INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONE: 2240-1
CABLE: CENTRATOM - TELEX 460392-1

SMR.379/9

COURSE ON BASIC TELECOMMUNICATIONS SCIENCE

9 January - 3 February 1989

CODING

Reginaldo Palazzo Jr.

FEE-UNICAMP, Dept. Telematica, Campinas, SP. Brazil

CODING: BLOCK AND CONVOLUTIONAL CODES

by

Reginaldo Palazzo Jr. (*)

1 - Introduction

It is known that when using orthogonal signaling waveforms one can make the probability of error arbitrarily small just by allowing M to grow. This is equivalent to allow the number of waveforms to grow. However, this implies that the bandwidth is also increasing. Here, we have the situation where coded waveform takes place since in general the same performance is obtained as in the orthogonal case but with less bandwidth expenditure.

The model of a digital communications system where the channel encoding and decoding is included is shown in Fig. 1.

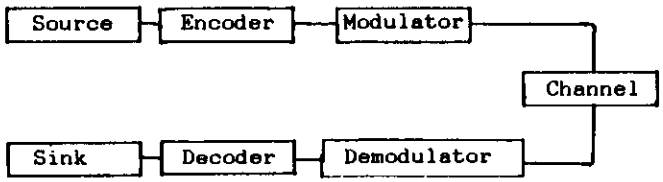

Fig. 1 - Model of a Digital Communications system

We are going to consider two types of encoding. The first one is related to block encoding whereas the second one with convolutional encoding.

(*) - FEE-UNICAMP, Dept Telematica, P.O.Box 6101, 13081 Campinas, SP, Brazil.

By a block encoding we mean that every block of k information bits are encoded into corresponding blocks of n bits where n > k. Since we are going to assume that the Galois field is of order two, GF(2), then the number of codewords is $M = 2^k$. The rate is then given by r = k/n.

By a convolutional encoding we mean that the code is a convolutional one whose encoder can be viewed as a linear finite state machine with an output sequence consisting of a selected set of linear combinations of the input sequence. The number of output bits from the shift-register for each input bit gives the amount of redundancy in the code. The rate is defined as r = k/n, where k information bits are shifted at a time resulting in n encoded bits as output.

The output of the encoder is fed into a modulator for proper processing reagrding the channel characteristics. The binary digits from the encoder output are mapped into elementary signaling waveforms. Usually binary PSK and FSK are the modulations employed. The channel is essentially an additive one whose noise is white Gaussian.

The output of the channel is then processed by a demodulator which consists basically of matched filters to the signal waveform corresponding to eachtransmitted bit. The demodulator output may or may not be quantized. When quantized, it is sampled at a rate 1/T, where T is the duration of the signal waveform. If the quantization levels are equal to 2, then one says that a hard-decison is taking place. If the decoder uses the hard-decison bits to recover the information bits then one says that the decoding process is a hard-decision decoding.

On the other hand, if no quantization takes place at the demodulator and the decoder operate on this analog output to recover the information sequence then one says that the decoding process is a soft-decision decoding. When the output of the demodulator is quantized to K levels, the combination modulator-channel-demodulator is a discrete-time, deiscrete-amplitude channel with input {0,1} and output {0,1,2,....,K). Under the assumption that the channel is the AWGN the resulting channel is a discrete memoryless channel, DMC.

When K = 2, the channel results in a binary input binary output channel with transition probabilities P(0/1) and P(1/0) and probabilities of correct reception P(0/0) and P(1/1). For the binary PSK and FSK we have that P(0/1) = P(1/0) = p and P(0/0) = P(1/1) = 1-p. Then the channel is symmetric and it is called binary symmetric channel, BSC.

When K > 2 and the modulator uses binary PSK and FSK the resulting channel is termed binary input output symmetric channel. In this case, the decoding process is also termed soft-decision decoding.

For q-ary input, we have that if K = q, then we have a q-ary input and q-ary output channel and the decoding process is termed a hard-decision decoding. If K > q, then the channel is a DMC and the decoding process is termed soft-decision decoding.

2 - LINEAR BLOCK CODES

A linear block code is characterized by having all codewords with the same length, say n, and that the closure property holds for the operation being used.

