, C_1, \ldots, C_{n-1})$ where $C_k=A_kB_k$, $k=0,1,\ldots,n-1$. This property is known as the convolution property of the DFT.

Let p^k be a power of a prime number. Galois rings are residue class polynomial rings $Z_{p^k}[x]/\Phi(x)$, denoted by $GR(p^k,r)$, where $Z_{p^k}[x]$ is the ring of polynomials over Z_{p^k} and $\Phi(x)$ is a monic irreducible polynomial of degree r over $Z_p[x]$ and hence over $Z_{p^k}[x]$ (McDonald 1974). For a Galois ring $GR(p^k,r)$ to support a DFT of length n, it is required that n and p must be relatively prime, i.e., $(n,p)=1$, and n should divide p^r-1, for only then can an element α of order n exist in $GR(p^k,r)$ (Dubois and Venetsanopoulos 1978). Now it is clear that $\oplus_{i=1}^{s} GR(p_i^{k_i},r)$ can support a DFT of length n if and only if the following two conditions are satisfied. Firstly $(n,p_i)=1$, where $i=0,1,\ldots,s$; and secondly, n divides $gcd\{(p_1^r-1), (p_2^r-1),\ldots,(p_s^r-1)\}$. Therefore throughout, it is assumed that the length of the code, denoted by n, is relatively prime to m.

A property, known as the conjugate symmetry property (Dubois and Venetsanopoulos 1978b), holds in the case of DFT over Galois rings. Let $((a_0, a_1, \ldots, a_{n-1})$ be an n-tuple over Z_{p^k} and $(A_0, A_1, \ldots, A_{n-1}) \in GR^n(p^k,r)$ be its transform vector, where $GR(p^k,r)$ is the extension ring of Z_{p^k} which supports the DFT. We have

$$A_j=\sum_{i=0}^{n-1}\alpha^{ij}a_i, \quad j=0,1,\ldots,n-1$$

where α is an element of order n in the group of units of $GR(p^k,r)$ denoted by $GR^*(p^k,r)$. The automorphism group of $GR(p^k,r)$ is a cyclic group of order r and the generator automorphism is $\sigma(\alpha)=\alpha^p$. The conjugacy constraint in this case is given by $A_{pj}=\sigma(A_j)$ where $j=0,1,\ldots,n-1$. All the n-tuples of $GR(p^k,r)$ which are DFT vectors of some n-tuple over Z_{p^k} satisfy this condition. This property is called the conjugate symmetry property of DFT over Galois rings. It is the counterpart of the conjugacy constraints (Blahut 1979, 1983) in the case of DFT over finite fields.

Definition 1

Given a positive integer n and a prime p relatively prime to n, the conjugacy class containing j, $(0 \leqslant j \leqslant n)$, denoted by $C_{p,n}(j)$ is the set $\{j, pj, p^2j, \ldots, p^{(e-1)}j\}$ where e is the least integer such that $p^ej=j(\bmod n)$. Such an integer exists because of the relative primality of n and p. The integer e is called the exponent of the conjugacy class $C_{p,n}(j)$.

Given p^k and length n, the DFT is constructed as follows. Choose the least integer r such that n divides $(p^r - 1)$. The required extension ring is $GR(p^k, r)$. The group of units $GR^*(p^k, r)$ contains a cyclic sub-group G_1 whose order is $(p^r - 1)$. Further, since n divides $(p^r - 1)$ and element α exists in G_1 whose order is n. Hence $GR(p^k, r)$ supports a DFT of length n over Z_{p^k}.

Definition 2

Let $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ be an n-tuple over Z_{p^k}. The DFT of $\mathbf{a}$ is defined as $A_j = \sum_{i=0}^{n-1} \alpha^{ij} a_i$; $j = 0, 1, \ldots, n-1$, where α is an element of multiplicative order n in $GR(p^k, r)$, where r is the least integer such that n divides $(p^r - 1)$. The vector $\mathbf{A} = (A_0, A_1, \ldots, A_{n-1})$ is called the transform vector or spectrum of $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$. The components $A_i, i = 1, 2, \ldots, n$ are called DFT coefficients or spectral components of $\mathbf{a}$.

Only those n-tuples over $GR(p^k, r)$, which satisfy the conjugacy constraints will be transform vectors of n-tuples over Z_{p^k}. Since DFT defines an isomorphism all such n-tuples will form a sub-ring, denoted by R_t, of pointwise product algebra $GR^n(p^k, r)$. We have $Z_{p^k}[x]/(x^n - 1) = R_t$. The following theorem, proved by Sundar Rajan and Siddiqi (1994), characterizes R_t.

