10-1970

SOME RESULTS ON THE DISTANCE PROPERTIES OF CONVOLUTIONAL CODES

Luther D. Rudolph
Syracuse University

Alexander Miczo
Syracuse University

SOME RESULTS ON THE DISTANCE PROPERTIES OF CONVOLUTIONAL CODES

LUTHER D. RUDOLPH

ALEXANDER MICZO

OCTOBER 1970

SYSTEMS AND INFORMATION SCIENCE
SYRACUSE UNIVERSITY

SOME RESULTS ON THE DISTANCE PROPERTIES OF CONVOLUTIONAL CODES

LUTHER D. RUDOLPH

ALEXANDER MICZO

This work was supported by the National Science Foundation under Grant GK-4737.

SYSTEMS AND INFORMATION SCIENCE

SYRACUSE UNIVERSITY

SYRACUSE, NEW YORK 13210

(315) 476-5541 EXT. 2368

ABSTRACT

Rate 1/2 binary convolutional codes are analyzed and a lower bound on free distance in terms of the minimum distances of two associated cyclic codes is derived. Next, the complexity of computing the free distance is discussed and a counterexample to a conjecture on the relationship of row distance to free distance for systematic codes is presented. Finally, an improved Gilbert bound for definite decoding is derived.

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

This report describes the results of an investigation of the distance properties of convolutional codes. The result of this effort, which developed along three relatively independent lines, are contained in Sections, 2, 3 and 4. These sections are independent of one another and relatively self-contained. It is assumed that the reader is familiar with the basic ideas of convolutional codes, say at the level of Lin's "Introduction to Error-Correcting Codes." (Prentice-Hall, 1970).

Section 2 contains an analysis of rate 1/2 binary convolutional codes. The attempt here was to find new ways to characterize the very simplest class of convolutional codes, with an eye to developing algebraic machinery that could be used to construct good codes. The major result in this section is a new lower bound on the free distance of a rate 1/2 binary convolutional code in terms of the minimum distances of two associated cyclic codes.

In Section 3, the complexity of computing the free distance of a systematic convolutional code is discussed. Previously known results on the relationship of row and column distance to free distance are summarized, and a new negative result on ^{row} distance is presented.

Section 4 is concerned with deriving an improved Gilbert lower bound for definite decoding of convolutional codes. Massey's bound is discussed and a tighter bound is obtained.

SECTION 2

ANALYSIS OF RATE 1/2 BINARY CONVOLUTIONAL CODES

Many of the most impressive advances in the theory of block codes, such as the development of the BCH codes and their decoding algorithm, have been a direct result of viewing the coding problem in the appropriate mathematical setting. There have been relatively few such advances in the theory of convolutional codes, and there is a general feeling among researchers that the appropriate algebraic framework has yet to be found. In this section, we present the results of an effort to find new ways to look at convolutional codes. We restricted our attention to rate 1/2 binary convolutional codes because this is both the most easily analyzed case and because these codes comprise the single most important class of convolutional codes from an applications point of view. First, we consider mathematical models of the encoder for a rate 1/2 binary convolutional code; three of these models are well known, one new. We next impose restrictions to eliminate bad codes from consideration. Finally, we derive a new lower bound on the free distance of codes in this class.

2.1 Parallel mathematical models

An encoder for a rate 1/2 binary convolutional code is any information-lossless 1-input, 2-output linear sequential machine as illustrated in Figure 2-1. Without

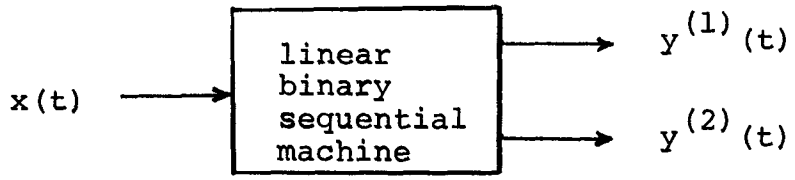


Figure 2-1. Parallel encoder for a rate 1/2 code

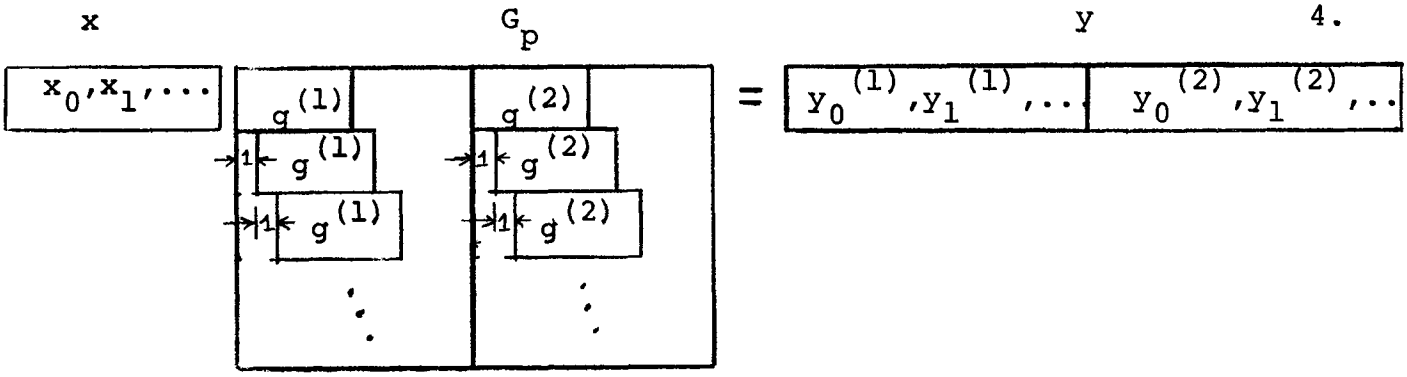
essential loss of generality[1], we will consider only polynomial (loop-free) encoders.

The encoder maps an information sequence $x(t)$ onto the pair of code sequences $(y^{(1)}(t), y^{(2)}(t))$. The code is defined to be the set of all possible pairs of code sequences that can be generated by the encoder. The encoding mapping is a linear transformation, with memory, over GF(2), the field of two elements. It may thus be conveniently described in terms of delay polynomials. If $x(D)$ is the D-transform of the input sequence and $y^{(1)}(D)$ and $y^{(2)}(D)$ are the D-transforms of the code sequences, then the encoding mapping may be described by the polynomial equation

$$[x(D)][g^{(1)}(D),\ g^{(2)}(D)] = [y^{(1)}(D),\ y^{(2)}(D)]$$

where $g^{(i)}(D) = g_0^{(i)} + \ldots + g_m^{(i)} D^m$, $i = 1,2$ are the generator polynomials of the code. (We take m to be the maximum of the degrees of $g^{(1)}(D)$ and $g^{(2)}(D)$; m is called the memory order of the code.)

An equivalent description in the time domain is given by



where $g^{(i)} = (g_0^{(i)}, g^{(i)}, \ldots, g_m^{(i)})$, $i = 1,2$. We may write this in the simpler form

$$x(G^{(1)}; G^{(2)}) = y, \text{ or even } xG_p = y.$$

The matrix $G_p = (G^{(1)}; G^{(2)})$ is called the generator matrix of the code.

2.2 Serial mathematical models

A parallel encoder can be converted to a serial encoder by addition of a switch that alternates between $y^{(1)}(t)$ and $y^{(2)}(t)$ as shown in Figure 2-2.

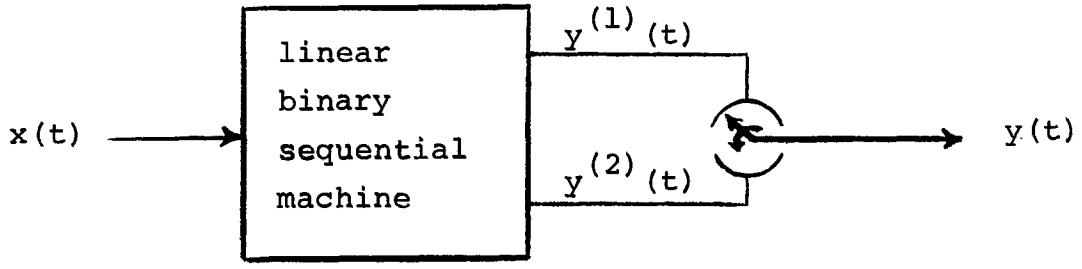
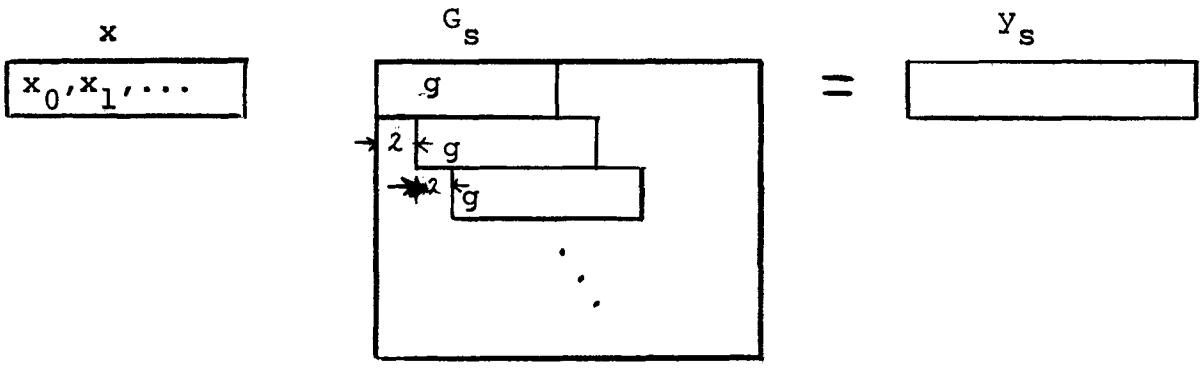


Figure 2-2. Serial encoder for a rate 1/2 code

A time domain description of the serial encoder is obtained by permuting the columns of G_p and y_p in the time domain description of the parallel encoder. This gives

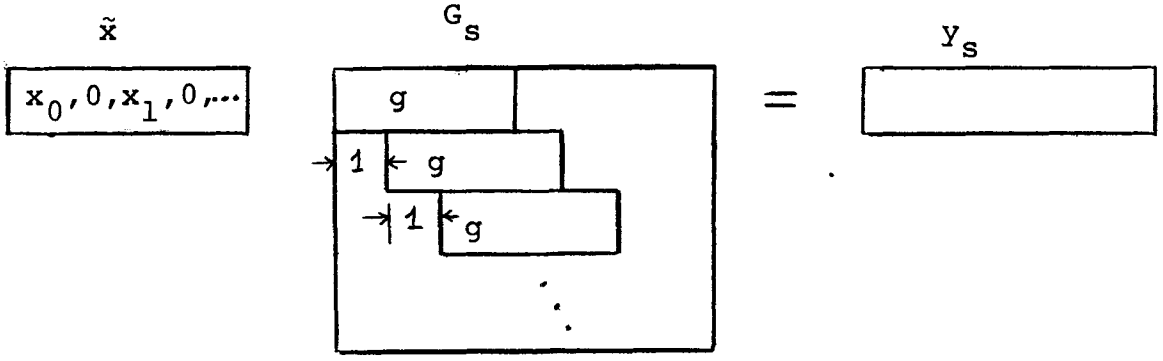


where $g = (g_0^{(1)}, g_0^{(2)}, g_1^{(1)}, g_1^{(2)}, \ldots)$

and $y_s = (y_0^{(1)}, y_0^{(2)}, y_1^{(1)}, y_1^{(2)}, \ldots)$.

The three descriptions given so far - the parallel polynomial and matrix descriptions and the serial matrix description - are all standard forms that appear in the literature. We next present a fourth possibility that we feel may be useful: a serial polynomial description.

Our intention is to convert the serial matrix description of the encoding function into an equivalent serial polynomial description. We can't do this directly because adjacent rows of G_s are shifted two time units with respect to one another. This is easily remedied by writing the matrix description in the modified form



The D-transform equivalent is

$$\tilde{x}(D)g(D) = y_s(D)$$

where $\tilde{x}(D) = x(D^2)$

and $$g(D) = g^{(1)}(D^2)+Dg^{(2)}(D^2)$$

$$y_s(D) = y^{(1)}(D^2)+Dy^{(2)}(D^2)$$

or, since $f(x^2) = f^2(x)$ for any polynomial f over a field of characteristic 2,

$$x^2(D)g(D) = y_s(D)$$

where

$$g(D) = g^{2(1)}(D) + Dg^{2(2)}(D)$$

$$y_s(D) = y^{2(1)}(D) + Dy^{2(2)}(D).$$

This says that a rate 1/2 binary code with unrestricted input is equivalent to a rate 1 binary code where the input sequences are required to be squares. We might thus view the encoder as a squaring circuit followed by g(D) as shown in Figure 2-3(a). An alternative model results from the observation that the derivative $x'(D) = \frac{d}{dD}x(D)$ of any delay polynomial x(D) over GF(2) is a square, and any square is the derivative of some delay polynomial. This allows us to view the encoder as a differentiating circuit followed by g(D) as shown in Figure 2-3(b).