A codeword is represented as a vector whose elements are selected from an alphabet with q elements. When q = 2, we say that the code is a binary linear block code. When q > 2, we say that the code is a nonbinary linear block code. Therefore, for a binary linear block code to be uniquely decodable it has to have at least 2^k codewords for blocks of k information bits. This assures that there is a one-to-one correspondence between the codewords and the blocks of information bits. On the other hand, since the codeword has block length n, then there are 2^n possible codewords in the code. Only 2^k of them are going to be selected. How to select them is of great concern to Coding Theory.

From the analogy to the signal design problem one can readily conclude that the 2^k codewords must be as farther apart as possible from each other.

The measure employed in this selection procedure is the Hamming distance. This distance is defined as the number of places where the two codewords differ. For instance, if u = (0,1,0,1) and v = (1,0,1,1), then the Hamming distance between u and v is equal to 3, since the first three elements of u and v are distinct.

The samllest Hamming distance among the 2 codewords is called the minimum Hamming distance of the code and it is denoted by d_{min}. Another related measure is the Hamming weight. It is defined as the number of elements not equal to zero of a

codeword.. Thus, the Hamming weight of u and v are 2, and 3 respectively.

Going a little bit deeper into linear block codes concepts, we are going to deal with elementary concepts of Linear Algebra, particularly with the concept of a Vector Space.

First of all, the codewords are going to be viewed as vectors in an n-dimensional space. Therefore, the set of all n-tuples form a vector space V. So, if we select k < n linearly independent vectors, the resulting set form a subspace W of dimension k of the vector space V. All the vectors in V which are orthogonal to the vectors of W form the null space of W, which is denoted by NW. The dimension of NW is n-k.

Therefore, the (n,k) linear block code is a set of 2^k n-tuples called codewords which forms a subspace over the field of 2 elements. Its null space is another linear block code with 2^{n-k} codewords of block length n.

Following the convention that a codeword is represented by a row vector, we have that u = (um1, um2, ..., umk) is an information vector and x = (vm1, vm2, ..., vmn) is a codeword. Then the set of k linear equations is represented in matrix form as

$$x = u.G$$

where G is the generator matrix of the code (subspace).

The generator matrix G is then given by

$$G = \begin{bmatrix} g1 \\ g2 \\ . \\ gk \end{bmatrix} = \begin{bmatrix} g11 & g12 & ..... & g1n \\ g21 & g22 & ..... & g2n \\ . & . & ..... & . \\ gk1 & gk2 & ..... & gkn \end{bmatrix}$$

Thus, any codeword is a linear combination of the vectors {gi} of G which form the basis for the (n,k) code.

A generator matrix which can be reduced by row operations and column permutations to the "echelon" form

$$G = [\, I_k : P \,] = \begin{bmatrix} 1 & 0 & ..... & 0 & p11 & p12 & ..... & p1n-k \\ 0 & 1 & ..... & 0 & p21 & p22 & ..... & p2n-k \\ . & . & ..... & . & . & . & ..... & . \\ 0 & 0 & ..... & 1 & pk1 & pk2 & ..... & pkn-k \end{bmatrix}$$

is said to be a generator matrix of a systematic code since the k information bits are present in the encoded output. When a generator matrix can not be reduced to the "echelon" form the code is said to be a nonsystematic one. The (n-k) redundant bits are called parity-check bits.

Since G is the generator matrix of the code, the null space of this code has also a generator matrix which is denoted by H. Then, we have that

$$G.H' = 0$$

as it should be and where ' means transpose. Note that G and H are k x n and (n-k) x n matrices. If G is in the "echelon" form the parity-check matrix H is given by

$$H = [\, -P' : I_{n-k} \,]$$

where the negative sign may be dropped for binary codes.

Example: Consider the (7,4) code with generator matrix

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

then the H' matrix is given by

$$H = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

There are many classes of linear block codes which suit several interesting applications. Among them, we mention a few which are the relevant ones.

HAMMING CODES - These codes have the following property

$$n = 2^m$$
$$k = 2^m - m - 1$$
$$d = 3$$

where $m \geq 2$. For $m = 2$, we have $n = 3$, $k = 1$, $d = 3$, and the code is the repetion code { 000, 111 }. The special property that these codes present is that the parity check matrix H can be easily described. Remember that H is an (n-k) x n matrix so that $n = 2^m - 1$ columns consist of all possible binary vectors with n-k = m elements but the all zero vector.