Theorem 1 (Sundar Rajan and Siddiqi 1994)

The sub-ring R_t of $GR^n(p^k, r)$, which contains all the transform vectors of n-tuples over Z_{p^k} is isomorphic to $\oplus_{i=0}^{t} GR(p^k, r_i)$ where t is the number of conjugacy classes and r_i are the exponents of the conjugacy classes for the integer n and prime p.

It can be shown (Madhusudhana 1987) that the number of conjugacy classes t is given by

$$t = \sum_{d|n}^{1} \frac{(\phi(d))}{e(d)}$$

where $e(d)$ is the least integer such that $d^e = 1 (\bmod q)$, where $q = n/d$ and $\phi(d)$ is the Euler's totient function.

For a given p^k, a cyclic code of length n over Z_{p^k} is an ideal in the ring of polynomials with coefficients from Z_{p^k} modulo the polynomial $(x^n - 1)$ and is generated by any monic polynomial $g(x)$ that divides $(x^n - 1)$ (Prithi Shankar 1979). (The transform domain study of cyclic codes for the case $g(x)$ being not monic leads to interesting situations which have been reported by Sundar Rajan and Siddiqi 1994.) If the roots of $g(x)$ in the group of units of the extension ring $GR(p^k, r)$ are consecutive powers of α, where α is a primitive nth root of unity, the cyclic code is called a BCH code over Z_{p^k}. Let $g(x)$ be the generator polynomial of a BCH code with the $2t$ consecutive roots being $\alpha^h, \alpha^{h+1}, \ldots, \alpha^{h+2t-1}$. Let $(G_1, G_2, \ldots, G_{n-1})$ be the transform vector of $g(x)$. We have

$$G_j = \sum_{i=0}^{n-1} \alpha^{ij} g_i = g(\alpha^j) = 0 \quad \text{for} \quad j = h, h+1, \ldots, h+2t-1$$

Hence a BCH code can be defined in transform domain as follows.

Definition 3

A BCH code that corrects up to t errors consists of the inverse DFT of vectors of R_t with $2t$ number of consecutive components equal to zero.

Such a code has minimum Hamming distance $2t+1$ and hence corrects up to t errors (Prithi Shankar 1979). Without loss of generality we assume that the BCH code under consideration for decoding has first $2t$ DFT components zero.

Example 1

A double error correcting BCH code over Z_9 of length 8. The appropriate extension ring is GR(9, 2). The conjugacy classes are $\{0\}$, $\{1,3\}$, $\{2,6\}$, $\{4\}$ and $\{5,7\}$. The transform matrix is

$$\begin{matrix}
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 31 & 75 & 28 & 80 & 68 & 24 & 71 \\
10 & 75 & 80 & 24 & 10 & 75 & 80 & 24 \\
10 & 28 & 24 & 31 & 80 & 71 & 75 & 68 \\
10 & 80 & 10 & 80 & 10 & 80 & 10 & 80 \\
10 & 68 & 75 & 71 & 80 & 31 & 24 & 28 \\
10 & 24 & 80 & 75 & 10 & 24 & 80 & 75 \\
10 & 71 & 24 & 68 & 80 & 28 & 75 & 31
\end{matrix}$$

where each entry ab represents $a+bx$, an element of $GR(9,2)$. The transform factor is $3+x$. The automorphism defining the conjugacy constraint is $\sigma(x)=8+8x$. For double error correction, it is sufficient to have four consecutive zeros. However, we consider the code with the first five DFT coefficients zeros. The conjugacy classes which take zeros are $\{0\}$, $\{1,3\}$, $\{2,6\}$ and $\{4\}$. A complete listing of all the codewords with their DFT coefficients is given in Table 1.

3. Equivalence of decoding BCH codes over Z_{p^k} to shift register synthesis over Galois ring

In this section we show that the problem of decoding BCH codes over Z_{p^k} is equivalent to the minimal feedback shift register synthesis problem over Galois rings. The BCH codes under consideration for decoding are t-error correcting of length n and without loss of generality, it is assumed that the first $2t$ consecutive DFT coefficients are zeros.

Let us associate with an n-tuple $\mathbf{a}=(a_0, a_1, \ldots, a_{n-1})$ over Z_{p^k} the polynomial