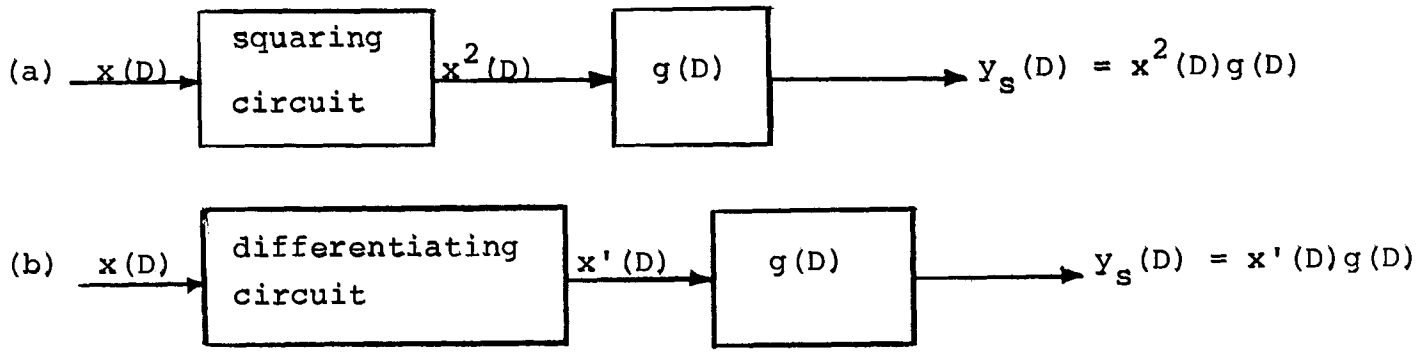


Figure 2-3. Serial encoder models

For the remainder of this section, we will use only the serial polynomial description of the encoding function, i.e. either

$$\text{(a)} \quad x^2(D)g(D) = y(D) \quad \text{or}$$
$$\text{(b)} \quad x'(D)g(D) = y(D).$$

(We will drop the subscript on y from this point on.) Our hope, of course, is that this new description will shed some light on the problem of selecting those polynomials g(D) that generate "good" codes. As a first step, we consider the possibility of placing restrictions on g(D) to eliminate "bad" codes from consideration.

2.3 Restrictions to eliminate bad codes

One class of codes we surely wish to eliminate from consideration are those subject to catastrophic error propagation. As shown by Massey and Sain[2], a rate 1/2 convolutional code with generator polynomials $g^{(1)}(D)$ and $g^{(2)}(D)$ will not be subject to catastrophic error propagation if and only if G.C.D.$(g^{(1)}(D), g^{(2)}(D)) = D^k$ for some k. We will always assume that either $g^{(1)}(D)$ or $g^{(2)}(D)$ has a nonzero constant term, in which case k = 0. The condition we wish to impose then is that $g^{(1)}(D)$ and $g^{(2)}(D)$ be relatively prime. The only question is: what form does this restriction take in the serial polynomial description?

Proposition 2-1. G.C.D.$(g^{(1)}(D), g^{(2)}(D)) \neq 1$ if and only if G.C.D.$(g(D), g'(D)) \neq 1$.

(Proof) Note that

$$g(D) = g^{(1)}(D^2) + Dg^{(2)}(D^2)$$

$$g'(D) = g^{(2)}(D^2).$$

Now suppose that G.C.D.$(g^{(1)}(D), g^{(2)}(D)) \neq 1$.
Then there is a nonzero α such that $g^{(1)}(\alpha) = g^{(2)}(\alpha) = 0$.
But then $g(\alpha) = g'(\alpha) = 0$ and G.C.D.$(g(D), g'(D)) \neq 1$. The converse follows easily by the same sort of argument.

So we conclude that the code will not be subject to catastrophic error propagation if and only if we impose

Restriction 1: G.C.D.$(g(D), g'(D)) = 1$.

We would like also to eliminate from consideration codes which have poor distance properties. In order to discuss this, we must first define the measure of "goodness" we plan to use. The free distance, d_{free}, of the convolutional code generated by $g(D)$ is the minimum Hamming weight taken over all nonzero code words $y(D) = x^2(D)g(D)$, i.e.

$$d_{free} = \min_{x(D) \neq 0} W_H(x^2(D)g(D)).$$

The constraint length of a rate K/N convolutional code with memory order m is defined to be $n_A = N(m+1)$. In our case, N = 2, so $n_A = 2(m+1)$. Our goodness measure is then the ratio of free distance to constraint length: $d_{free}/2(m+1)$. (This is roughly the ratio of free distance to the degree of $g(D)$, since the degree of $g(D) = g^{(1)}(D^2) + Dg^{(2)}(D^2)$ is either 2m or 2m+1.)

From the standpoint of distance properties, the worst codes (other

than the trivial null code) are those whose generators are squares. For suppose $g(D) = f^2(D)$. Take $x(D) = (D^e+1)/f(D)$ where e is the exponent of $f(D)$ (i.e. the smallest integer such that $f(D)$ divides D^e+1). Then the code word corresponding to $x(D)$ is

$$y(D) = x^2(D)g(D) = \left(\frac{D^e+1}{f(D)}\right)^2 \cdot f^2(D) = D^{2e}-1$$

which has weight 2. Hence $d_{free} \leq 2$ for any code whose generator is a square. This suggests that perhaps we should consider only codes whose generators are as "far" from being square as possible, i.e. generators that are squarefree in the sense that they have no repeated roots, and hence no squared factors. This turns out to be the case as we show.

Proposition 2-2. If $g(D) = f^2(D)h(D)$, then $g(D)$ and $h(D)$ generate the same code.

(Proof) Let $y(D)$ be any code word in C_g, the code generated by $g(D)$. Then

$$\begin{aligned} y(D) &= x^2(D)g(D) \\ &= x^2(D)f^2(D)h(D) \\ &= (x(D)f(D))^2 h(D) \end{aligned}$$

which is a code word in C_h, the code generated by $h(D)$. Conversely, let $y(D)$ be any code word in C_h. Then

$$y(D) = x^2(D)h(D)$$

$$= \frac{x^2(D)f^2(D)h(D)}{f^2(D)}$$

$$= \left(\frac{x(D)}{f(D)}\right)^2 \cdot g(D)$$

which is a code word in C_g.

Although g(D) and h(D) both generate codes with the same d_{free}, the memory order associated with h(D) is less than that associated with g(D) (unless $f^2(D) = 1$). Hence $d_{free}/2(m+1)$ is greater for C_g and we are justified in imposing

Restriction 2: g(D) is squarefree.

We next show that restrictions 1 and 2 are equivalent.

Proposition 2-3. G.C.D.(g(D),g'(D)) $\neq 1$ if and only if g(D) is squarefree.

(Proof) Suppose G.C.D.(g(D),g'(D)) $\neq 1$. Then $g^{(1)}(D)$ and $g^{(2)}(D)$ have a common factor by proposition 2-1, i.e.

$$g^{(1)}(D) = f(D)\ \tilde{g}^{(1)}(D)$$

$$g^{(2)}(D) = f(D)\ \tilde{g}^{(2)}(D).$$

Then

$$g(D) = \tilde{g}^{2(1)}(D) + D\tilde{g}^{2(2)}(D)$$

$$= f^2(D)\tilde{g}^{2(1)}(D) + Df^2(D)\tilde{g}^{2(2)}(D)$$

$$= f^2(D)\tilde{g}(D).$$

Conversely, suppose $g(D)$ is not squarefree. Then

$$g(D) = f^2(D)h(D).$$

$$g'(D) = f^2(D)h'(D).$$

But

$$g(D) = g^{2(1)}(D) + Dg^{2(2)}(D)$$

$$g'(D) = g^{2(2)}(D)$$

thus

$$f^2(D)h(D) = g^{2(2)}(D) + Dg^{2(2)}(D)$$

$$f^2(D)h'(D) = g^{2(2)}(D)$$

and it follows that $f(D)$ divides both $g^{(1)}(D)$ and $g^{(2)}(D)$ so that G.C.D.$(g^{(1)}(D), g^{(2)}(D)) \neq 1$ by proposition 2-1.

We thus arrive, from quite different starting points, at the same restricted class of codes, namely the "squarefree codes" whose generators have no repeated roots. The problem of catastrophic error propagation has been eliminated without the loss of any "good" codes. (Also see Forney[1].)

2.4 A lower bound on d_{free}

We conclude this section by deriving a lower bound on the free distance of a squarefree rate 1/2 binary convolutional code. First we establish notation and list some elementary properties.

Let $a(x) = \sum_i a_i x^i$ be any polynomial over GF(2). Then

1. $|a(x)|$ is the degree of $a(x)$.
2. $||a(x)||$ is the Hamming weight of $a(x)$.
3. $a(x)b(x)$ is the usual polynomial product.
4. $a(x) \circ b(x) = \sum a_i b_i x^i$ is the component-by-component product.

Property 1. $||a(x)x^n|| = ||a(x)||$.

Property 2. $||a(x) \circ b(x)|| \leq \min (||a(x)||, ||b(x)||)$.

Property 3. $$||\sum_j a_j(x)|| = \sum_j ||a_j(x)|| - 2 \sum_{i \neq j} ||a_i(x) \circ a_j(x)||$$

$$+ 4 \sum_{i \neq j \neq k \neq i} ||a_i(x) \circ a_K(x)|| - \ldots$$

(Proof by inclusion-exclusion)

Property 4. $||a(x)|| - ||a(x) \circ b(x)|| = ||a(x) \circ (1(x) + b(x))||$.

Property 5. $||a(x)|| - ||a(x) \circ b(x)|| \geq ||a(x) \circ c(x)|| - ||a(x) \circ b(x) \circ c(x)||$.

Property 6. If $n > |b(x)|$, then $||a(x)(x^n + 1) + b(x)|| > ||b(x)||$.

(Proof) By property 3,

$$||a(x)(x^n+1)+b(x)|| = ||a(x)|| + ||x^n a(x)|| + ||b(x)|| - 2||a(x) \circ x^n a(x)|| - 2||a(x) \circ b(x)|| - 2||x^n a(x) \circ b(x)|| + 4||a(x) \circ x^n a(x) \circ b(x)||.$$

But if $n > |b(x)|$, then $x^n a(x) \circ b(x) = 0$. Noting that

$||x^n a(x)|| = ||a(x)||$ by property 1, we then have

$$||a(x)(x^n+1)+b(x)||-||b(x)|| = 2(||a(x)||-||a(x)\circ x^n a(x)||-||a(x)\circ b(x)||).$$

But $x^n a(x)$ and $b(x)$ are disjoint, from which it follows that

$$||a(x)\circ x^n a(x)||+||a(x)\circ b(x)|| \leq ||a(x)||.$$

Property 7. If $n > |b(x)|$, then $||(x^n+1)^i(a(x)(x^n+1)+b(x))|| \geq ||b(x)||$ $i=1,2,\ldots,$.

(Proof) Case I: i odd. Then $i = 2k+1$ and

$$||(x^n+1)^{2k+1}(a(x)(x^n+1)+b(x))||=||(x^{(2k+2)n}+1)a(x)+(x^n+1)^{2k+1}b(x)||$$

But $|(x^n+1)^{2k+1}b(x)| < (2k+2)n$ since $|b(x)| < n$. Hence, by property 6,

$$||(x^{(2k+2)n}+1)a(x)+(x^n+1)^{2k+1}b(x)|| \geq ||(x^n+1)^{2k+1}b(x)|| \geq ||b(x)||.$$