For instance the (7,4) Hamming code has as its H matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Note that no two columns of H are linearly dependent. However, for m greater than 1, it is possible to find 3 columns of H which add to zero. Thus, the minimum distance is 3.

To obtain the generator matrix from H is a simple matter: 1) first find H'; 2) the last n-k rows have to be the identity matrix; 3) the k first rows form the P matrix of G; 4) remains to include the I identity matrix.

HADAMARD CODES - A Hadamard code is obtained by selecting the rows of a Hadamard matrix. A Hadamard matrix is an n x n matrix of +1 and -1 with the property that any row differs from any other row in exactly n/2 positions. Thus,

$$H2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad H4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

and in general

$$H2n = \begin{bmatrix} Hn & Hn \\ Hn & -Hn \end{bmatrix}$$

Letting +1 → 0 and -1 → 1, we obtain the Hj matrices with elements zeros and ones.

Therefore, for Hadamard codes with block length n, the number of codewords is 2n and minimum distance n/2. In general, a Hadamard code has the following property

$$n = 2^m$$

$$k = \log 2n = m + 1$$

$$d = n/2 = 2^{m-1}$$

when m is a positive integer. When $n \neq 2^m$, Hadamard codes do exist but they are nonlinear.

GOLAY CODE - The Golay code is a binary linear (23, 12) code with d = 7. The extended Golay code can be obtained by adding an overall parity to the (23, 12) code resulting in the (24, 12) code with d = 8.

The Golay code (23, 12) can be generated by the generator polynomial $g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$.

NEW SHORT CONSTRAINT LENGTH CONVOLUTIONAL CODES DERIVED FROM A NETWORK FLOW APPROACH

1 - Introduction

A great deal of research into finding good convolutional codes suitable for source and channel coding problems has been conducted, since convolutional codes have better performance than linear block codes when used with Viterbi decoding or sequential decoding techniques. Despite Forney's effort [1], in establishing an algebraic structure to the encoding problem of convolutional codes, it seems that little is known about it. In view of the difficulty in achieving this goal, many researchers have proposed interesting algorithms [2]-[10], based upon a selection criterion and a computer aided search which led to the determination of good systematic and nonsystematic convolutional codes with long constraint length but limited to a small number of dimensionless rate values.

In this paper a new approach will be presented. It also uses a selection criterion, lowest bit error probability, together with a search. The class of codes searched is reduced substantially by imposing the properties of maximum flow and conservation of flow. The association of maximum flow (branch codewords generated by submatrices with maximum Hamming weight) and conservation of flow (lowest maximum eigenvalue), for a given constraint length K (number of shift-registers) and rate r = b/n with b and n integers, are the properties that optimum nonsystematic time invariant convolutional codes satisfy. These properties are based on the stability property when transfer function equations of convolutional codes are formulated as a control type of problem.

These properties allow determination of good convolutional codes (most of them by hand calculation) for short constraint length and any rate $r = b/n$. Thus, we are able to find good codes for a large range of rates.

2 - Review of Linear Dynamical Discrete Time Systems

The purpose of this section is to review some basic concepts of linear dynamical discrete time systems, [11]-[12], which characterize the behavior and structure of linear trellis codes.

Since the transfer function equations can be interpreted as a linear discrete-time system, we consider a general linear discrete-time system described by linear state difference equations of the form,

$$E(i+1) = A(i)E(i) + B'(i)u(i) \tag{1}$$

and output equations

$$T(i) = H(i)E(i) + D(i)v(i) \tag{2}$$

where $i = 0, 1, 2, 3, \ldots$, $E(i)$ is a state matrix with its elements representing intermediate states, $u(i)$ is a control input, $T(i)$ is the output, $A(i)$ represents the transition matrix, $B'(i)$ is a control matrix, $H(i)$ is the output condition matrix, $v(i)$ is the measurement error sequence, and $D(i)$ is a measurement error matrix at time $t(i) = i$ with elements in the real field.