$$a(x)=a_0+a_1x+a_2x^2+\ldots+a_{n-1}x^{n-1}\in Z_{p^k}[x]$$

Codeword	Spectrum	Codeword	Spectrum
0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	4 5 2 6 5 4 7 3	00 00 00 00 00 04 00 55
0 1 5 8 0 8 4 1	00 00 00 00 00 51 00 48	4 6 7 5 5 3 2 4	00 00 00 00 00 55 00 04
0 2 1 7 0 7 8 2	00 00 00 00 00 12 00 87	4 7 3 4 5 2 6 5	00 00 00 00 00 16 00 43
0 3 6 6 0 6 3 3	00 00 00 00 00 63 00 36	4 8 8 3 5 1 1 6	00 00 00 00 00 67 00 82
0 4 2 5 0 5 7 4	00 00 00 00 00 24 00 75	5 0 5 7 4 0 4 2	00 00 00 00 00 71 00 68
0 5 7 4 0 4 2 5	00 00 00 00 00 75 00 24	5 1 1 6 4 8 8 3	00 00 00 00 00 32 00 17
0 6 3 3 0 3 6 6	00 00 00 00 00 36 00 63	5 2 6 5 4 7 3 4	00 00 00 00 00 83 00 56
0 7 8 2 0 2 1 7	00 00 00 00 00 87 00 12	5 3 2 4 4 6 7 5	00 00 00 00 00 44 00 05
0 8 4 1 0 1 5 8	00 00 00 00 00 48 00 51	5 4 7 3 4 5 2 6	00 00 00 00 00 05 00 44
1 0 1 5 8 0 8 4	00 00 00 00 00 52 00 37	5 5 3 2 4 4 6 7	00 00 00 00 00 56 00 83
1 1 6 4 8 8 3 5	00 00 00 00 00 13 00 76	5 6 8 1 4 3 1 8	00 00 00 00 00 17 00 32
1 2 2 3 8 7 7 6	00 00 00 00 00 64 00 25	5 7 4 0 4 2 5 0	00 00 00 00 00 68 00 71
1 3 7 2 8 6 2 7	00 00 00 00 00 25 00 64	5 8 0 8 4 1 0 1	00 00 00 00 00 20 00 20
1 4 3 1 8 5 6 8	00 00 00 00 00 76 00 13	6 0 6 3 3 0 3 6	00 00 00 00 00 33 00 06
1 5 8 0 8 4 1 0	00 00 00 00 00 37 00 52	6 1 2 2 3 8 7 7	00 00 00 00 00 84 00 45
1 6 4 8 8 3 5 1	00 00 00 00 00 88 00 01	6 2 7 1 3 7 2 8	00 00 00 00 00 45 00 84
1 7 0 7 8 2 0 2	00 00 00 00 00 40 00 40	6 3 3 0 3 6 6 0	00 00 00 00 00 06 00 33
1 8 5 6 8 1 4 3	00 00 00 00 00 01 00 88	6 4 8 8 3 5 1 1	00 00 00 00 00 57 00 72
2 0 2 1 7 0 7 8	00 00 00 00 00 14 00 65	6 5 4 7 3 4 5 2	00 00 00 00 00 18 00 21
2 1 7 0 7 8 2 0	00 00 00 00 00 65 00 14	6 6 0 6 3 3 0 3	00 00 00 00 00 60 00 60
2 2 3 8 7 7 6 1	00 00 00 00 00 26 00 53	6 7 5 5 3 2 4 4	00 00 00 00 00 21 00 18
2 3 8 7 7 6 1 2	00 00 00 00 00 77 00 02	6 8 1 4 3 1 8 5	00 00 00 00 00 72 00 57
2 4 4 6 7 5 5 3	00 00 00 00 00 38 00 41	7 0 7 8 2 0 2 1	00 00 00 00 00 85 00 34
2 5 0 5 7 4 0 4	00 00 00 00 00 80 00 80	7 1 3 7 2 8 6 2	00 00 00 00 00 46 00 73
2 6 5 4 7 3 4 5	00 00 00 00 00 41 00 38	7 2 8 6 2 7 1 3	00 00 00 00 00 07 00 22
2 7 1 3 7 2 8 6	00 00 00 00 00 02 00 77	7 3 4 5 2 6 5 4	00 00 00 00 00 58 00 61
2 8 6 2 7 1 3 7	00 00 00 00 00 53 00 26	7 4 0 4 2 5 0 5	00 00 00 00 00 10 00 10
3 0 3 6 6 0 6 3	00 00 00 00 00 66 00 03	7 5 5 3 2 4 4 6	00 00 00 00 00 61 00 58
3 1 8 5 6 8 1 4	00 00 00 00 00 27 00 42	7 6 1 2 2 3 8 7	00 00 00 00 00 22 00 07
3 2 4 4 6 7 5 5	00 00 00 00 00 78 00 81	7 7 6 1 2 2 3 8	00 00 00 00 00 73 00 46
3 3 0 3 6 6 0 6	00 00 00 00 00 30 00 30	7 8 2 0 2 1 7 0	00 00 00 00 00 34 00 85
3 4 5 2 6 5 4 7	00 00 00 00 00 81 00 78	8 0 8 4 1 0 1 5	00 00 00 00 00 47 00 62
3 5 1 1 6 4 8 8	00 00 00 00 00 42 00 27	8 1 4 3 1 8 5 6	00 00 00 00 00 08 00 11
3 6 6 0 6 3 3 0	00 00 00 00 00 03 00 66	8 2 0 2 1 7 0 7	00 00 00 00 00 50 00 50
3 7 2 8 6 2 7 1	00 00 00 00 00 54 00 15	8 3 5 1 1 6 4 8	00 00 00 00 00 11 00 08
3 8 7 7 6 1 2 2	00 00 00 00 00 15 00 54	8 4 1 0 1 5 8 0	00 00 00 00 00 62 00 47
4 0 4 2 5 0 5 7	00 00 00 00 00 28 00 31	8 5 6 8 1 4 3 1	00 00 00 00 00 23 00 86
4 1 0 1 5 8 0 8	00 00 00 00 00 70 00 70	8 6 2 7 1 3 7 2	00 00 00 00 00 74 00 35
4 2 5 0 5 7 4 0	00 00 00 00 00 31 00 28	8 7 7 6 1 2 2 3	00 00 00 00 00 35 00 74
4 3 1 8 5 6 8 1	00 00 00 00 00 82 00 67	8 8 3 5 1 1 6 4	00 00 00 00 00 86 00 23
4 4 6 7 5 5 3 2	00 00 00 00 00 43 00 16		

