C. M. Melas

A Cyclic Code for Double Error Correction

Since N. M. Abramson first applied shift-register sequences to correction of single and double adjacent errors,[1] a number of investigators have devised codes using such sequences for burst error correction.[2-4] Bose and Ray-Chandhuri have applied these sequences to construction of codes of arbitrary Hamming distance.[5]

A group of codes will be developed using maximal-length shift-register sequences for the correction of all double errors in a message. The codes described are systematic, and can be readily constructed for any block length. They can be implemented in the outstandingly simple manner described by J. E. Meggitt.[6]

A criterion will also be given which can be applied to the construction of sequence codes of arbitrary Hamming distance.

General structures of the codes

The codes described are Hamming-type codes for messages of n bits consisting of k information digits $D_1 \cdots D_k$, and m parity digits $P_1 \cdots P_m$ derived from the information digits in m equations, defined as follows:

$$\begin{vmatrix} a_{11}a_{12} \cdots \cdots \cdots a_{1n} \\ a_{21} \qquad\qquad\qquad \cdot \\ \cdot \qquad\qquad\qquad \cdot \\ \cdot \qquad\qquad\qquad \cdot \\ \cdot \qquad\qquad\qquad \cdot \\ \cdot \qquad\qquad\qquad \cdot \\ \cdot \qquad\qquad\qquad \cdot \\ a_{m1} \cdots \cdots \cdots a_{mn} \end{vmatrix} \cdot \begin{vmatrix} D_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ D_k \\ P_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ P_m \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{vmatrix} \qquad (1)$$

The nxm matrix a completely defines the code, and is sometimes called the check matrix. In decoding, the left side of (1) is generated, and if no errors occurred the right side will be the null vector as in (1); an error in the j^{th} bit will make the right side of (1), called the corrector, identical to the j^{th} column vector in the matrix. If all column vectors of the check matrix are different, a single-error correcting code results, since the right side of (1) will be a different vector for each single error.

In a shift-register sequence code for single errors, column vectors can be expressed as T, xT, $x^2T \cdots x^nT$, where the transformation matrix x satisfies the equation

$$cx^m + c_{m-1}x^{m-1} \cdots c_1x + 1 = 0\,, \qquad (2)$$

where 1 represents the identity matrix.

Polynomial (2) is chosen primitive, thus $x^{2m-1}+1=0$, and the vectors form a cyclic group of $n=2^{m-1}$ elements. A large number of x matrices will satisfy Eq. (2), each giving rise to a different check matrix. Also, the choice of T is arbitrary. For simplest instrumentation, x may be chosen to be the Associated Matrix, as in the Abramson SEC-DAEC Code.[1]

For the purposes of this discussion, the code will be defined by Eq. (2) only, since the code properties are independent of the choice of x.

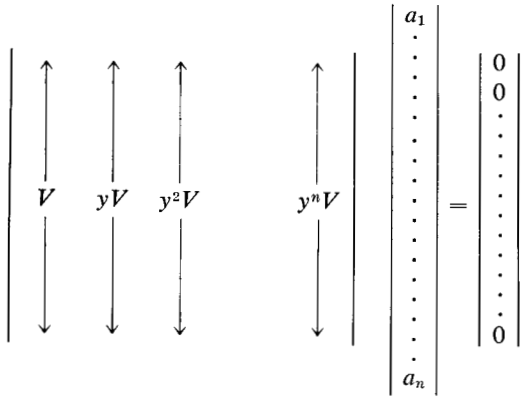
Let us now consider how the check matrix could be expanded for the correction of multiple errors. If more than one error is present, the corrector assumes the value of the sum of the vectors corresponding to the bits in error. For example, an error in bits D_1 and D_2 will result in the corrector vector $T+xT=(x+1)T$.

This vector is identical to a corrector for a single error occurring p bits after the first bit in error.

$$(x+1)T = x^pT \text{ or } x^p+x+1=0\,. \qquad (3)$$

The value of p can be found by multiplying polynomial (2) by a polynomial $Q(x)$, such that $(x^m+c_{m-1}x^{m-1} \cdots c_1x+1)Q(x)=x^p+x+1$. This value is independent of the error position in the message. We will define p as the sequence shift corresponding to an $x+1$ error pattern, since the corrector sequence for the $x+1$ error pattern is the same as the sequence for single errors but shifted by p positions. All other multiple error patterns producing a non-zero corrector can be characterized by their respective sequence shifts, and can be distinguished from the single errors by adding a second deck matrix derived from another primitive polynomial $F(y)$.

$$\begin{vmatrix} \uparrow & \uparrow & \uparrow & & \uparrow \\ T & xT & x^2T & \cdots & x^nT \\ \downarrow & \downarrow & \downarrow & & \downarrow \end{vmatrix} \cdot \begin{vmatrix} a_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{vmatrix}$$



The total message length is still n so that the allowable information bits have been reduced to accommodate the additional parity bits.