The solution of the state difference equations is given by

$$E(i) = \Phi(i, i_o)E(i_o) + \sum_{j=i_o}^{i-1} \Phi(i, j+1)B'(j)u(j) \quad i \geq i_o + 1 \tag{3}$$

where

$$\Phi(i, i_o) = \begin{cases} A(i-1)A(i-2) \ldots A(i_o) & , \ i \geq i_o + 1 \\ I & , \ i = i_o \end{cases} \tag{4}$$

and $E(i_o)$ is the initial state condition. From (3), the steady state solution is

$$E(i) = \Phi(i, i_o)E(i_o) \tag{5}$$

Since we are primarily interested in the time invariant case, it follows that $A(i) = A$ for $i_o \leq i \leq i-1$. Diagonalization of the matrix A is sometimes useful. With this objective in mind, one can show that

$$\Phi(i, i_o) = A^{(i-i_o)} = V \cdot \underline{\rho}^{(i-i_o)} \cdot V^{-1} \tag{6}$$

where $\underline{\rho}$ is the diagonal matrix having the distinct eigenvalues, ρ_j, of A as its elements and V is the eigenvector matrix with its j-th row being the components of the eigenvector v_j for $1 \leq j \leq n$, [11]. Let $E(i_o) = E_o$, then the solution of the state difference equations is given by

$$E(i) = \sum_{j=1}^{n} (\rho_j)^{(i-i_o)} v_j w_j E_o \tag{7}$$

where w_j is the j-th row of V^{-1}. Once we know the eigenvalues of the transition matrix A we are able to test the stability condition.

Definition 1 ([11]) : Let the state difference equation be $E(i+1) = f(E(i), u(i), i)$ with the nominal solution $E_o(i_o)$, then the nominal solution is stable in the sense of Lyapunov if for any $t_o = t_{oi}$ and any $\varepsilon \geq 0$ there

exists a $\delta(\varepsilon, t_o) \geq 0$ such that $|E(i_o) - E_o(i_o)| \leq \delta$ implies $|E(i) - E_o(i)| \leq \varepsilon$ for all $i \geq i_o$.

Stability in the Lyapunov's sense simply means that the nominal solution $E_o(i_o)$ is a continuous, convergent, and consequently bounded function with respect to the parameter D, a pairwise error probability. More specifically, $E_o(i_o)$ is the sum of all path values, where each path value is given by the product of branch values belonging to each path. By a path, it is meant a chain of branches linking an initial state, e_o, to an ending state e_2, (see Figure 1).

From Definition 1, the following Theorem establishes the stability conditions for the system in consideration.

Theorem 2 ([11]) : The time invariant linear discrete time system

$$E(i+1) = A \cdot E(i)$$

1 - is stable in the Lyapunov sense if and only if all eigenvalues of A have moduli not greater than 1.

2 - is asymptotically stable if and only if all eigenvalues of A have moduli strictly less than 1.

3 - is exponentially stable if and only if it is asymptotically stable.

Proof : see [11].

If for all j, $|\rho_j| < 1$, then the linear discrete time system is stable. The equivalent correspondence to convolutional codes with the same set of eigenvalues would be a noncatastrophic code. On the other hand, if

for at least one j, $|\rho_j| \geq 1$, then the linear discrete time system is unstable, correspondingly, the specific convolutional code would be classified as catastrophic.

3 - Eigenvalue Problem

It is well known that convolutional codes can be represented by tree, trellis or state diagrams, [13].

Consider a split state diagram of a time invariant convolutional code. The evolution in time of its state is described by

$$E(i+1) = A \cdot E(i) + B \tag{8}$$

where $E(i)$ is a column vector representing the transfer function from the initial state to the intermediate states at time i, A is the transition matrix, and B is the column vector representing the initial condition (see Figure 1). The response of this system, or equivalently, its transfer function, is given by

$$T(i) = H(i) \cdot E(i) \tag{9}$$

where $H(i)$ is the row output matrix (see Figure 1). The solution of (8) is

$$E(i) = (I - A)^{-1} B = \sum_{k=0}^{\infty} A^k \cdot B \tag{10}$$

substituting (9) in (10), we have

$$T(i) = H(i) \sum_{k=0}^{\infty} A^k B \tag{11}$$

since we have a time invariant system, (11) becomes

$$T = \sum_{k=0}^{\infty} HA^kB = H(I - A)^{-1}B \tag{12}$$

which is the steady state output equation.