Table 1. Codewords and their spectrum of the BCH code of Example 1.

Let $\mathbf{c}=(c_0,c_1,\ldots,c_{n-1})$ be the transmitted codeword and $\mathbf{r}=(r_0,r_1,\ldots,r_{n-1})$ and $\mathbf{e}=(e_0,e_1,\ldots,e_{n-1})$ be the received and error vectors, respectively. The associated polynomials are

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

$$r(x)=r_0+r_1x+r_2x^2+\ldots+r_{n-1}x^{n-1}$$

$$e(x)=e_0+e_1x+e_2x^2+\ldots+e_{n-1}x^{n-1}$$

Assuming that only $\delta\leqslant t$ errors have occurred, we have only δ non-zero coefficients in $e(x)$. Let

$$e(x)=e_{i_1}x^{i_1}+e_{i_2}x^{i_2}+\ldots+e_{i_\delta}x^{i_\delta}$$

i.e., $i_1,i_2,\ldots,i_\delta$ are the locations of the errors and $e_{i_1},e_{i_2},\ldots,e_{i_\delta}$ are the magnitudes of the errors. Both locations and magnitudes are unknown. The decoding problem is to find these. Instead of finding error locations and magnitudes which means finding $e(x)$, we obtain the transform vector of $e(x)$, the inverse DFT of which gives $\mathbf{e}$ straightaway.

Let α be the transform factor of the DFT. We define S_j (the jth syndrome) as $S_j=r(\alpha^j)$. Note that S_j is nothing but the jth DFT coefficient of the received vector. Since $r(x)=c(x)+e(x)$ and $c(x)=0$ for $x=\alpha^0,\alpha^1,\ldots,\alpha^{2t-1}$, the syndromes contain information due to errors only, i.e., the first $2t$ DFT coefficients of the error vector are equal to the syndromes $S_0,S_1,\ldots,S_{2t-1}$. So our aim is to obtain $e(x)$ such that δ, the degree of $e(x)$ is at a minimum and also the first $2t$ DFT coefficients are equal to syndromes.

Let us define the polynomial $A(x)$, called the error locater polynomial, by

$$A(x)=(1-x\alpha^{i_1})(1-x\alpha^{i_2})\ldots(1-x\alpha^{i_\delta})$$

The degree of $A(x)$ is δ, which is utmost t, and $A(x)$ is a polynomial with coefficients in $GR(p^k,r)$. Let

$$A(x)=1+A_1x+A_2x^2+\ldots+A_{n-1}x^{n-1}$$

The inverse DFT of $A(x)$ is given by $A(\alpha^{-j}), j=0,1,\ldots,n-1$, which is same as $A(x)$ evaluated at α^{-j}. We denote this inverse DFT of $A(x)$ by $\boldsymbol{\Gamma}=(\Gamma_0,\ \Gamma_1,\ldots,\Gamma_{n-1})$. Note that $\boldsymbol{\Gamma}$ is an n-tuple over $GR(p^k,r)$. Since $A(x)$, in general, does not satisfy the conjugacy constraints, inverse DFT is not an n-tuple over Z_{p^k}. By the definition of $A(x)$, $A(\alpha^{-j})$ is equal to zero if and only if j is an error location. Thus $A(x)$ has been defined in such a way that in $\boldsymbol{\Gamma}$, $\Gamma_i=0$ for all those i for which $e_i\neq 0$. Hence $\Gamma_ie_i=0$ for all $i=0,1,\ldots,n-1$. By the convolution property of the DFT, the convolution of transform vector of $\boldsymbol{\Gamma}$ and the transform vector of error vector, denoted by $\mathbf{E}=(E_0,E_1,\ldots,E_{n-1})$ is equal to the zero vector. That is