Case II: i even. Then $i = 2k$ and

$$||(x^n+1)^{2k}(a(x)(x^n+1)+b(x))|| = ||(x^n+1)^{2k+1}a(x)+(x^{2kn}+1)b(x)||.$$

We can express $a(x)$ as

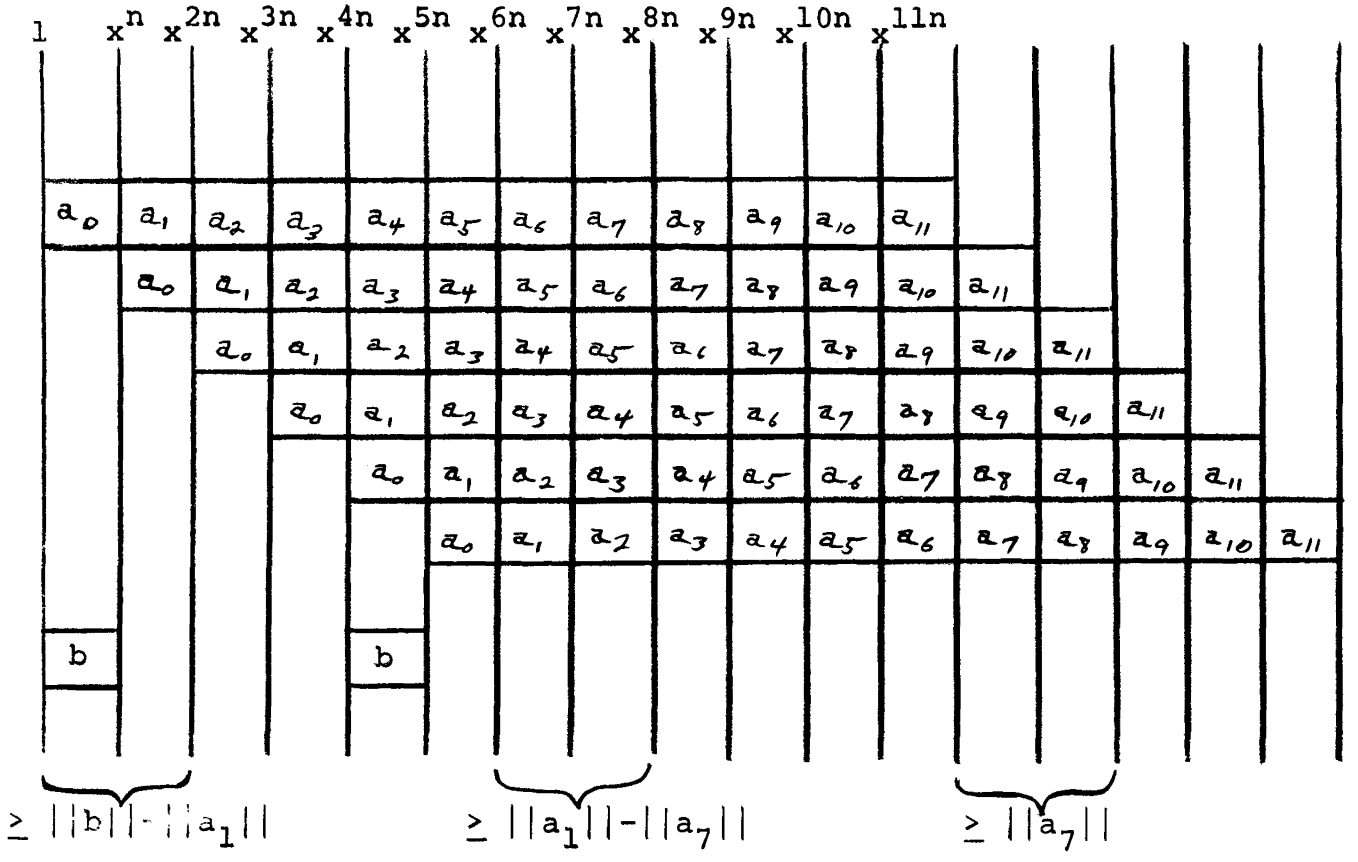
$$a(x) = \sum_{i=0}^{\ell n} x^{in} a_i(x) \qquad \text{where } |a_i(x)| < n.$$

Then we can write the above as

$$||(x^n+1)^{2k+1}\sum_{i=0}^{\ell n} x^{in}a_i(x) + (x^{2kn}+1)b(x)||$$

which is shown pictorially in Figure 2-4 for $k=2$ and $\ell=11$. The argument goes as follows: The weight contribution from the initial two

"blocks" is at least $||b(x)||-||a_1(x)||$ by property 6. Then we look at the two consecutive blocks with $a_1(x)$ in the lower left corner. The weight contribution from this pair of blocks is at least $||a_1(x)||-||a_{2k+3}(x)||$. Next, we look at the pair of blocks with $a_{2k+3}(x)$ in the lower left corner, and so forth. We continue until the upper right corner of a pair of consecutive blocks is the zero polynomial. Then we add up the weight contributions as shown in Figure 2-4, and everything cancels except $||b(x)||$.



$$||(x^n+1)^5\, a(x) + (x^{4n}+1)b(x)|| \geq ||b||-||a_1||+||a_1||-||a_7||+||a_7|| = ||b||$$

Figure 2-4. Visual aid for the proof of property 7.

We are now in a position to prove the theorem. Let d_g denote the minimum distance of the cyclic code generated by $g(D)$ and d_h the minimum distance of the dual code generated by $h(D) = (D^e+1)/g(D)$ where e is the exponent of $g(D)$.

Theorem 2-1. Let d_{free}, d_g and d_h be the convolutional and cyclic code distances associated with a generator $g(D)$.
Then

$$d_{free} \geq \min(d_g, 2d_h).$$

(Proof) Let $y(D)$ be any code word in the convolutional code generated by $g(D)$. Then

$$y(D) = x^2(D)g(D)$$

for some information polynomial $x(D)$. We can always write $x^2(D)$ as

$$x^2(D) = f^2(D)(D^e+1)^{2i}$$

where $i \geq 0$ is chosen so that $f(D)$ is not divisible by D^e+1. Then

$$y(D) = f^2(D)g(D)(D^e+1)^{2i}.$$

Case I: $f(D)$ not divisible by $h(D)$.
In this case we can write

$$f^2(D) = p(D)h(D)+r(D) \quad \text{where } r(D)\neq 0 \text{ and } |r(D)|<|h(D)|$$

But $|\tilde{r}(D)h(D)(D^e+1)| < 2e$, since $|\tilde{r}(D)| < |g(D)|$.
Then, by property 7,

$$||y(D)|| \geq ||\tilde{r}(D)h(D)(D^e+1)||.$$

But $\tilde{r}(D)h(D)$ is a nonzero word in the cyclic code generated by $h(D)$ and hence has weight at least d_h. Then $\tilde{r}(D)h(D)(D^e+1)$ must have weight at least $2d_h$, since $|\tilde{r}(D)h(D)|<e$.

In case I, $||y(D)|| \geq d_g$ and in Case II, $||y(D)|| \geq 2d_h$.
Since $y(D)$ was arbitrary, we conclude that $d_{free} \geq \min(d_g, 2d_h)$.

Example: Suppose we take $g(D)$ to be the generator of the (31,11) BCH code. This code has minimum distance $d_g = 11$, and its dual code, the (31,20) code, has minimum distance $d_h = 6$. Then $g(D)$ generates a rate 1/2 convolutional code with free distance

$$d_{free} \geq \min(d_g, 2d_h) = \min(11,12) = 11.$$

Since the degree of $g(D)$ is 20, the convolutional code generated by $g(D)$ has memory order 10, and the measure of goodness is

$$d_{free}/2(m+1) \geq 11/22 = .5$$

REFERENCES FOR SECTION 2

1. G.D. Forney, "Algebraic Structure of Convolutional Codes," presented at the 1969 International Symposium on Information Theory, Ellenville, New York.

2. J.L. Massey and M.K. Sain, "Inverses of Linear Sequential Circuits," IEEE Trans. on Electronic Computers, C-17, 330-337 (1968).

SECTION 3

ON THE COMPLEXITY OF COMPUTING D_{free}

In this section, we consider the problem of computing the free distance of a systematic rate 1/N convolutional code. In the general case, there is at present no attractive alternative to generating all code words corresponding to information sequences of length, 1,2,...,L+1, where L is sufficiently large to ensure that at least one code word of weight d_{free} has been generated. Whatever measure we choose, it is clear that the complexity of the computation is highly dependent on the parameter L. In what follows, we summarize the previously known results on L and its associate L*, and present some new results and conjectures based on partial results.

3.1 Rate 1/N systematic convolutional codes

An encoder for a rate K/N convolutional code is any information-lossless K-input, N-output linear sequential machine. For our purposes, it is sufficient to consider the subclass of rate 1/N systematic convolutional codes. The encoder then takes the form shown in Figure 3-1. We will consider only polynomial encoders, i.e. encoders with no internal feedback loops. This entails no real loss of generality and simplifies the analysis.

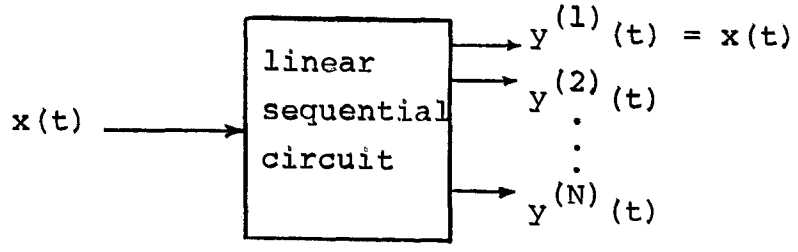


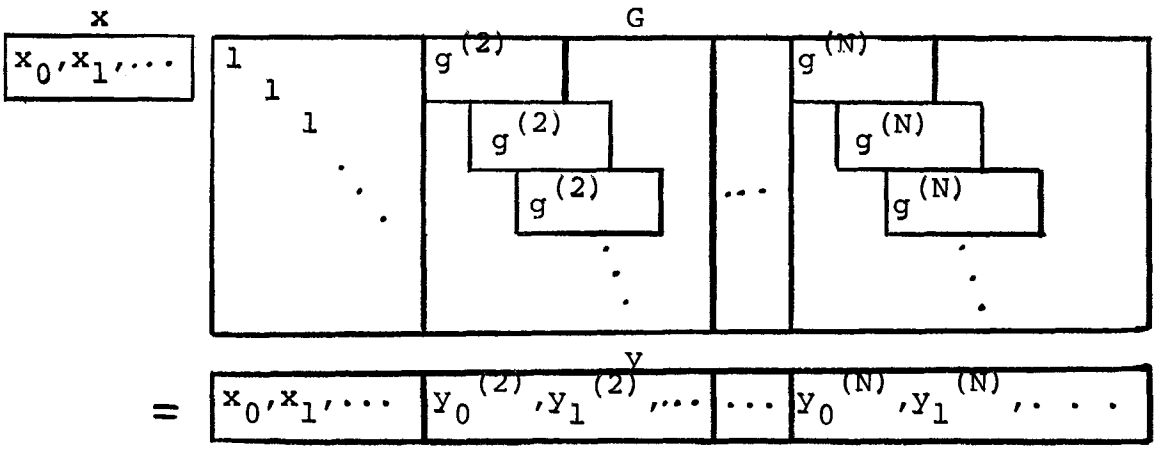
Figure 3-1. Encoder for a rate 1/N systematic convolutional code

The encoder maps an information sequence $x(t)$ onto an N-tuple of code sequences $(x(t), y^{(2)}(t), \ldots y^{(N)}(t))$. The encoding function is thus a linear transformation with memory over GF(q), the field of q elements. This mapping can be described conveniently either as a matrix of finite dimension over GF(q)[D], the ring of delay polynomials over GF(q), or as a matrix of infinite dimension over GF(q) with time as an explicit parameter. In the delay polynomial domain, the encoding function for a rate 1/N systematic convolutional code can be written as

$$[x(D)][1, g^{(2)}(D), \ldots, g^{(N)}(D)] = [x(D), y^{(2)}(D), \ldots, y^{(N)}(D)]$$

where $g^{(i)}(D) = g_0^{(i)} + g_1^{(i)} D + \ldots + g_m^{(i)} D^m$ is the <u>generator polynomial</u> that relates $x(D)$, the D-transform of the input sequence, to $y^{(i)}(D)$, the D-transform of the i^{th} code sequence. The parameter m, the maximum degree among the generator polynomials (we can consider all the generator polynomials to be of degree m by allowing high order coefficients to be zero) is called the <u>memory order</u> of the code.

An equivalent description in the time domain is given by



where $g^{(i)} = (g_0^{(i)}, g_1^{(i)}, \ldots, g_m^{(i)})$. It is convenient to write this

simply as

$$xG = y.$$

The matrix G is commonly referred to as the generator matrix of the code. The time domain description of the encoding function will be used in the remainder of this section.

Let $d_H(x,y)$ denote the Hamming distance between x and y, and $W_H(x) = d_H(x,0)$ the Hamming weight of x. Then the free distance of a convolutional code with generator matrix G is defined to be

$$d_{free} = \min_{x \neq x'} d_H(xG,x'G)$$

or, since the code is linear,

$$d_{free} = \min_{x \neq 0} W_H(xG).$$

Without loss of generality, we may take $x_0 \neq 0$, which allows a third equivalent definition:

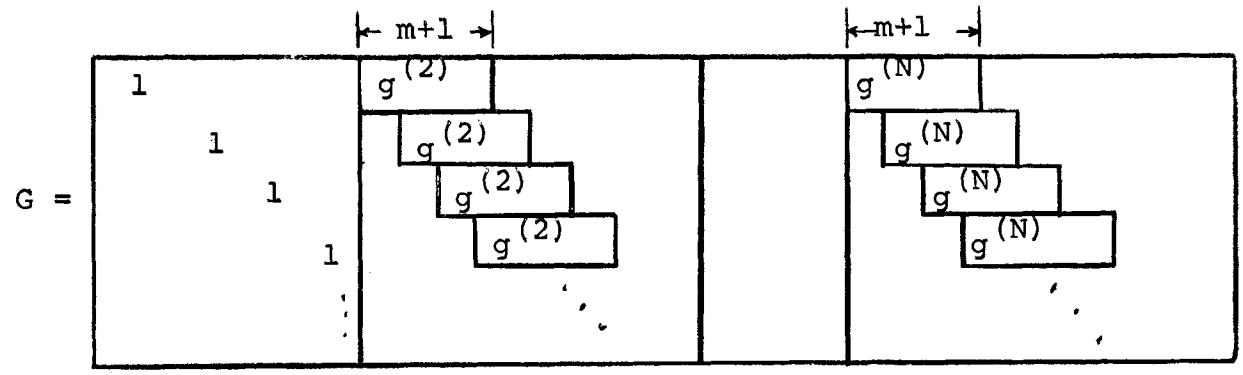
$$d_{free} = \min_{x_0 \neq 0} W_H(xG).$$

In words, this last definition states that d_{free} is the minimum weight taken over all nonzero elements in the row space of G that have nonzero first components.