Let $\tilde{P}(\tilde{b}, \tilde{A}, \tilde{h})$ be a partition of the class of convolutional encoders when represented by its split state diagram with constraint length K and rate $r = b/n$ such that $\tilde{b}$ is a set of $(2^b - 1)$-dimensional initial condition vector, $\tilde{h}$ is a set of $(2^b - 1)$-dimensional output condition vector, and $\tilde{A}$ is a finite set of transition matrices corresponding to each possible tap connection between shift registers and modulo 2 adders for each fixed value of $\tilde{b}$ and $\tilde{h}$ with $\tilde{b}_i$ and $\tilde{h}_i$ the Hamming distances for $1 \leq i \leq 2^b - 1$. Figure 2, shows a partition $\tilde{P}(2, \tilde{A}, 2)$.

The Cayley-Hamilton Theorem states that a transition matrix A satisfies a characteristic polynomial $P(\rho)$

$$P(\rho) = \det.(\rho . I - A)$$

where $\det.(\cdot)$ means determinant, whose solution is the set of eigenvalues. From this, we can find the associated eigenvectors. Thus, for each transition matrix, there exists a characteristic polynomial of the form

$$P(\rho) = \rho^n + a_{n-1}D^{e_1}\rho^{n-1} + ... + a_1D^{e_{n-1}}\rho + a_oD^{e_n} \tag{13}$$

where $0 < D < 1$, $a_i = \pm 1$, and e_i are linear combinations of integer valued constants (accumulated branch Hamming distances along a path) inherent to the structure of the convolutional encoder with $1 \leq i \leq n$.

From section 1, we saw that equation (7) depends upon the eigenvalues and associated eigenvectors, but as will be shown a few steps ahead, by weakenning the upper bound, the final bit error bound will depend upon the eigenvalues alone. Therefore, we can compare performance of convolutional codes being generated by encoders belonging to each partition $\tilde{P}(\tilde{b}, \tilde{A}, \tilde{h})$ and their sets of eigenvalues.

From equations (6) and (12), we have

$$T = \sum_{k=0}^{\infty} H . V . \underline{\rho}^k . V^{-1} . B \tag{14}$$

where $\underline{\rho}$ is an $n \times n$ diagonal matrix which we represent by $\underline{\rho} = \text{diag}.(\rho_1, \rho_2, ..., \rho_n)$.

Let $\rho_{max} = \max\{\rho_1, \rho_2, \rho_3, ..., \rho_n\}$ and $\underline{\rho}_{max}$ a $n \times n$ diagonal matrix with value ρ_{max}. Define $\underline{\rho} \leq \underline{\rho}_{max}$ as the term by term inequality.

Let us define $\tilde{H}$ and $\tilde{B}$ as a new $1 \times (M-1)$ output and a new $(M-1) \times 1$ input matrices with M the number of states, such that

$$\begin{aligned} \tilde{B}_{n,1} &= \max\{B\} \\ \tilde{H}_{1,n} &= \max\{H\}, \qquad \text{for all } n. \end{aligned} \tag{15}$$

when at least one element of the matrix B or H is zero, otherwise H and B are unchanged.

Let us append to each transition in the split state diagram that was originated by the information digit "1" by z. Equation (12) with the above definitions becomes

$$T(z) = \sum_{k=0}^{\infty} \tilde{H}(z) . A^k(z) . \tilde{B}(z) \tag{16}$$

The bit error probability is known to be given by

$$P_b \leq (1/2b)\ \left[d/dz\right]T(z)\Big|_{z=1}$$

Thus, taking derivative of (16) with respect to z

$$P_b \leq (1/2b)\{\tilde{H}'(z)(I - A(Z))^{-1}\tilde{B}(z) + \tilde{H}(z)(I - A(z))^{-1}\tilde{B}'(z) + \\ + \tilde{H}(z) \cdot (I - A(z))^{-1}A'(z)(I - A(z))^{-1}\tilde{B}(z)\}\Big|_{z=1} \qquad (17)$$

where $H'(z) = \left[d/dz\right]H(z)$. Since $\tilde{H}'(z) = \underline{0}$, (17) becomes

$$P_b \leq (1/2b)\ \{\sum_{k=0}^{\infty} \tilde{H}A^k\tilde{B} + \sum_{k=0}^{\infty}\sum_{j=0}^{\infty} \tilde{H}A^kA'A^j\tilde{B}\} \qquad (18)$$