$$\sum_{i=0}^{n-1} A_iE_{k-i}=0, \quad k=0,1,\ldots,n-1$$

Because $A(x)$ has a degree equal to δ we have $A_j=0$ for $j>\delta$. Therefore

$$\sum_{i=0}^{\delta} A_iE_{k-i}=0 \quad k=0,1,\ldots,n-1$$

Since $A_0 = 1$, we have

$$E_k = -\sum_{i=0}^{\delta} A_i E_{k-i}, \quad k = 0, 1, \ldots, n-1$$

The coefficients A_i, $i = 1, 2, \ldots, \delta$, are unknown and among the n components of E only $2t$ are known which are equal to syndromes. Thus

$$S_k = -\sum_{i=0}^{\delta} A_i S_{k-i}, \quad k = \delta, \delta+1, \ldots, 2t-1$$

involve only the known syndromes and the δ unknown components of A. From the above equations it follows that the problem of obtaining $A_0, A_1, \ldots, A_\delta$ is nothing but synthesizing the minimal feedback shift register with tap coefficients $A_0, A_1, \ldots, A_\delta$ that generate the $S_0, S_1, \ldots, S_{2t-1}$. Note that $S_0, S_1, \ldots, S_{2t-1}$ and $A_0, A_1, \ldots, A_\delta$ belong to a Galois ring and the requirement of minimizing δ is taken care of since the synthesis is for the minimal length shift register. The problem of decoding BCH codes over Z_m is accordingly, equivalent to the minimal shift register synthesis problem over the Galois ring. By recursive extension, $S_{2t}, S_{2t+1}, \ldots, S_{n-1}$ can be obtained and the inverse Fourier transform of $(S_0, S_1, \ldots, S_{n-1})$ straightaway gives the error vector $(e_0, e_1, \ldots, e_{n-1})$.

4. A sample computation of BCH decoding algorithm

In this section we display the computation of the algorithm for the BCH code given in Example 1. We assume that the transmitted codeword is (1 1 6 4 8 8 3 5) and the error vector is (0 0 0 5 0 1 0 0). Then the received vector is (1 1 6 0 8 0 3 5). The transform vector of the received vector is (60, 73, 87, 46, 30, 30, 12, 30). Hence, we have the syndromes

$$S_0 = 60, \; S_1 = 73, \; S_2 = 87, \; S_3 = 46, \; S_4 = 30$$

$$\text{i.e., } S(x) = (60) + (73)x + (87)x^2 + (46)x^3 + (30)x^4$$

The computation of every step except the last of the algorithm for the above given $S(x)$ is shown in Table 3.

In the final step we have

$$a_0^{(5)} = (10) + (12)x + (10)x^2$$

$$b_0^{(5)} = (60) + (46)x \quad \text{and} \quad L(A_0^{(5)}) = 2$$

Hence the connection polynomial is

$$a(x) = (10) + (12)x + (10)x^2$$

and

$$S(x)a(x) = b(x) (\text{mod } x^5)$$

We have

μ	t	θ	$\theta(\text{mod } 3^{(2-t)})$	μ	t	θ	$\theta(\text{mod } 3^{(2-t)})$	μ	t	θ	$\theta(\text{mod } 3^{(2-t)})$
01	0	01	01	30	1	10, 40, 70	10	60	1	20, 50, 80	20
02	0	02	02	31	0	31	31	61	0	61	61
03	1	01, 04, 07	01	32	0	32	323	62	0	62	62
04	0	04	04	33	1	11, 44, 77	11	63	1	21, 54, 87	21
05	0	05	05	34	0	34	34	64	0	64	64
06	1	02, 05, 08	02	35	0	35	35	65	0	65	65
07	0	07	07	36	1	12, 45, 78	12	66	1	22, 55, 88	22
08	0	08	08	37	0	37	37	67	0	67	67
10	0	10	10	38	0	38	38	68	0	68	68
11	0	11	11	40	0	40	40	70	0	70	70
12	0	12	12	41	0	41	41	71	0	71	71
13	0	13	13	42	0	42	42	72	0	72	72
14	0	14	14	43	0	43	43	73	0	73	73
15	0	15	15	44	0	44	44	74	0	74	74
16	0	16	16	45	0	45	45	75	0	75	75
17	0	17	17	46	0	46	46	76	0	76	76
18	0	18	18	47	0	47	47	77	0	77	77
20	0	20	20	48	0	48	48	78	0	78	78
21	0	21	21	50	0	50	50	80	0	80	80
22	0	22	22	51	0	51	51	81	0	81	81
23	0	23	23	52	0	52	52	82	0	82	82
24	0	24	24	53	0	53	53	83	0	83	83
25	0	25	25	54	0	54	54	84	0	84	84
26	0	26	26	55	0	55	55	85	0	85	85
27	0	27	27	56	0	56	56	86	0	86	86
28	0	28	28	57	0	57	57	87	0	87	87
				58	0	58	58	88	0	88	88