3.2 Row and column distance

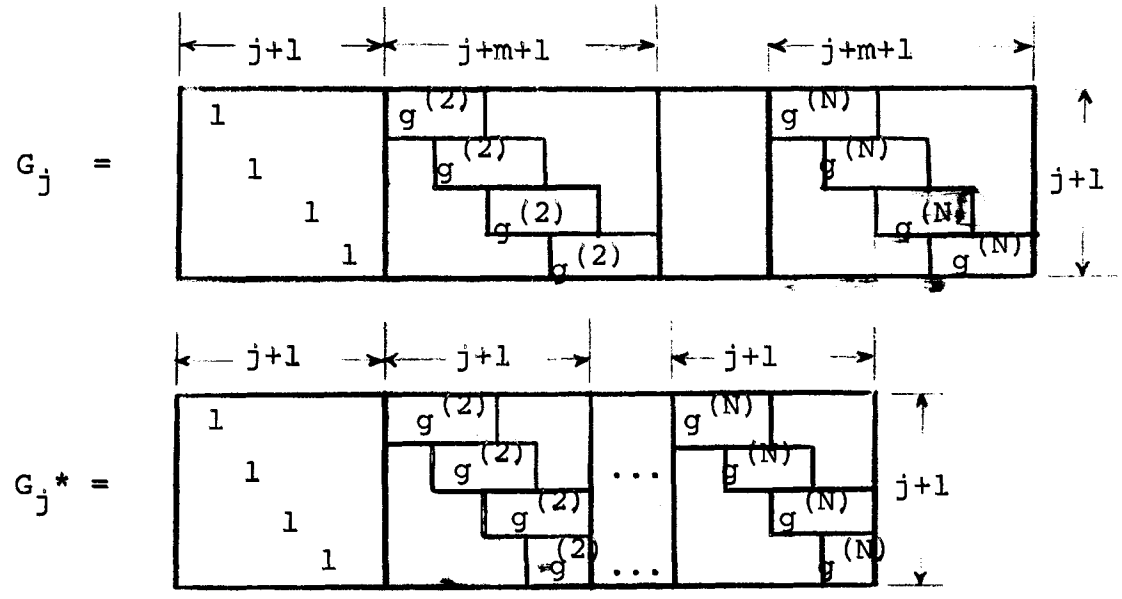
In principle, the free distance of the code generated by G

can be found by generating a list of all nonzero elements of the row space of G. The problem is that the generator matrix



has an infinite number of rows and columns so that the row space is an infinite collection of infinite-dimensional vectors, Fortunately, it turns out that d_{free} can always be found by examining certain finite submatrices of G.

With this in mind, we define the following:



G_j is the submatrix of G consisting of the first j+1 rows of G with all-zero columns deleted, and G_j^* is the submatrix of G consisting of the first j+1 columns of each block of G with all-zero rows deleted.

Note that the row space of G_j is essentially (ignoring trailing zeros) a subcode of the code generated by G. This is not true of the row space of G_j^*.

Costello[1,2] has defined the order j row distance of the convolutional code generated by G to be

$$r_j = \min_{x_0 \neq 0} W_H(x_j G_j)$$

where x_j denotes the truncated input vector $(x_0, x_1, \ldots, x_j)$. Similarly, the order j column distance of the code generated by G is defined by

$$d_j = \min_{x_0 \neq 0} W_H(x_j G_j^*).$$

Costello has shown that d_j, r_j, d_{free} and the Hamming weight of a row of G are related by

$$0 < d_j \leq d_{j+1} \leq d_{free} \leq r_{j+1} \leq r_j \leq W_H(1, g^{(2)}, \ldots, g^{(N)}) \quad j=0,1,2,\ldots$$

What we want here is $d_j = d_{free}$ or $r_j = d_{free}$ for sufficiently large j. It turns out that this is the case for classes of codes whose encoders have polynomial (feedback-free) inverses, i.e. "stable" codes which are not subject to catastrophic error propagation. Systematic codes are such a class. Massey[3] has shown that, for stable codes, infinite weight information sequences produce infinite weight code words. Hence we have

(a) There exists a finite L such that

$$r_j = d_{free} \text{ for all } j \geq L.$$

(b) There exists a finite L* such that

$$d_j = d_{free} \text{ for all } j \geq L^*.$$

These relationships are summarized in Figure 3-2.

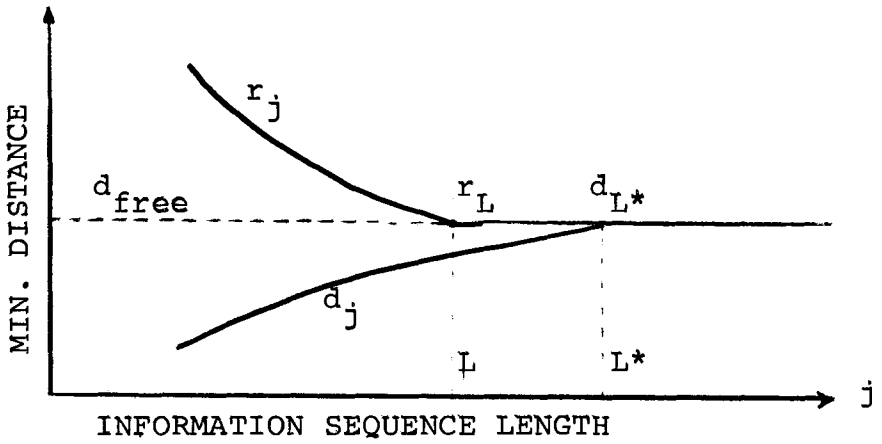


Figure 3-2. Minimum distance relationships

We can find d_{free} by computing either d_{L*} or r_L. The complexity of this computation is critically dependent on having good bounds on L or L*. (It seems unlikely that a simple general method of calculating L or L* exactly will be found.) The remainder of this section is devoted to the consideration of such bounds.

3.3 Results on L*

Costello has given the following upper bound on L*:

For rate 1/N systematic convolutional codes,

$$L^* < (N-1)(m+1)m.$$

This bound increases as the square of the memory order m for a fixed rate 1/N. Costello conjectured that the bound could be improved

to $L^* < 2m$. In a previous note[4], the authors proved the following strong counterexample:

For any fixed N, there exists no fixed s such that $L^* \leq sm$ for all m.

In other words, for any fixed rate 1/N, L^* increases faster than linearly with m. In the above mentioned note, the authors suggested that perhaps L^* increases no faster than m log m. However, more recent investigations put this in doubt. We now sketch one of these.

In what follows, we discuss a class of rate 1/2 codes for which we believe m log m is a lower bound on L^*, although we haven't been able to complete the proof. The construction is of the same form as in (4).

We consider codes in which each of the two subgenerators $g'(x)$ is a polynomial for which the coefficients of x^0, x^3, x^9,..., x^{3^N} are 1's and all other coefficients are zeros. The memory order m for a code in this class is $m = 2 \times 3^N - 1$ and the Hamming weight of the generator is $2(n+1) + 1$. We conjecture that d_{free} for the code is also $2(n+1) + 1$. Assuming this conjecture is true we can derive a lower bound on L^* which is of the order m log m. We begin by noting that the standard selection procedure, i.e., selecting rows of the generator matrix such that for each row selected the first subgenerator aligns with the second subgenerator polynomial of the previously selected row, enables us to get out to column $\frac{m+1}{2} + \frac{(2m+2)}{2} \cdot \frac{(m+1)}{2} = \frac{1}{2}(n+1)(m+1)$ before the column distance reaches d_{free}. Hence we can get to column $\frac{1}{2}n(m+1)$ and still achieve column distance strictly less than d_{free}. From the formula for memory order we get

$$n = \mathrm{Log}_3 \frac{m+1}{2}$$

and this implies we can get to column $\left(\frac{m+1}{2}\right)\log\left(\frac{m+1}{2}\right)$ and still have column distance less that d_{free}. Hence $L^* > \left(\frac{m+1}{2}\right)\text{Log}_3\left(\frac{m+1}{2}\right)$.

3.4 Results on L

By a slight modification of the argument used to derive the upper bound on L*, Costello derived the following upper bound on L:

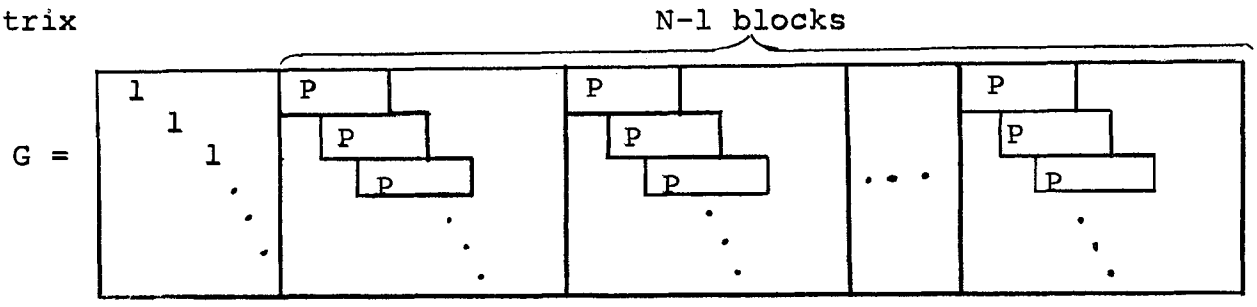
For rate 1/N systematic convolutional codes,

$L < (N-1)(m+1)m-m.$

Again, this is a quadratic function of m and it was thought that perhaps a linear bound could be found. (The fact that L* increases faster than linearly with m does not imply that the same is true of L, since it is always true that L < L*.) A conjecture attributed to Neumann[5] (communicated to the authors by Massey) suggested that $L \leq m+1$. If we do not require that the rate 1/N be fixed, then we can state the following weak counterexample:

There exists no fixed s such that L < sm for all m.

The proof proceeds as follows. Consider the code with generator matrix



where p(D) is a primitive polynomial of degree m. The weight of any code word in the row space of G is the sum of the weight contributions from the first (identity) block, plus N-1 equal contributions from the remaining N-1 blocks. Hence

$$d_{free} = \min_{x_0 \neq 0} W_H(xG)$$

$$= \min_{x_0 \neq 0} \{W_H(x) + (N-1) W_H(xP)\}$$

where P denotes any of the last N-1 blocks of G. First, suppose that x(D) is dual to p(D), i.e. $x(D)p(D) = D^e - 1$ where e is the exponent of p(D). Then since p(D) is primitive, x(D) is a maximal length sequence of degree at least $2^m - m - 1$ and weight $W_H(x) = 2^{m-1}$. In this case,

$$W_H(xG) = W_H(x) + (N-1) W_H(xP)$$

$$= 2^{m-1} + 2(N-1).$$

This, of course, implies that

$$d_{free} \leq 2^{m-1} + 2(N-1).$$

Next, let x'(D) be any information polynomial of degree less than $2^m - m - 1$. Then the weight contribution from each of the last N-1 blocks of G must be at least 3 (p(D) generates a cyclic Hamming code which has minimum distance 3). Then

$$W_H(x'G) = W_H(x') + (N-1) W_H(x'P)$$

$$\geq 1 + 3(N-1).$$

Now choose $N = 2^M$. Then

$$W_H(x'G) \geq 1+3(N-1) \geq 1+3(2^m-1) = 2^m+2(2^m-1) > 2^{m-1}+2(N-1) \geq d_{free}$$

which implies that $r_j > d_{free}$ for $j < 2^m-m-1$. Hence L grows faster than linearly (in fact exponentially) with m, which completes the proof.

Note that this result does not hold for N fixed. It merely states that L grows more rapidly than linearly with m if we place no restriction on the rate. The fact that L grows exponentially with m is due to the fact that the rate is allowed to decrease exponentially.