The correctors for single errors consist of n vectors with the components of x^jT and y^jV. Consider now a multiple error pattern with a sequence shift of p for one matrix and q for the other. The corresponding correctors will assume the components of the vectors $x^{j+p}T$ and $y^{j+q}V$ and if the relative shift $p-q$ is not 0 a new set of correctors is generated. Disjoint sets of correctors will exist for each error pattern with a different value of $p-q$, mod n. If both matrices have the same dimensions, 2^m+1 different error patterns can be corrected. The addition of an all-check parity bit will allow correction of $2^{2m}+2$ different multiple error patterns.

The primitive polynomials to be matched need not be of the same order, as long as the vector cycle-set periods are multiples of each other. The number of correctible error patterns is then equal to $1+\gamma$, where γ is the cycle set period of the smaller set. The code is now defined by two primitive polynomials or, as Meggitt has shown, by a single polynomial with two primitive roots. Burst correction properties of these codes were previously investigated by the author.[3]

Double error correction

Theorem

A code defined by any primitive polynomial $P(x)$ and its inverse $P^*(x)$ will correct all double errors. If $P(x)$ is of even order, an all-check parity bit must be added for single error correction.

An inverse polynomial $P^*(x)$ will be defined as follows:

$$\text{If } P(x)=x^m+c_{m-1}x^{m-1}+\cdots\cdot c_1x+1=0\,, \tag{4}$$

then

$$P(x^{-1})=x^{-m}+c_{m-1}x^{-(m-1)}+\cdots\cdot c_1x^{-1}+1=0\,. \tag{5}$$

Multiplying by x^m:

$$x^mP(x^{-1})=P^*(x)=1+c_{m-1}x+c_{m-2}x^2\cdots c_1x^{m-1}+x^m=0\,. \tag{6}$$

If matrix x satisfies $P(x)=0$, the inverse matrix x^{-1} will satisfy $P^*(x)=0$, and the vectors x^jT of one check matrix will follow the opposite sequence to the vectors $x^{-j}V$ in the other check matrix. $P^*(x)$ is primitive also since the sequence lengths are the same as for $P(x)$.

Proof

If the code is to correct any two errors in a block 2^n-1 bits long, a different relative sequence shift $p-q$ must exist for every error pattern of the type $1+x^r$, where $r\leq 2^{m-1}-1$. No larger values of r need be considered because of the cyclic nature of the code. The first and last bits of the message are treated as adjacent bits.

If the sequence shift corresponding to the $1+x^r$ error is p for polynomial $P(x)$, it follows, as was previously stated, that x^p+x^r+1 is divisible by $P(x)$. But then $x^{-p}+x^{-r}+1$ is divisible by $P^*(x)$. Since $x^r(x^{-p}+x^{-r}+1)=x^{r-p}+1+x^r$ is also divisible by $P^*(x)$, the sequence shift for $P^*(x)$ is by definition $q=r-p$, for the $1+x^r$ error. The relative shift is therefore $p-q=2p-r$.

We require $p-q$ to be different for every r. Algebraically, if

$$1+x^r=x^p\,, \tag{7}$$

$$\text{then}\quad 1+x^{r+2k}\neq x^{p+k} \tag{8}$$

for all r and k such that $r+2k\leqslant 2^{m-1}-1$. This condition is necessary and sufficient for the code to correct all $1+x^r$ errors ($r\leqslant 2^{m-1}-1$). If (8) is not satisfied for a particular value of k, there will be no distinction between the $1+x^r$ error and the $1+x^{r+2k}$ error. Certainly $2p-r$ is different for any odd and any even value of r, and consequently Ineq. (8) is a sufficient condition for every possible double error.

We will assume now that the opposite is true, and

$$1+x^{r+2k}=x^{k+p} \tag{9}$$

and show that this set of polynomials is not divisible by any primitive polynomial of order m, and cannot describe any relation between vectors in the check matrix. Eliminating p, between (7) and (9)

$$x^{r+2k}+x^{r+k}+x^k+1=0\,. \tag{10}$$

Equation (10) can be factored to:

$$(1+x^k)(1+x^{r+k})\,. \tag{11}$$

Equation (11) is divisible by a primitive polynomial of order m if and only if: $r+k\geqslant 2^m-1$, which is impossible since $r+2k\leqslant 2^{m-1}-1$. Therefore, Eq. (9) does not hold, and every double error has a distinct set of correctors.

If single errors are to be corrected $p-q=2p-r\neq 0$ in the equation $1+x^r=x^p$, since 0 is the relative shift for single errors. It will be shown that this condition is always met if $P(x)$ is of odd order.