Table 2. Representation of non-zero elements of $GR(9,2)$.

$$S_j = -(a_1 S_{j-1} + a_2 S_{j-2})$$

Putting $j = 5$, 6, 7 successively we obtain the unknown syndromes $S_5 = (26)$, $S_6 = (12)$ and $S_7 = (53)$. The inverse transform of $(S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7)$ gives the error vector to be $(0\ \ 0\ \ 0\ \ 5\ \ 0\ \ 1\ \ 0\ \ 0)$. Hence the transmitted codeword is $(1\ \ 1\ \ 6\ \ 4\ \ 8\ \ 8\ \ 3\ \ 5)$.

Appendix

Shift register synthesis algorithm over Galois ring

This algorithm is exactly same as that of Reeds and Sloane (1985) except that instead of the ring Z_m it is discussed for Galois rings. This is given only for the purpose of completion.

Now we proceed to describe the shift register synthesis algorithm over a Galois ring. For Z_{p^k}, the minimal shift register synthesis algorithm has been obtained by Reeds and Sloane. We now show that this algorithm is also valid for minimal shift register synthesis over Galois rings. Our presentation is very similar to that of Reeds and Sloane (1985) and familiarity with that paper will be useful in following the algorithm.

The following property of the Galois rings is the only idea that is required to be known, apart from the Reeds-Sloane algorithm for shift register synthesis over Z_m,

	$t=0$	$t=1$
Step 0 $k=0$	$a_0^{(0)}(x)=(10)$ $b_0^{(0)}(x)=(00)$ $L(A_0^{(0)})=0$ $u_{00}=(20)$ $t_{00}=1$ $f(0,0)=0$	$a_1^{(0)}(x)=3(10)$ $b_1^{(0)}(x)=(00)$ $L(A_1^{(0)})=0$ $u_{10}=(10)$ $t_{10}=2$ $f(1,0)=0$
Step 1 $k=1$	$a_0^{(1)}(x)=(10)$ $b_0^{(1)}(x)=3(20)=(60)$ $L(A_0^{(1)})=1$ $u_{01}=(73)$ $t_{01}=0$ $g=f(0,1)=1$	$a_1^{(1)}(x)=3(10)$ $b_1^{(1)}(x)=3.3.(20)=(00)$ $L(A_1^{(1)})=0$ $u_{11}=(10)$ $t_{11}=1$ $g=f(1,1)=0$ $h=0;\ r=0$
Step 2 $k=2$	$a_0^{(2)}(x)=(10)$ $b_0^{(2)}(x)=(60)+(73)x$ $L(A_0^{(2)})=2$ $u_{02}=(87)$ $t_{02}=0$ $g=f(0,2)=1$ $h=0; r=1$	$a_1^{(2)}(x)=3(10)+(40)x$ $b_1^{(2)}(x)=(00)$ $L(A_1^{(2)})=1$ $u_{12}=(76)$ $t_{12}=0$ $g=f(1,2)=1$ $h=0;\ r=1$
Step 3 $k=3$	$a_0^{(3)}(x)=(10)+(72)x$ $b_0^{(3)}(x)=(60)+(40)x$ $L(A_0^{(3)})=2$ $u_{03}=(53)$ $t_{03}=0$ $g=f(0,3)=1$ $h=1; r=2$	$a_1^{(3)}(x)=(30)+(36)x$ $\mathrm{b}_1^{(3)}(x)=(30)x$ $L(A_1^{(3)})=2$ $u_{13}=(20)$ $t_{13}=1$ $g=f(1,3)=0$ $h=1; r=1$
Step 4 $k=4$	$a_0^{(4)}(x)=(10)+(12)x+(10)x^2$ $b_0^{(4)}(x)=(60)+(46)x$ $L(A_0^{(4)})=2$ $u_{04}=(10)$ $t_{04}=2$	$a_1^{(4)}(x)=(30)+(36)x+(30)x^2$ $b_1^{(4)}(x)=(30)x$ $L(A_1^{(4)})=2$ $u_{14}=(10)$ $t_{14}=2$

Table 3. Computation steps.

to obtain an algorithm that works over the Galois ring. In the Galois ring $GR(p^k, r)$ any non-zero element μ can be written as θp^t where θ is a unit and $0 \leqslant t \leqslant k-1$. In this representation the integer t is unique and θ is unique modulo (p^{k-t}). Note that this property holds for Z_{p^k} since Z_{p^k} is nothing but the Galois ring $GR(p^k, 1)$.