We now consider the more interesting case when N is fixed. What we would like to have here is an infinite class of codes for each rate (or just for rate 1/2, since this could then be extended to any fixed rate 1/N by duplicating the $G^{(2)}$ block of the rate 1/2 codes N-1 times), so that a strong counterexample similar to the one for L* could be proved. We have not been able to find such a class, but individual counterexamples for all rates have now been found. Costello presented counterexamples for $N \geq 3$, but was unable to find one for N = 2. We now present a counterexample for this case as well:

Consider the code generated by

$$G = \begin{bmatrix} 1 & & & 1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ 1 & & & 1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ & 1 & & 1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ & & \ddots & \ddots \end{bmatrix}$$

with $(g^{(2)}(D) = (D+1)\,\phi\,(D)$ where $\phi(D)$ is the generator of the (73,45) cyclic projective geometry code associated with $PG(2,2^3)$. This convolutional code has memory order 29, yet

$$r_j = \begin{cases} > 13 & \text{all } j < 45 \\ \leq 13 & \text{all } j > 45 \end{cases}$$

which implies that $L \geq 45 > 30 = m+1$.

This may be verified as follows. Let $x(D)$ be the polynomial dual to $\phi(D)$, i.e. $x(D) = (D^{73}-1)/\phi(D)$. In this case, it is known that $x(D)$ is a difference set polynomial of degree 45 and weight 9. Then the weight of the code word xG is

$$\begin{aligned} W_H(xG) &= W_H(x)+W_H(xG^{(2)}) \\ &= 9+W_H(x(D)g^{(2)}(D)) \\ &= 9+W_H((D^{73}-1)(D+1)) \\ &= 13 \end{aligned}$$

which of course implies $d_{free} \leq 13$. Next, let $X'(D)$ be any information polynomial of degree less than 45. Then the weight contribution from the $G^{(2)}$ block must be at least 10 (the (73,45) code has minimum distance 10). To prove that $W_H(x'G) = W_H(x') + W_H(x'G^{(2)}) > 13$, it is sufficient to show that if $W_H(x') \leq 3$, then $W_H(x'G^{(2)}) > 13$. This is the case as the reader may verify for himself by exhaustive inspection.

As stated earlier, we would like to have a strong counterexample like the one found for the conjecture on L^*. Since we have not been able to push this through (although we believe that a strong counterexample exists), we must settle for the conjecture:

> For all fixed N, there exists no fixed s
> such that $L \leq sm$ for all m.

This would be proved if we could show that all the rate 1/2 convolutional codes associated with $PG(2,2^k)$, $k \geq 3$, behave like the example above. In this case we would have an infinite class of rate

1/2 "projective geometry convolutional codes" for which L would increase faster than linearly with m. In fact, L would increase roughly as $m^{\log_3 4}$ which would be faster than the m log m suggested for L* in our previous note. (Of course, if L increases faster than m log m, then so does L*.) The problem of proving that the class of projective geometry codes behaves in this manner revolves around the fact that for $k > 3$, $x(D) = (x^n-1)/\phi(D)$ may be a multiple of the difference set polynomial associated with $PG(2,2^k)$ rather than the difference set polynomial itself.

3.5 A conjecture on row distance

Our study has suggested the following conjecture on the row distance of a rate 1/2 systematic code:

> Let e be the exponent of $g^{(2)}(D)$ (i.e. the smallest integer such that $g^{(2)}(D)$ divides D^e-1). Then $r_e = d_{free}$.

We have not made much headway on this; in fact, all we have to show at this time is an example of an attempted "easy" proof that doesn't work. Our attempt was based on the following idea: Suppose x(D) is any information polynomial. Define r(D) and r'(D) by

$$x(D) = q(D)(D^e-1)+r(D)$$

$$x(D) = q'(D)(D^e-1)/g^{(2)}(D)+r'(D).$$

What we hoped to show was that for any x either $W_H(xG) \geq W_H(rG)$ or $W_H(xG) \geq W_H(r'G)$, which would imply that no x(D) of degree

greater than e could be the lowest degree information sequence to produce a code word of weight d_{free}. (We would have to do something about the cases r(D) = 0 and r'(D) = 0, but since the approach doesn't work anyway, we didn't pursue this.)

The following is an example of an information sequence x(D) of degree greater than e and generator polynomial $g^{(2)}(D)$ for which both r(D) and r'(D) generate higher weight code words than does x(D):

$$X(D) = D^{31}+D^{28}+D^{18}+D^{10}+D^{2}+1$$

$$g^{(2)}(D) = D^{10}+D^{9}+D^{8}+D^{7}+D^{6}+D^{4}+D^{3}+D^{2}+D+1$$

For this generator we have e = 30 and q(D) = D

$$q'(D) = D^{11}+D^{10}+D^{9}+D^{6}+D^{3}+1$$

$$r(D) = D^{28}+D^{18}+D^{10}+D^{2}+D+1$$

$$r'(D) = D^{19}+D^{18}+D^{17}+D^{14}+D^{13}+D^{12}+D^{11}+D^{10}+D^{9}+D^{8}+D^{6}+D^{5}+D^{4}+D^{3}+D^{2}+D.$$

It is easily verified that for these polynomials

$$W_H(xG) = W_H(x)+W_H(xG^{(2)}) = 6+22 = 28$$

$$W_H(rG) = W_H(r)+W_H(rG^{(2)}) = 6+26 = 32$$

$$W_H(r'G) = W_H(r')+W_H(r'G^{(2)}) = 16+18 = 34.$$

REFERENCES FOR SECTION 3

1. D.J. Costello, Jr., "A Construction Technique for Random-Error-Correcting Convolutional Codes," IEEE Trans. on Information Theory, IT-15, pp. 631-636, September, 1969.
2. D.J. Costello, Jr., "Construction of Convolutional Codes for Sequential Decoding," Ph.D. Thesis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, August, 1969.
3. J.L. Massey, "Catastrophic Error-Propagation in Convolutional Codes," presented at the Midwest Symposium on Circuit Theory, (Notre Dame, Indiana, 1968).
4. A. Miczo and L.D. Rudolph, "A Note on the Free Distance of a Convolutional Code," IEEE Trans. on Information Theory (Corres.), IT-16, pp. 646-648, September 1970.
5. B. Neumann, "Distance Properties of Convolutional Codes," M.S. Thesis, Department of Electrical Engineering, Mass. Inst. of Technology, Cambridge, Mass., August, 1968.

SECTION 4

GILBERT BOUND FOR DEFINITE DECODING

In this section we develop a Gilbert bound for convolutional codes decoded by what is known as the definite decoding[1] method. In order to make this report self-contained, the development here follows very closely the original development by James L. Massey[2] and retains the basic form of his result while improving on the constant multiplier. In the process we develop a theorem of interest in its own right which upper bounds the number of distinct sequences obtainable from an L-stage nonsingular linear FSR such that the sequences have fractional weight δ or less, $0 < \delta \leq 1/3$.

4.1 Convolutional codes

We define a rate $R = \frac{K}{N}$ convolutional code of memory order m over GF(2), the binary number field, by the semi-infinite generator matrix G (see fig. 1). The submatrices I_d and O are the KxK identity and all-zero matrices, respectively, and the submatrices $G_0, G_1, \ldots, G_m$ are (N-K) x K binary matrices. We use $\underline{i}_u$ to denote a K-dimensional column vector where the components of $\underline{i}_u$ are the K binary information bits to be encoded at time instant u. The N encoded digits at time instant u are the components of the column vector whose first K components form $\underline{i}_u$ (we consider only codes in canonic, systematic form) and whose last N-K components, called the parity bits, form the vector $\underline{P}_u$ given by

$$\underline{P}_u = G_0\underline{i}_u + G_1\underline{i}_{u-1} + \ldots + G_m\underline{i}_{u-m}. \qquad (1)$$

$G \times I =$

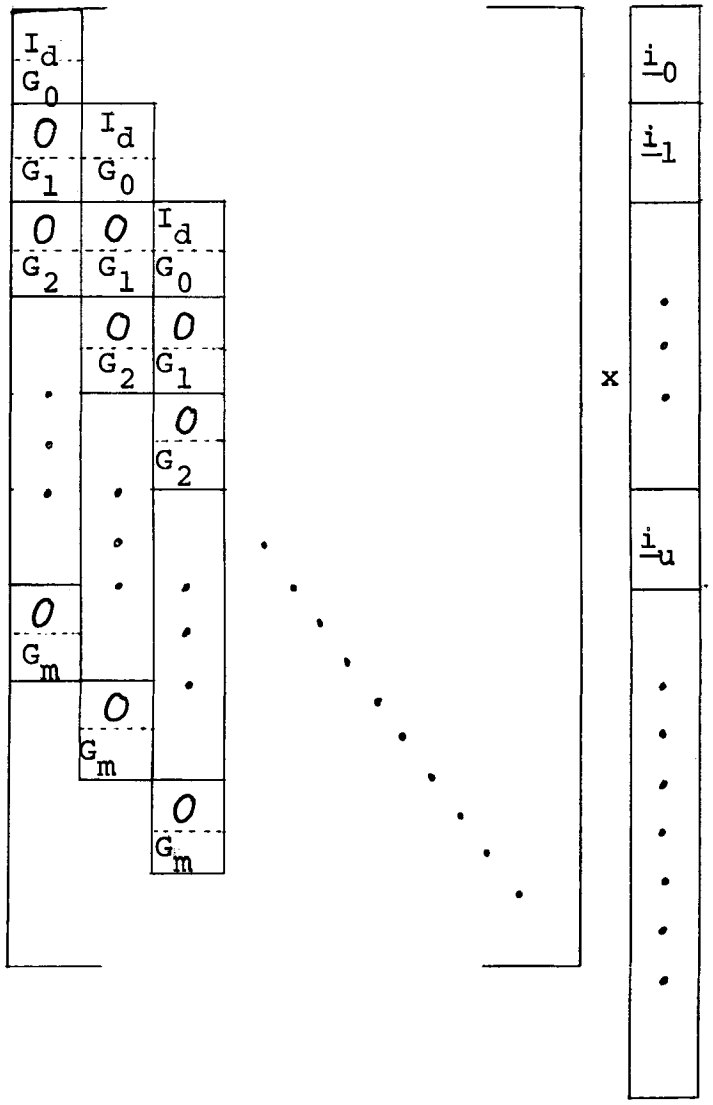


Figure 4-1

The minimum distance of convolutional codes is dependent upon the decoding method. In feedback decoding the estimate of $\underline{i}_u$ is made on the basis of received bits from time unit u through u+m on the assumption that all preceding information vectors have been correctly decoded. With this assumption, decoding of $\underline{i}_0$ is typical of the decoding at any time u and so the feedback decoding minimum distance, d_{FD}, is defined as the fewest number of positions in which two encoded sequences with differing values of $\underline{i}_0$ are found to disagree over time span 0 through m. By the linearity of the code, this is

$$d_{FD} = \min_{\underline{i}_0 \neq \underline{0}} W_H(\underline{i}_0, \underline{i}_1, \ldots, \underline{i}_m, \underline{P}_0, \underline{P}_1, \ldots, \underline{P}_m) \qquad (2)$$

where $W_H(\ )$ denotes the Hamming weight, i.e. the number of nonzero components among the vectors, of the enclosed vectors. There are (m+1)N positions within the time span 0 through m and this number is called the feedback-decoding constraint length, denoted n_{FD}. Most coding bounds are concerned with the ratio of minimum distance to the constraint length, in this case d_{FD}/n_{FD}, and Massey[3] derives the following Gilbert bound for feedback decoding of convolutional codes:

$$H(d_{FD}/n_{FD}) \geq 1-R$$

where $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function.

The decoding method which is called definite-decoding makes no assumptions about the correctness of previous decoding decisions and so the decoding of $\underline{i}_u$ depends not only on the information and

parity digits from time u through u+m but also on the information digits at times u-m through u-1 since these information bits affect the parity bits $\underline{P}_u$ through $\underline{P}_{u+m}$. We can again make $\underline{i}_0$ typical of the general case but we must require that $\underline{i}_u$ be permitted to assume values other than $\underline{0}$ for $u < 0$. We then get for n_{DD}, the definite decoding constraint length, the following:

$$n_{DD} = (2m+1)K+(m+1)(N-K). \tag{3}$$

Again, by the linearity of the code, we can obtain the definite-decoding minimum distance by taking the code word of minimum Hamming weight. We get

$$d_{DD} = \min_{\underline{i}\neq\underline{0}} W_H(\underline{i}_{-m}, \underline{i}_{-m+1}, \ldots, \underline{i}_m, \underline{P}_0, \underline{P}_1, \ldots, \underline{P}_m). \tag{4}$$

Comparing equations (2) and (4) we see that $d_{DD} \leq d_{FD}$ since we can readily get d_{DD} at least as low as d_{FD} simply by setting $\underline{i}_{-m} = \ldots = \underline{i}_{-1} = \underline{0}$. Hence, upper bounds on d_{FD} are upper bounds on d_{DD} but lower bounds on d_{FD} are not lower bounds on d_{DD}.

4.2 Gilbert bounds

In developing codes one of the goals is to obtain codes with high minimum distance since this has a direct bearing on the ability to correct errors. Plotkin bounds on linear block codes and convolutional codes place upper limits on the minimum distance that can be expected for a given block size or constraint length. Gilbert lower bounds demonstrate the theoretical existence of a minimum distance d such that there must exist at least one code with minimum distance surpassing d for a given constraint length. We restrict our

attention in this report to Gilbert bounds for definite decoding and refer the interested reader to Massey's work for a derivation of the Gilbert bound associated with feedback decoding of convolutional codes.