Assume the opposite is true and $2p-r=0$. This would yield the equation

$$1+x^r+x^{r/2}=0\,. \tag{12}$$

If we change the bounds of r to $r\leqslant 2^m-2$, only even values of r need be considered; setting $2t=r$

$$x^{2t}+x^t+1=0\,. \tag{13}$$

Multiplying (13) by $1+x^t$, gives the equivalent equation

$$x^{3t}+1=0\,, \qquad (14)$$

where $t\leqslant 2^{m-1}-1$. Equation (14) is divisible by a primitive polynomial of order m if and only if its cycle set period 2^m-1 is divisible by 3. This is the case if and only if m is even.

The codes just described will correct all double and single errors in a block of 2^m-1 bits. If m is odd, $2m$ parity bits are needed; if m is even, $2m+1$ parity bits are required.

An example of this code will be shown here using the polynomials:

$P(x)=x^5+x^2+1$ and

$P(y)=y^5+y^3+1=P^*(x)$.

The x and y matrices will be chosen as follows:

$$x=\left|\begin{array}{ccccc} 0&0&1&0&1\\ 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0 \end{array}\right| \qquad y=x^{-1}=\left|\begin{array}{ccccc} 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 1&0&0&1&0 \end{array}\right|$$

Let $T=V=\{10000\}$. The check matrix is:

1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0 0 0 0
0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0 0 0
0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0 0
0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0
0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1

1 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0
0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1
0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0
0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0 0
0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0 0 0

In this example, $P(x)=x^5+x^2+1$, does not divide into any of the polynomials (12) or (13) for any $r<15$, and single errors are corrected without an additional parity bit.

On the other hand $P(x)=x^4+x+1$ is a factor of the polynomial $x^{10}+x^5+1=0$ and an all-check parity bit is necessary for the code defined by $P(x)$ and $P^*(x)=x^4+x^3+1$, to distinguish a single error from an $1+x^5$ double error.

Codes for arbitrary Hamming distance

The Hamming or minimum distance of a code can be defined as the minimum number of errors in a message that is interpreted by that code as a correct message. A code with Hamming distance $2d+1$ will correct any t errors in the message.

The class of codes under discussion is defined by the equation $R(x)=0$, where $R(x)$ is the product of two or more primitive polynomials. The equation $R(x)=0$, or in general $Q(x)R(x)=0$, where $Q(x)$ is any polynomial, describes the vectors in the check matrix that add up to the null vector. Since any x^jT vector in the matrix is a corrector for a single error in bit a_j, the number of terms in the polynomials corresponds to the number of bits in error yielding a null corrector. In particular, the minimum number of terms of any polynomial $Q(x)R(x)$, is the Hamming distance for that code providing the order of that polynomial is less than the message length.

Consider for example, a code derived from the primitive polynomial $R(x)=x^4+x+1$. The Hamming distance, according to the previous definition, is 3, for no polynomial $Q(x)$ will make $Q(x)R(x)$ contain less than three terms, and be less than order 15. The code is a single-error correcting code as previously described. Any primitive polynomial code has minimum distance of 3, for there always exists a "p" less than the cycle length, for which x^p+x+1 is divisible by that polynomial. No two-term polynomial of order less than the cycle length is divisible by a primitive polynomial by definition.

The code described in a previous example $R(x)=P(x)\cdot P^*(x)=(x^5+x^2+1)(x^5+x^3+1)=0$ or $x^{10}+x^8+x^7+x^5+x^3+x^2+1=0$, has a minimum distance of 5. Multiplication of $R(x)$ by $Q(x)=x^5+x^3+x^2+x+1$, yields $R(x)Q(x)=x^{15}+x^7+x^3+x+1$, a polynomial with 5 terms. No other polynomial $Q(x)$ will result in fewer terms for $R(x)Q(x)$, if $R(x)Q(x)$ is less than order 31.

The same criterion applies to the Bose-Chandhuri codes, or more precisely to their generalization in terms of a single polynomial described by Peterson.[7]

References

1. N. M. Abramson, "A Class of Systematic Codes for Non-Independent Errors." *IRE Transactions PGIT* **IT-5,** 150 (1959).
2. P. Fire, "A Class of Multiple Error-correcting Binary Codes for Non-Independent Errors." Report RSL-E2, Sylvania Corporation, Mountain View, California.
3. C. M. Melas, "A New Group of Codes for Correction of Dependent Errors in Data Transmission." *IBM Journal of Research and Development,* **4,** 58 (January 1960).
4. S. M. Reiger, "Codes for the Correction of 'Clustered' Errors." Rand Corporation report P-1677, April 1959.
5. R. C. Bose and D. N. Ray-Chandhuri, "On a Class of Error Correcting Binary Group Codes." *Information and Control,* **3,** 68 (March 1960).
6. J. E. Meggitt, "Error Correcting Codes for Correcting Bursts of Errors." This issue, p. 329.
7. W. W. Peterson, "Cyclic Codes." Chapter VII of a monograph on error correcting codes.

Received May 10, 1960