Substituting (6) and $\underline{\rho}$ by $\underline{\rho}_{max}$ in (18) and observing that this condition still holds for $A' = \left[d/dz\right]A(z)$, then

$$P_b \leq (1/2b)\ \{\sum_{k=0}^{\infty} \tilde{H} \cdot V \cdot \underline{\rho}^k \cdot V^{-1} \cdot \tilde{B} + \sum_{k=0}^{\infty}\sum_{j=0}^{\infty} \tilde{H} \cdot V\underline{\rho}^k \cdot V^{-1} \cdot V\underline{\rho} \cdot V^{-1} \cdot V\underline{\rho}^j \cdot V^{-1} \cdot \tilde{B}\}$$

Let $H_{1,n} = \max\{H\} = D^{\tilde{h}}$, or $B_{n,1} = \max\{B\} = D^{\tilde{b}}$ for all $1 \leq n \leq M-1$. Substituting this condition in (19), we have

$$P_b \leq (1/2b)\ [D^{\tilde{b}+\tilde{h}}/(1 - \rho_{max})^2] \qquad (20)$$

Before establishing some lemmas that are consequence of these steps, we need to define the optimality criterion adopted.

Definition 3 : In the class $\tilde{P}(\tilde{b}, \tilde{A}, \tilde{h})$ of asymptotically stable (noncatastrophic) convolutional encoders with constraint length K and rate $r = b/n$, an encoder is optimum if and only if it attains the lowest upper bound on the bit error probability given by (20) for each fixed value of $\tilde{b}$ and $\tilde{h}$.

With this definition, we have the following:

Lemma 4 : For each fixed value of $\tilde{b}$ and $\tilde{h}$ in the class $\tilde{P}(\tilde{b}, \tilde{A}, \tilde{h})$ of asymptotically stable (noncatastrophic) convolutional encoders with constraint length K and rate $r = b/n$, $\gcd(b, n) = 1$, the optimum encoder is the one with the lowest maximum eigenvalue ρ_{max} among all maximum eigenvalues associated with the set of transition matrices $\tilde{A}$.

Lemma 5 : In the class $\tilde{P}(\tilde{b}, \tilde{A}, \tilde{h})$ of asymptotically stable (noncatastrophic) convolutional encoders with constraint length K and rate $r = b/n$, for $\tilde{P}(\tilde{b}_1, \tilde{A}_1, \tilde{h}_1)$ and $\tilde{P}(\tilde{b}_2, \tilde{A}_2, \tilde{h}_2)$ such that $\tilde{b}_1 \neq \tilde{b}_2$ and $\tilde{h}_1 \neq \tilde{h}_2$, if $\rho_{1,max}$, the associated eigenvalue of A_1, equals $\rho_{2,max}$, the associated eigenvalue of A_2, then the optimum encoder is the one such that $\tilde{b}_i + \tilde{h}_i$, $i = 1, 2$, is the largest.

Proof : Substitution of the assumptions in equation (20).

As an example of lemma 5, consider the convolutional encoder belonging to the class $\tilde{P}(4, \tilde{A}, 4)$ with the following octal representation 5775 with $\tilde{b}_1 = 4$ and $\tilde{h}_1 = 4$. Its characteristic polynomial is given by

$$P(\rho) = \rho^3 - D^2 \cdot \rho^2 - D^2 \cdot \rho$$

Now, consider another convolutional encoder belonging to the class $\tilde{P}(2, \tilde{A}, 3)$ with the following octal representation 175 with $\tilde{b}_2 = 2$ and $\tilde{h}_2 = 3$. Its

characteristic polynomial is

$$P(\rho) = \rho^3 - D^2 \cdot \rho^2 - D^2 \cdot \rho$$

Therefore, both convolutional encoders have the same ρ_{max}, even though the corresponding transition matrices are different. Since they have the same ρ_{max}, the optimum encoder between these two must have the greatest $\tilde{b}_i + \tilde{h}_i$. For the convolutional encoder 5775, $\tilde{b}_1 + \tilde{h}_1 = 8$. For the convolutional encoder 175, $\tilde{b}_2 + \tilde{h}_2 = 5$. Thus, the optimum convolutional encoder between these two is the 5775.