From equation (1) we obtain the matrix equation

$$\begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_m \end{bmatrix} = \begin{bmatrix} \underline{i}'_0 & \underline{i}'_{-1} & \cdots & \underline{i}'_{-m} \\ \underline{i}'_1 & \underline{i}'_0 & \cdots & \underline{i}'_{-m+1} \\ & & & \\ \underline{i}'_m & \underline{i}'_{m-1} & \cdots & \underline{i}'_0 \end{bmatrix} \begin{bmatrix} \underline{G}'_0 \\ \underline{G}'_1 \\ \\ \underline{G}'_m \end{bmatrix} \qquad (5)$$

where the primes are used to denote the transpose of the given matrices. We shall hereafter consider only the case N = K+1, i.e. where $\underline{P}_u$ is a single binary bit and $R = \frac{K}{K+1}$. Thus the matrices G_j become simply K-dimensional row vectors which we denote by $\underline{G}_j$. The matrix of information vectors in (5) will be referred to as the <u>i-matrix</u>, the vector $(\underline{i}_{-m}, \underline{i}_{-m+1}, \ldots, \underline{i}_m)$ will be called the <u>i-vector</u>, and the vector on the left side of (5) will be called the <u>P-vector</u>. The combined <u>i</u>-vector and P-vector, i.e. the vector

$$(\underline{i}_{-m}, \underline{i}_{-m+1}, \ldots, \underline{i}_m, P_0, P_1, \ldots, P_m)$$

will be called the <u>code-vector</u>.

In deriving a Gilbert lower bound on d_{DD} we would like ideally to determine how many code vectors exist with $\underline{i}_0 \neq \underline{0}$ such that the <u>i</u>-matrix has rank r and W_H(code-vector) < d. If we denote this number as M_r then these M_r code vectors can appear in at most a fraction 2^{-r} of all the codes. Hence if

$$\sum_{r=1}^{m+1} M_r 2^{-r} < 1,$$

then there must exist at least one code such that $d_{DD} \geq d$. Unfortunately we cannot determine M_r exactly but can only bound it. We are also hampered by the fact that the bound on M_r can only be shown to hold for $r < 1/3(m+1)$. Therefore the approach to solving the problem will be to consider that fraction of the code vectors with $\underline{i}_0 \neq \underline{0}$ and W_H(code vector) $< d$ for which the $\underline{i}$-matrix has rank $r \geq \Delta(m+1)$, $0 < \Delta \leq 1/3$, and show that this fraction approaches zero as n_{DD} grows arbitrarily large. Then we show that the summation between the limits $r = 1$ and $r = \Delta(m+1)$ is less than 1 for sufficiently large n_{DD} so that there must exist at least one code for which $d_{DD} \geq d$.

4.3 Periodic matrices

Before going into the derivation of the Gilbert bound it is necessary to develop some relationships between periodic matrices and linear feedback shift registers (FSR's). We define a periodic matrix as the matrix

$$\begin{bmatrix} \underline{a}'_0 & \underline{a}'_{-1} & \cdots & \underline{a}'_{-n} \\ \underline{a}'_1 & \underline{a}'_0 & \cdots & \underline{a}'_{-n+1} \\ \vdots & & & \vdots \\ \underline{a}'_m & \underline{a}'_{m-1} & \cdots & \underline{a}'_{-n+m} \end{bmatrix} \quad \begin{matrix} n \geq r \\ \\ 1 \leq r \leq m \end{matrix} \qquad (6)$$

(where each $\underline{a}_j$ is a K-dimensional binary column vector) such that the

matrix has rank r, its first r rows are linearly independent, where we denote the (j+1)-st row as A_j, and the linear combination of the first r rows which produces the (r+1)-st row includes the first row with a multiplier of 1.

We note that the periodic matrix has at least r+1 columns. Also note that since A_r is a linear combination of previous rows including row A_0 with multiplier of 1, it then follows that row A_0 is a linear combination of the following r rows. This immediately implies that the last column is some linear combination of the preceding columns and so the first r columns have rank r even with the last column deleted. Hence by the nature of the matrix the submatrix consisting of rows $A_1, A_2, \ldots, A_r$ must also have rank r and so A_{r+1} must satisfy the same linear combination of A_r and in general we have

$$A_j = \sum_{g=1}^{r} C_g A_{j-g} \qquad (C_r = 1) \qquad j = r, r+1, \ldots, m. \tag{7}$$

Equation (7) leads directly to

$$\underline{a}_j = \sum_{g=1}^{r} C_g \underline{a}_{j-g} \qquad (C_r = 1) \qquad j = r-n, r-n+1, \ldots, m. \tag{8}$$

If we denote the h-th digit in $\underline{a}_j$ as a_{jK+h-1}, h = 1, 2, ..., K, then (8) in turns leads to

$$a_j = \sum_{g=1}^{r} C_g a_{j-gK} \qquad (C_r = 1) \qquad j = (r-n)K, (r-n)K+1, \ldots, mK+K-1. \tag{9}$$

The <u>outer-fringe</u> of a periodic matrix will be defined to be

$$(\underline{a}_{-n}, \underline{a}_{-n+1}, \ldots, \underline{a}_{m}) = (a_{-nK}, a_{-nK+1}, \ldots, a_{mK+K-1})$$

and we note that this is an (m+n+1)K-component column vector. The recursion (9) states that the outer-fringe is an (m+n+1) digit output segment of a linear feedback shift register (FSR) with tap connections every K-th stage as determined by $C_1, C_2, \ldots, C_r$. Since $C_r = 1$, the last stage of this rK-stage linear FSR is always tapped, i.e., the FSR is nonsingular and in this case all output sequences are periodic. Note that the outer-fringe may not contain a complete period of an output sequence since the latter may be as great as $K(2^r-1)$. Also note that the outer-fringe cannot be an output segment of an FSR with fewer than rK stages since then the periodic matrix would be found to have rank less than r. These facts are summarized as:

Theorem 1: The outer-fringe of a rank r periodic matrix is an (m+n+1)K>2rK digit output segment of an unique rK-stage nonsingular linear FSR and of no shorter linear FSR tapped only every Kth stage.

The next theorem shows that for matrices of the form (5) with rank $r < 1/3(m+1)$, if the first $s \leq r$ rows are linearly independent, then the last (r-s) rows are linearly independent of preceding rows and the rows in between are not only dependent but satisfy a recursion of the form (7).

Theorem 2: If the i-matrix in (5) has rank $r<1/3(m+1)$, then the reduced i-matrix

$$\begin{bmatrix} \underline{i}'_r & \underline{i}'_{r-1} & \cdots & \underline{i}'_{r-m} \\ \underline{i}'_{r+1} & \underline{i}'_r & \cdots & \underline{i}'_{r-m+1} \\ & & \vdots & \\ \underline{i}'_{m-r} & \underline{i}'_{m-r-1} & \cdots & \underline{i}'_{-r} \end{bmatrix}$$

is a periodic matrix of rank L, $L \leq r$, whenever $\underline{i}_0 \neq 0$.

Proof: Let I_j denote the (j+1)-st row in the i-matrix of (5) and let s be the least index such that I_s is a linear combination of preceding rows. Let I_{s-L} be the first row appearing with multiplier 1 in the unique combination of the first S rows which forms I_s, then

$$I_s = \sum_{g=1}^{L} C_g I_{s-g} \qquad (C_L = 1) \tag{10}$$

and we note that $L \leq s \leq r$.

If $s = r$, the L rows immediately preceding I_s have rank L and so, as in previous arguments on periodic matrices, row I_s plus the L-1 preceding rows have rank L. Hence I_{s+1} and consequently all rows satisfy the recursion (10) and the reduced $\underline{i}$-matrix is periodic of rank L.

If $s<r$ then there must be some row I_t, $t>s$, which is linearly independent of preceding rows and which does not satisfy the recursion, i.e.,

$$I_j = \sum_{g=1}^{L} C_g I_{j-g} \qquad (C_L = 1) \quad j = s, s+1, \ldots, t-1 \tag{11}$$

but

$$I_t \neq \sum_{g=1}^{L} C_g I_{t-g} \qquad (C_L = 1). \tag{12}$$

We now show that the $\underline{i}$-matrix (5) has rank (m+1)-(t-s). To this end note that (11) and (12) are equivalent to

$$\underline{i}_j = \sum_{g=1}^{L} C_g \underline{i}_{j-g} \qquad j = s-m, s-m+1, \ldots, t-1 \tag{13}$$

and

$$\underline{i}_t \neq \sum_{g=1}^{L} C_g \underline{i}_{t-g} \tag{14}$$

Now suppose some row I_u, for any $u \geq t$, can be written as a linear combination of preceding rows, i.e.,

$$I_u = \sum_{h=1}^{u} a_h I_{u-h} \tag{15}$$

or, equivalently

$$\underline{i}_j = \sum_{h=1}^{u} a_h \underline{i}_{j-h} \qquad j = u-m, u-m+1, \ldots, u \tag{16}$$

This implies

$$\underline{i}_t = \sum_{h=1}^{u} a_h \underline{i}_{t-h}. \tag{17}$$

The terms in the summation on the right of (17) involve only $\underline{i}_j$ for j in the range such that (13) holds. Hence, using (13) in (17) we get

$$\underline{i}_t = \sum_{h=1}^{u} a_h \sum_{g=1}^{L} C_g \underline{i}_{t-h-g} = \sum_{g=1}^{L} C_g \sum_{L=1}^{u} a_h \underline{i}_{t-g-h}. \tag{18}$$

Note that (16) involves summation for j = u-m to j = u and in the righthand side of (18) the quantity t-g corresponds to the j in (16). Hence, since t-g>u-m for all g $\leq$ L we substitute (16) into (18) to get

$$\underline{i}_t = \sum_{g=1}^{L} C_g \underline{i}_{t-g} \tag{19}$$

This contradicts (14) and so we conclude I_u is linearly independent of preceding rows for u $\geq$ t and the only rows which can be written as linear combinations of preceding rows are the t-s rows satisfying (11). Since the matrix has m+1 rows, it has rank r = (m+1) - (t-s). This gives t = (m+1)-r+s>m-r and so the rows of the reduced $\underline{i}$-matrix satisfy the recursion (11) and hence the reduced $\underline{i}$-matrix is periodic and has rank L $\leq$ r.

From (5) we get the following matrix equation:

$$\begin{bmatrix} p_r \\ p_{r+1} \\ \vdots \\ p_{m-r}{}_1 \end{bmatrix} = \begin{bmatrix} \underline{i}'_r & \underline{i}'_{r-1} & \cdots & \underline{i}'_{r-m} \\ \underline{i}'_{r+1} & \underline{i}'_r & \cdots & \underline{i}'_{r-m+1} \\ & & \vdots & \\ \underline{i}'_{m-r}{}_1 & \underline{i}'_{m-r-1}{}_1 & \cdots & \underline{i}'_{-r} \end{bmatrix} \begin{bmatrix} \underline{G}'_0 \\ \underline{G}'_1 \\ \vdots \\ \underline{G}'_m \end{bmatrix} \tag{20}$$

Note that the reduced $\underline{i}$-matrix differs slightly from the reduced

i-matrix of theorem 2 in that the last row here is numbered m_1-r. The reasoning is as follows. Theorem 2 was shown to hold for all $r \leq (m+1)/3$. If we now place a tighter restriction on r and require that $r \leq (m+1)/4$, then the reduced i-matrix has at least 2r-1 rows. This coupled with the fact that $L \leq r$ indicates that if the outer-fringe of this matrix is not an integral multiple of L, then we can eliminate as many as L-1 rows from the matrix to get a further reduced matrix of rank L with m_1 rows such that the outer-fringe is an integral multiple of L. This we now do and henceforth the reduced i-matrix is that matrix containing m_1-2r+1 rows.

We call the left side of (20) the reduced p-vector and note that it is an (m_1-2r+1)-component vector uniquely determined by the reduced i-matrix. The outer-fringe of the reduced i-matrix is an $(m+m_1-2r+1)K$ component vector that we call the reduced i-vector. The combination of the reduced p-vector and reduced i-vector will be called the reduced code vector.

Lemma 1: If the reduced i-matrix is periodic of rank L, then the reduced p-vector is an output segment of an L-stage nonsingular linear FSR uniquely determined by the reduced i-matrix. In particular,

$$p_j = \sum_{g=1}^{L} C_g p_{j-g} \qquad (C_L=1) \qquad j = r+L, r+L+1, \ldots, m_1-r \qquad (21)$$

where C_g, $g = 1,2,\ldots,L$ are the FSR connections uniquely determined by the reduced i-matrix.

Proof: From (9) we see that the digits in each column of the reduced i-matrix satsify the recursion (21). But (20) shows that the reduced p-vector is always a linear combination of these columns and hence

also satisfies the recursion (21).