Example

Consider $GR(9,2)=Z_9[x]/(x^2+x+2)$. Any non-zero element μ of $GR(9,2)$ is of the form $a+bx$ where $a, b \in Z_9$ and it is denoted by ab. The representation of μ in the form θp^t for the elements of $GR(9,2)$ is given in Table 2.

Let $GR^*(p^e, r)$ denote the set of all units of the Galois ring $GR(p^e, r)$. The sequence $S_0, S_1, \ldots, S_{n-1}$ where $S_i \in GR(p^e, r)$, is said to be generated by a linear feedback shift register of length δ if there are elements $a_0=1, a_1, a_2, \ldots, a_\delta \in GR(p^e, r)$ such that

$$\sum_{i=0}^{\delta} a_i S_{j-i} = 0, \quad j = \delta, \delta+1, \dots, n-1 \tag{A 1}$$

Let $a(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_\delta x^\delta$ and $S(x) = S_0 + S_1 x + S_2 x^2 + \dots + S_{n-1} x^{n-1}$. Clearly $a(x)$ and $S(x)$ are in $GR(p^e, r)[x]$. Then (A 1) can be written as

$$S(x)a(x) = b(x)(\mathrm{mod}\, x^n) \quad a(0) = 1 \tag{A 2}$$

for some polynomial $b(x) \in GR(p^e, r)[x]$ of degree $\leqslant \delta - 1$. Thus the length of the shift register is $\delta = \max\{\deg a(x),\ 1 + \deg b(x)\}$. We write $A = (a(x), b(x))$ and define

$$L(A) = \max\{\deg a(x),\ 1 + \deg b(x)\}$$

By convention $\deg(0) = -\infty$.

The algorithm

Let $S_0, S_1, \dots, S_{n-1} \in GR(p^e, r)$ Our aim is to find $A = (a(x), b(x))$ of minimal length $\delta = L(A)$ satisfying (2). The following more general problem is considered. For all $i = 0, 1, 2, \dots, e-1$, find pairs $A_i = (a_i(x), b_i(x))$ such that

$$S(x)a_i(x) = b_i(x)(\mathrm{mod}\, x^n), \quad a_i(0) = p^i$$

and $L(A_i) = \delta_i$ is minimized. This algorithm is an iterative procedure that for all $0 \leqslant k \leqslant n$, $0 \leqslant i \leqslant e$ calculates the pairs

$$A_i^{(k)} = (a_i^{(k)}(x), b_i^{(k)}(x))$$

satisfying

$$S(x)a_i^{(k)}(x) = b_i^{(k)}(\mathrm{mod}\, x^k); \quad a_i^{(k')}(0) = p^i$$

and minimizing $L(A_i(k))$. Let $p^{t_{ik}} (0 \leqslant t_{ik} \leqslant e)$ be the highest power of p dividing the coefficient of x^k in

$$S(x)a_i^{(k')}(x) - b_i^{(k)}(x)$$

($t_{ik} = e$ if the coefficient of x^k is zero). Then at the kth step in the iteration, the following property holds for all $0 \leqslant j \leqslant k$. For all $0 \leqslant g < e$ either

$$L(A_g^{(j+1)}) = L(A_g^{(j)})$$

or else there exists $h = f(g, j)$ with

$$g + t_{hj} < e \tag{A 3}$$

$$L(A_g^{(j+1)}) = j + 1 - L(A_h^{(j)}) \tag{A 4}$$

$$L(A_g^{(j+1)}) > L(A_g^{(j)})$$

This property is analogous to the condition that Massey gives (Massey 1969, eqns. (11)–(13)) for the finite field case. Given this data our algorithm calculates

$A_i^{(k+1)}$ and $f(i,k), 0\leqslant i<e$, such that property P_k holds. The quantities $L(A_i^{(j)})$ also obey the inequality

$$L(A_{i+1}^{(k)})\leqslant L(A_i^{(k)})\leqslant L(A_i^{(k+1)})$$

Step 0: We start the algorithm with $k=0$ and for each $i=0,1,\ldots,e-1$, define

$$a_i^{(0)}(x)=p^i,\ b_i^{(0)}(x)=0;\quad a_i^{(1)}(x)=p^i,\ b_i^{(1)}(x)=p^iS_0$$

and

$$A_i^{(0)}=(a_i^{(0)}(x),\ b_i^{(0)}(x)),\ A_i^{(1)}=(a_i^{(1)}(x),b_i^{(1)}(x))$$

Let $S_0=Up^t$ for $U\in GR^*(p^e,r)$, $0\leqslant t\leqslant e$. (if $S_0=0$ set $U=1$ and $t=e$).