4.4 A bound on output sequences of FSR's

In this section the results will be expressed in terms of the fractional weight of a vector v which we define to be the quantity $\frac{1}{n}W_H(\underline{v})$ where n is the dimension of $\underline{v}$. Also, we use [x] to denote the integer part of x. We derive an upper bound on the number of vectors or sequences of length n with fractional weight δ or less, $0<\delta\leq 1/3$, such that no two segments coincide in any span of L consecutive digits.

We begin by establishing a lower bound on the average row weight of a matrix in which the rows consist of all L-tuples having Hamming weights between 0 and k and some of the L-tuples with Hamming weight k+1. Before developing the bound we briefly discuss the problem and establish terminology.

Consider the matrix A_k (Fig. 4-2) which consists of all L-tuples with Hamming weights between 0 and k. It will be shown that for $k\leq[L/3]$ the bottom half of the matrix consists only of rows of Hamming weight k (this assumes the rows are ordered in ascending order by binary value). The fraction f_k is defined to be the number of rows of weight k that occur in the first half of A_k divided by the total number of rows of weight k. It is readily seen that

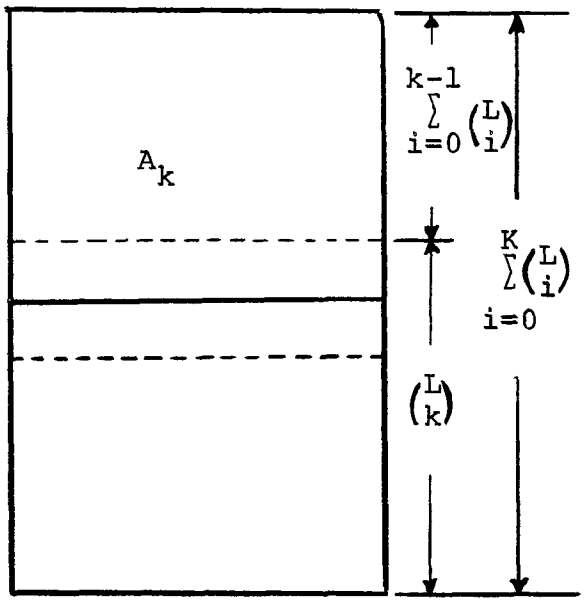


Fig. 4-2

$f_k = \frac{1}{2} \cdot \frac{\binom{L}{K} - \sum_{i=0}^{k-1} \binom{L}{i}}{\binom{L}{k}}$. Thus each half of the matrix consists of

$\sum_{i=0}^{k-1} \binom{L}{i} + f_k \cdot \binom{L}{k}$ rows and throughout this report we will denote this number as $\sum_{i=0}^{(k-1)+f_k} \binom{L}{i}$.

If it can be shown that the average row weight of A_k is greater than $(k-1)+f_k$ and the average row weight of A_{k+1} is greater than $k+f_{k+1}$ then we maintain that the average row weight of intermediate matrices which have some but not all rows of weight k+1 is always greater than k+f where again f represents the fraction of those rows of weight k which appear in the top half of the matrix. Note by the way that f = 1 corresponds to the matrix in which the top half has all L-tuples between weight 0 and k and the bottom half has exactly $\sum_{i=0}^{K} \binom{L}{i}$ rows of weight k+1.

Now consider what happens when we begin with the matrix A_k and start adding rows of weight k+1. The matrix A_k has average row weight $\overline{W}_r = \frac{\sum_{i=0}^{K} \binom{L}{i} \cdot i}{\sum_{i=0}^{k} \binom{L}{i}} = \frac{N_1}{N_2}$. If we add z rows of weight k+1 then the average row weight becomes $\overline{W}_r = \frac{N_1 + Z \cdot (k+1)}{N_2 + Z}$. If we treat this as a function of the number of rows and differentiate, we get $\overline{W}'_r(z) = \frac{(N_2+Z) \cdot (k+1) - (N_1 + Z \cdot (k+1))}{(N_2+Z)^2} = \frac{N_2(k+1) - N_1}{(N_2+Z)^2}$. From this we see that the average row weight increases more rapidly when we add the first few rows of weight k+1 and increases less rapidly

as we approach the matrix A_{k+1}. A graph depicting the behavior of the average row weight versus the number of rows might appear as illustrated in Figure 4-3.

We turn our attention now to proving the lower bound on average row weight of the matrix A_k. We begin with the following:

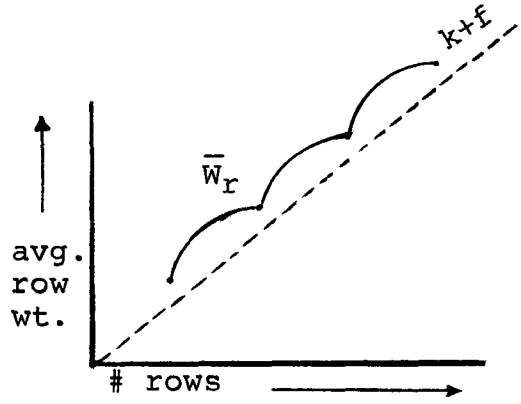


Figure 4-3

Lemma 2: $\binom{L}{k} \geq \sum_{i=0}^{k-1} \binom{L}{i}$ for $k \leq [L/3]$.

Proof: We prove by induction on k for arbitrary L and for all k such that k satisfies $k \leq [L/3]$.

For K = 1 we get

$$\binom{L}{1} > \binom{L}{0}.$$

Now assume the lemma true for k, we attempt to show the lemma true for k+1 (we require $k+1 \leq [L/3]$)

$$\binom{L}{k+1} = \frac{L-k}{k+1} \cdot \binom{L}{k} \geq \frac{L-L/3}{L/3} \cdot \binom{L}{k} \geq 2\binom{L}{k}$$

but $\binom{L}{k} \geq \sum_{i=0}^{k-1} \binom{L}{i}$ by induction hypothesis

so, $\binom{L}{k+1} \geq \binom{L}{k} + \sum_{i=0}^{k-1} \binom{L}{i} = \sum_{i=0}^{k} \binom{L}{i}$.

Lemma 3: $$\frac{\sum_{i=0}^{k} \binom{L}{i} \cdot i + \sum_{i=0}^{k} \binom{L}{i} \cdot (k+1)}{2\sum_{i=0}^{k} \binom{L}{i}} > k \text{ for } k \leq [L/3].$$

Proof: Again we prove by induction and for k = 1 we get

$$\frac{\binom{L}{1} + \binom{L}{0}(1+1) + \binom{L}{1}(1+1)}{2\{\binom{L}{0} + \binom{L}{1}\}} = \frac{2\binom{L}{0} + 3\binom{L}{1}}{2\binom{L}{0} + 2\binom{L}{i}} > 1$$

Now we assume the lemma true for k-1 and consider the matrix illustrated in Figure 4-4. Since $k \leq [L/3]$ it is easily seen that the broken lines are in the correct positions relative to the solid line dividing the two halves of the matrix. We begin by summing the ones in the three parts of the matrix. By the induction hypothesis sub-matrix (A) has average row weight greater than k-1 so has more than $2(k-1)\sum_{i=0}^{k-1}\binom{L}{i}$ ones.

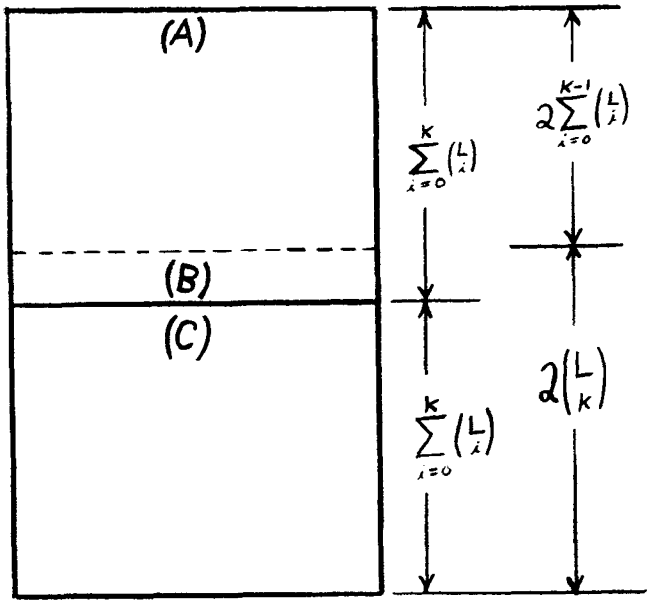


Figure 4-4

Since $k \leq [L/3]$ all rows of part (B) have k ones and so part (B) has $k(\binom{L}{k} - \sum_{i=0}^{k-1}\binom{L}{i})$ ones. Part (C) has $(k+1)\sum_{i=0}^{k}\binom{L}{i}$ ones and now adding the number of ones in all three parts of the matrix we see that it has more than $2k\sum_{i=0}^{k}\binom{L}{i}$ ones and so has average row weight greater than k.

Lemma 4: The average row weight of an MxL matrix in which all rows are distinct and $M > 2(\sum_{i=0}^{k}\binom{L}{i} + f\cdot\binom{L}{k+1})$ is greater than k+f where $0 \leq f < 1$ and $k \leq [L/3]$.

Proof: It is immediately obvious that any MxL matrix in which all rows are distinct must have average row weight at least as great as

the minimal weight matrix described in preceding paragraphs. As also was mentioned previously we need only consider the matrices $A_k, A_{k+1}, \ldots$. Hence we wish to show that

$$k+f_{k+1} < \frac{\sum_{i=0}^{k} \binom{L}{i} \cdot i + f_{k+1} \cdot \binom{L}{k+1} \cdot (k+1) + \{ \sum_{i=0}^{k} \binom{L}{i} + f_{k+1} \cdot \binom{L}{k+1} \} \cdot (k+1)}{2\{ \sum_{i=0}^{k} \binom{L}{i} + f_{k+1} \cdot \binom{L}{k+1} \}}$$

$$= \frac{\sum_{i=0}^{k+1} \binom{L}{i} \cdot i}{\sum_{i=0}^{k+1} \binom{L}{i}} ,$$

where $f_{k+1} = \frac{1}{2} \cdot \frac{\binom{L}{k+1} - \sum_{i=0}^{k} \binom{L}{i}}{\binom{L}{k+1}}$.

To show this inequality holds we begin by substituting its value for f_{k+1} and multiplying both sides by $2\binom{L}{k+1}$ to get

$$(2k+1)\binom{L}{k+1} - \sum_{i=0}^{K} \binom{L}{i} < \frac{2\binom{L}{k+1} \sum_{i=0}^{k+1} \binom{L}{i} \cdot i}{\sum_{i=0}^{k+1} \binom{L}{i}}$$

this is equivalent to

$$(2k+1)\binom{L}{K+1} \sum_{i=0}^{k+1} \binom{L}{i} < 2\binom{L}{k+1} \sum_{i=0}^{k+1} \binom{L}{i} \cdot i + \{ \sum_{i=0}^{K} \binom{L}{i} \} \{ \sum_{i=0}^{k+1} \binom{L}{i} \}$$

and if we bring the $(k+1)^{st}$ terms out from the summations and eliminate like terms we get

$$2k\binom{L}{k+1}\sum_{i=0}^{K}\binom{L}{i} < \binom{L}{k+1}^2 + 2\binom{L}{k+1}\sum_{i=0}^{K}\binom{L}{i}\cdot i + \{\sum_{i=0}^{K}\binom{L}{i}\}^2$$

now add $-2\binom{L}{k+1}\sum_{i=0}^{K}\binom{L}{i}$ to both sides

$$2(k-1)\binom{L}{k+1}\sum_{i=0}^{K}\binom{L}{i} < 2\binom{L}{k+1}\sum_{i=0}^{K}\binom{L}{i}\cdot i + \{\binom{L}{k+1} - \sum_{i=0}^{K}\binom{L}{i}\}^2$$

divide through by $2\binom{L}{k+1}$

$$(k-1)\sum_{i=0}^{K}\binom{L}{i} < \sum_{i=0}^{K}\binom{L}{i}\cdot i + \frac{1}{2\binom{L}{k+1}}\{\binom{L}{k+1} - \sum_{i=0}^{K}\binom{L}{i}\}^2$$

Since the second term on the right is non-negative we can apply the preceding lemma to the remaining terms of the inequality and verify that it is true.

Corollary: $\sum_{i=0}^{k+f}\binom{L}{i} \leq 2^{LH(\frac{k+f}{L})}$ for $k \leq [L/3]$ and $L \geq 2$.

Proof: The bound is known true for conventional summations[4] so we need only show it is true for intermediate values between k and k+1. To this end we consider the slopes of the two curves. Between the points k and k+1 the summation has the constant slope $\binom{L}{k+1}$ and $2^{LH(\frac{k+f}{L})}$ has slope $L\cdot 2^{LH(\frac{k+f}{L})}\cdot(\log_e 2)\cdot(\log_2(\frac{L-(k+f)}{k+f}))$.