Then

$$L(A_i^{(0)})=0$$

and

$$\begin{aligned}L(A_i^{(1)})&=1 \quad \text{if} \quad i+t<e\\ &=0 \quad \text{if} \quad i+t\geqslant e\end{aligned}$$

We also define

$$u_{i0}=U,\ t_{i0}=i+t \quad \text{if} \quad i+t<e$$

$$u_{i0}=1,\ t_{i0}=e \quad \text{if} \quad i+t\leqslant e$$

Finally, we set $f(i,0)=0$ for all i.

The following step is carried out for each $k=1,2,\ldots,n-1$.

Step k (This produces $A_i^{(k+1)}$)

For each $i=0,1,\ldots,e-1$, we have the following calculations. Define $u_{ik}\in GR^*(p^e,r)$ and t_{ik}, $0\leqslant t_{ik}\leqslant e$, by

$$S(x)a_i^{(k)}(x)=b_i^{(k)}+u_{ik}p^{t_{ik}}x^k(\text{mod } x^{k+1})$$

($u_{ik}p^{t_{ik}}$ is the current discrepancy in the notation by Massey 1969)

Case I: If $t_{ik}=e$, set $A_i^{(k+1)}=A_i^{(k)}$.

Case II: If $t_{ik}<e$, define $g=e-1-t_{ik}$ so that $0\leqslant g<e$ and put $f(i,k)=g$.

There are now two subclasses.

Case II(a): If $L(A_g^{(g)})=0$, we set

$$A_i^{(k+1)}=A_i^{(k)}+(0,u_{ik}p^{t_{ik}}x^k).$$

Case II(b); If $L(A_g^{(k)})>0$, then for some $0\leqslant v<k$ we have

$$L(A_g^{(v)})<L(A_g^{(v+1)})=L(A_g^{(k)}) \tag{A 5}$$

v is the time of the most recent length change in the sequence $L(A_g^{(0)})$, $L(A_g^{(1)})$, $L(A_g^{(2)})\ldots.$ From (A 3), (A 4) and (A 5) it follows that

$$L(A_g^{(k)}) = L(A_g^{(v+1)}) = v + 1 - L(A_h^{(v)})$$

where $h = f(g, v)$ and $g + t_{hv} < e$.

Using $g = e - 1 - t_{ik}$ we have $t_{hv} \leqslant t_{ik}$. Thus the power of p from the past can be used to annihilate the power of p in the current discrepancy and we define

$$a_i^{(k+1)}(x) = a_i^{(k)} - u_{ik} u_{hv}^{-1} p^{t_{ik} - t_{hv}} x^{k-v} a_h^{(v)}(x)$$

$$b_i^{(k+1)}(x) = a_i^{(k)} - u_{ik} u_{hv}^{-1} p^{t_{ik} - t_{hv}} x^{k-v} b_h^{(v)}(x)$$

and

$$A_i^{(k+1)} = (a_i^{(k+1)}(x),\ b_i^{(k+1)}(x))$$

Then

$$S(x) a_i^{(k+1)}(x) = b_i^{(k+1)}(x) \quad \text{and} \quad a_i^{(k+1)}(0) = p^i$$

This concludes Step k.

At the end of step $(n-1)$ the algorithm terminates and the desired pair $A = (a(x), b(x))$ is given by $A_0^{(n)} = (a_0^{(n)}(x),\ b_0^{(n)}(x))$.

The proof for the correctness of this algorithm is same as that of Reeds-Sloane for shift register synthesis over Z_m. One obtains the proof for Galois rings by simply changing Z_m to $GR(p^k, r)$ in the Reeds-Sloane algorithm.

References

Blahut, R. E., 1979, *Algebraic Code in the Frequency Domain.* CISM Courses and Lectures, No. 258 (New York: Springer-Verlag); 1983, *Theory and Practice of Error Control Codes* (California: Addison-Wesley).

Dubois, E., and Venetsanopoulos, A. N., 1978 a, The discrete Fourier transform over finite rings with applications to fast convolution. *IEEE Transactions on Computers*, **27**, 593-596; 1978 b, Convolution using a conjugate symmetry property for the generalized discrete Fourier transform. *IEEE Transactions of Acoustics, Speech and Signal Processing*, **26**, 165-169.

Madhusudhana, H. S., 1987, On Abelian codes which are closed under cyclic shifts. M.Tech thesis, Indian Institute of Technology, Kanpur, India.

McDonald, B. R., 1974, *Finite Rings with Identity* (New York: Marcel Dekker).

Massey, J. L., 1969, Shift register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, **16**, 122-127.

Prithi Shankar, 1979, On BCH codes over arbitrary integer rings. *IEEE Transactions on Information Theory*, **25**, 480-483.

Reeds, J. A., and Sloane, N. J. A., 1985, Shift register synthesis (modulo *m). SIAM Journal of Computing*, **14**, 505-513.

Sundar Rajan, B., and Siddiqi, M. U., 1994, A transform approach to cyclic codes over Z_m. *Applicable Albebra, Communication and Computing* to be published.