The value $\binom{L}{k+1}$ can be rewritten as $\binom{L}{k}\cdot\frac{L-k}{k+1}$ and we observe that in the area of interest, i.e. $k \leq [L/3]$ and $L \geq 2$, the function

$$2^{LH(\frac{k+f}{L})}$$

has the greater slope and thus it is not possible for the numerical value of the summation to overtake the value of $2^{LH(\frac{k+f}{L})}$ between the points k and k+1.

Theorem 3: For any $n = k.L > 0$, k a positive integer, and any δ, $0<\delta\leq\frac{1}{3}$, the number M of binary n-digit segments in any set such that each segment has fractional weight δ or less and no two segments coincide in any span of L consecutive digits is bounded by $2^{LH(\delta)+1}$.

Proof: Consider all k of the MxL submatrices which can be constructed from the set of n-digit segments. At least one of these submatrices must have fractional weight δ or less and so the average row weight of this submatrix is bounded by δL. Applying lemma 4 and its associated corallary we see that

$$M \leq 2\left\{\sum_{i=0}^{[\delta L]}\binom{L}{i} + (\delta L - [\delta L])\binom{L}{[\delta L]+1}\right\} \leq 2^{LH(\delta)+1}. \quad (22)$$

An immediate application of theorem 3 yields:

Lemma 5: For any $n = kL>0$, k a positive integer, and any δ, $0<\delta\leq 1/3$, of the 2^L distinct output sequences of length n obtainable from an L-stage nonsingular linear FSR, fewer than $2^{LH(\delta)+1}$ have fractional weight δ or less.

Proof: Note that any L consecutive digits in an output segment determine a state of the FSR so that any two segments which agree in such a span must agree everywhere thereafter. But since the output sequences of the FSR are periodic the segments must also agree in their previous digits and hence must be the same segment. The lemma now follows from Theorem 3.

Lemma 6: Given fixed values of m, m_1, n, k, and r in the $\underline{i}$-matrix of equation (20), the number of distinct outer-fringes of rank r periodic matrices such that the outer fringe has fractional weight δ or less, $0<\delta\leq 1/3$, is less than $2^{2KrH(\delta)+1}$.

Proof: It is easily shown that if the shortest linear FSR which can generate an n-digit, $n \geq 2L$, segment has length L, than any 2L successive digits in the segment uniquely determine the FSR. Hence, from theorem 1, we conclude that any 2Kr successive digits in the outer-fringe uniquely determine the entire outer-fringe. Thus there can be no more valid outer-fringes of fractional weight δ or less than there are $(m+m_1-2r+1)K > 2rK$ digit segments of fractional weight δ or less such that no two coincide in any 2rK consecutive positions. By theorem 3, this number is less than $2^{2rKH(\delta)+1}$.

4.5 Gilbert bound for definite decoding

We now have the necessary tools to develop the Gilbert bound for definite decoding. Recall that in an earlier section, it was mentioned that the rank problem would be handled by breaking the set of code vectors of fractional weight δ or less and $\underline{i}_0 \neq \underline{0}$ into two sets; set S_1 contains the code-vectors such that the $\underline{i}$-matrix has rank r satisfying $r \geq \Delta\,(m+1)$ and set S_2 contains those for which the $\underline{i}$-matrix has rank r, $r < \Delta\,(m+1)$. In looking at set S_2 we will want to work with the reduced $\underline{i}$-vectors and so the paragraph following equation (20) tells us that Δ must be less than 1/4.

We permit Δ to remain arbitrary for the present and note that for $r < \Delta'(m+1)$, if the reduced p-vector has fractional weight δ_p', it must then have absolute weight $(m_1-2r+1)\delta_p' \geq (2m-3r+2)\delta_p'$ and so the entire code vector must have fractional weight δ' satisfying

$$\delta' \geq \frac{2m-3r+2}{n_{DD}}\delta_p' > \frac{2-3\Delta}{2K+1}\,\delta_p'. \tag{23}$$

Similarly, if the reduced $\underline{i}$-vector has fractional weight δ_i, then

$$\delta' \geq \frac{(m+m_1-2r+1)K}{n_{DD}}\,\delta_i \geq \frac{(2m-3r+2)K}{n_{DD}}\,\delta_i$$

$$\geq \frac{\{2(m+1)-3\Delta(m+1)\}K}{(m+1)(2K+1)-K}\,\delta_i = \frac{(2-3\Delta)K}{(2k+1)-\frac{K}{m+1}}\,\delta_i$$

so,

$$\delta' > (1-\tfrac{3}{2}\Delta)\cdot\frac{2k}{2k+1}\cdot\delta_i \tag{24}$$

Our object now is to show that for some fixed δ we can always demonstrate the existence of a code with minimum distance $d_{DD} \geq \delta n_{DD}$ as n_{DD} grows arbitrarily large. Toward this end we consider first the set S_1. The set S_1 cannot contain more than all code vectors of fractional weight δ or less and each vector in S_1 appears in a fraction at most $2^{-(m+1)\Delta}$ of all codes.

Hence fraction F_1 of codes which contain any vector in S, satisfies

$$F_1 \leq \sum_{j=0}^{[\delta n_{DD}]} \binom{n_{DD}}{j} 2^{-(m+1)\Delta} \leq 2^{-n_{DD}\{\frac{\Delta}{2k+1} - H(\delta)\}}.$$

In considering the S_2 we begin by choosing

$$\delta_i = \frac{2K+1}{2K} \cdot \frac{1}{1-1.5\Delta} \quad \delta < 1/2 \tag{25}$$

and

$$\delta_p = \frac{2K+1}{2-3\Delta} \delta < 1/2 \tag{26}$$

where we note that the inequalities impose the restriction $\delta < \frac{1-1.5\Delta}{2K+1}$. From equations (23) and (24) we observe that any vector in S_2 must have both fractional weight δ_i or less in its reduced i-vector and fractional weight δ_p or less in its reduced p-vector. Hence, the number of distinct reduced code-vectors in S_2 such that the reduced i-matrix has some given rank L is less than

$$2^{2KLH(\delta_i)+1} \cdot 2^{LH(\delta_p)+1} = 2^{2KLH(\delta_i) + LH(\delta_p)+2}$$

which follows from the fact that lemma 6 gives the first factor as bounding the number of reduced i-vectors to be considered whereas lemmas 1 and 5 give the second factor as bounding the number of p-vectors to be considered with any given i-vector. We note also that the reduced i-vector is a non-zero output segment from a KL-stage nonsingular linear FSR and hence must have at least one non-zero digit every KL digits. The fact that the reduced i-vector has $(m+m_1-2r+1)K$ components along with the inequality $L \leq r < m/4$ gives us

$$(m+m_1-2r+1)K > 2LK.$$

Hence, the reduced i-vector must have at least two non-zero components and so we must have fractional weight $\delta_i > \frac{2}{3} \cdot \frac{1}{KL}$. From this we conclude

that S_2 contains no reduced $\underline{i}$-vector such that $L \leq \frac{2}{3K\delta_i}$.

The fraction of codes containing any reduced code-vector such that the reduced $\underline{i}$-vector has rank L is at most 2^{-L}. Then the fraction F_2 of codes containing any code-vector in S_2 satisfies

$$F_2 < \sum_{L=[\frac{2}{3k\delta_i}+1]}^{\Delta(m+1)} 2^{2kLH(\delta_i)+LH(\delta_p)+2-L} .$$

We are interested in asymptotic results as $n_{DD} = (m+1)(2K+1)-K$ grows arbitrarily large and so we replace $\Delta(m+1)$ in the above summation with ∞. We also use equations (25), (26) and the convexity of the entropy function to obtain

$$F_2 < \sum_{L=[Z_1 \cdot \frac{1}{\delta}+1]}^{\infty} 2^{-L\{1-Z_2 \cdot H(\delta)\}} + 2 \qquad (27)$$

where for convenience of notation we use Z_1 and Z_2 to represent $\frac{2}{3} \cdot \frac{2(1-1.5\Delta)}{2k+1}$ and $\frac{3}{2} \cdot \frac{2k+1}{1-1.5\Delta}$ respectively.

Summing up the geometric series yields

$$F_2 < \frac{2^{-\{1-Z_2 \cdot H(\delta)\}[Z_1 \cdot \frac{1}{\delta}+1]} + 2}{1-2^{-\{1-Z_2 \cdot H(\delta)\}}}$$

provided that

$$H(\delta) < \frac{1}{Z_2} . \qquad (28)$$

Combining the expressions for F_1 and F_2 we get the result

that the fraction of codes containing any element of S_1 or S_2 is at most

$$F_1 + F_2 < 2^{-n_{DD}\{\frac{\Delta}{2k+1} - H(\delta)\}} + \frac{2^{-\{1-Z_2 \cdot H(\delta)\}[Z_1 \cdot \frac{1}{\delta} + 1] + 2}}{1-2^{-\{1-Z_2 \cdot H(\delta)\}}} .$$

If δ is required to be sufficiently small so that $H(\delta) < \frac{\Delta}{2k+1}$, then the first term on the right vanishes as n_{DD} gets large. If we choose $\Delta = \frac{2}{9}$ and if we choose δ to satisfy

$$H(\delta) < \frac{1}{5} \cdot \frac{1}{2K+1} \tag{29}$$

then it can be verified that (25), (26) and (28) are all satisfied. We now need to show that the second term on the right is less than 1. From the summation, equation (27), it can be seen that the term will take on its maximum value when $H(\delta) = \frac{1}{5} \cdot \frac{1}{2K+1}$. By means of some algebraic manipulation, substituting $\frac{1}{5} \cdot \frac{1}{2K+1}$ for $H(\delta)$, and evaluating Z_1 and Z_2 for $\Delta = \frac{2}{9}$ we get

$$F_2 < \frac{2^{-(1-\frac{9}{20})[\frac{8}{9} \cdot \frac{1}{2K+1} \cdot \frac{1}{\delta} + 1] + 2}}{1-2^{-\frac{11}{20}}} . \tag{30}$$

We now make use of the fact that, for $K \geq 1$,

$$H(\delta) < \frac{1}{5} \cdot \frac{1}{2K+1} \leq \frac{1}{15} = 0.06666...$$

and for δ's satisfying the inequality we have*

$$8.4\delta < H(\delta)$$

*Using L'Hospitals' rule and the fact that the limit of the sum is equal to the sum of the limits, it is readily seen that $\frac{H(\delta)}{\delta}$ grows without bound as $\delta \to 0$.

so we get

$$\frac{1}{8.4\delta} > \frac{1}{H(\delta)} > 5(2K+1)$$

or,

$$\frac{1}{\delta} > 42(2K+1).$$

From equation (30) it is evident that the bound on F_2 takes on its maximum value for the minimum value of $\frac{1}{\delta}$ so we choose $\frac{1}{\delta} = 42(2K+1)$. We then get $F_2 < 4 \times (.683)^{38} \div (1-.683) < 1$. Hence, whenever (29) is satisfied, not all codes contain code vectors with $\underline{i}_0 \neq \underline{0}$ and fractional weight δ or less. We conclude that there exists at least one code with definite-decoding minimum distance d_{DD} satisfying

$$H(\frac{d_{DD}}{n_{DD}}) \geq \frac{1}{5} \cdot \frac{1}{2K+1}$$

for n_{DD} sufficiently large. We have thus obtained the following:

Theorem 4: For N = K+1 (and hence $R = \frac{K}{K+1}$), and for all n_{DD} sufficiently large there exists at least one convolutional code such that

$$H(\frac{d_{DD}}{n_{DD}}) \geq \frac{1}{5} \cdot \frac{1}{2K+1} = \frac{1}{5} \cdot \frac{1-R}{1+R}.$$

Since this report adheres quite closely to Massey's original development, the remarks following Theorem 4 of his work which culminate in his Theorem 5 also apply here and hence the Theorem holds for any N > K.

The question of whether or not the constant factor can ever be totally eliminated must be considered in light of the upper bound on the set S_1. For the upper bound it was required that the quantity $\left(\frac{\Delta}{2K+1} - H(\delta) \right)$ remain positive so the bound would eventually vanish for large n_{DD}. Thus it is immediately seen that $H(\delta)$ must be less than $\frac{\Delta}{2K+1}$, for $0 < \Delta \leq \frac{1}{3}$, as long as the upper bound on F_1 remains in its present form. Any hope for getting some other form would seemingly hinge on developing some functional relationship between low weight outer-fringe vectors and the rank of the matrix.

REFERENCES FOR SECTION 4

1. J.P. Robinson, "Error Propagation and Definite Decoding of Convolutional Codes", IEEE Trans. on Information Theory, IT-14, pp. 121-128, January 1968.
2. J.L. Massey, "Some Algebraic and Distance Properties of Convolutional Codes", Error Correcting Codes (ed. H.B. Mann). New York, John Wiley, 1968.
3. J.L. Massey, Threshold Decoding, M.I.T. Press, pp. 15-17, 1963.
4. J.M. Wozencroft and B. Reiffen, Sequential Decoding, M.I.T. Press and Wiley, (see appendix), 1961.