From lemma 4 and lemma 5, we have that P_b attains the lowest upper bound among all asymptotically stable encoders when ρ_1 is the lowest maximum eigenvalue and $\tilde{b}_1 + \tilde{h}_1$ is the largest. So, for a fixed ρ_{max} and rate $r = 1/n$, we have that P_b has the lowest bound when $\tilde{b}_1 = \tilde{h}_1 = n$. This implies that the Hamming weight of this (binary) vector component is n. For rates $r = b/n$, since each one of the (2^b-1) (binary) vector components represents the Hamming weight of an associated output sequence, the sum of these Hamming weights can be shown to be $n \cdot 2^{b-1}$. Thus, the optimum convolutional code with rate $r = b/n$ and constraint length K attains the lowest P_b when the $(2^b - 1)$ (binary) vector components of $\tilde{b}_1$ and $\tilde{h}_1$ are almost evenly distributed with total Hamming weight $n \cdot 2^{b-1}$.

As a consequence of these facts, the maximum flow property has been shown. To link this property with the conservation of flow (lowest maximum eigenvalue), the augmented transition matrix (inclusion of the starting and ending states) is needed.

Let A' be the augmented transition matrix. For $K = 3$, $r = 1/n$ and partition $\tilde{P}(a, \tilde{A}, h)$, the augmented matrix A' is as follows (see Figure 2):

$$A' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ D^a z & 0 & D^{a_7} z & 0 & 0 \\ 0 & D^{a_2} & 0 & D^{a_6} & 0 \\ 0 & D^{a_4} z & 0 & D^{a_5} z & 0 \\ 0 & 0 & D^h & 0 & 0 \end{bmatrix}$$

with $z = 1$. In general, for rates $r = b/n$ and constraint length K, A' is a $(M+1) \times (M+1)$ matrix with $M = 2^{b(K-1)}$. From the Cayley-Hamilton Theorem, the characteristic polynomial for A' is

$$P'(\rho) = \det(\rho \cdot I - A') = \rho^2 \cdot \det(A)$$

So, A' contains the same set of eigenvalues of A.

Let us define $M_i(A')$ and $M^j(A')$ as the product of the nonzero elements of the i-th row and the product of the nonzero elements of the j-th column of the augmented transition matrix A', respectively, that is,

$$M_i(A') = \prod_{j=1}^{n} a_{ij} \quad \text{and} \quad M^j(A') = \prod_{k=1}^{n} a_{kj}$$

For rate $r = b/n$ and constraint length K, if $M_i(A') = M^j(A') = D^{\Phi}$, with $\Phi = n \ 2^{b-1}$, implies that the partition $\tilde{P}(\tilde{b}, \tilde{A}, \tilde{h})$ is such that the sum of the $(2^b - 1)$ vector components of $\tilde{b}$ and $\tilde{h}$ equals Φ, respectively. Consequently, $\tilde{b}$ and $\tilde{h}$ assume their maximal value, since this value of Φ is the maximum total weight of a block code with 2^b codewords. So, from lemma 4, P_b attains the lowest upper bound when ρ_{max} is the lowest maximum eigenvalue in the ensemble of transition matrices $\tilde{A}$. The existence of at

least one transition matrix A having ρ_{max} minimum is guaranteed by the fact that the characteristic polynomial

$$P(\rho) = a_n \rho^n + a_{n-1} \rho^{n-1} + ... + a_1 \rho + a_o$$

with half of the coefficients a_i, being positive, half being negative and less than 1, has the smallest positive real root when $a_o \geq 0$, [14]. But $a_o > 0$, implies that there exists a loop of iptermediate states containing at least one branch value of the shortest path which is less than the corresponding branch value of the shortest path of the optimum known code. Hence, the condition $a_o \leq 0$ is a restriction to the characteristic polynomial where ρ_{max} minimum is included. When $a_o = 0$, the characteristic polynomial has the lowest maximum positive real eigenvalue.

This argument lend credency to the conjecture that the properties of maximum flow and conservation of flow must be satisfied by the optimum convolutional code. However, we were not able to prove that <u>always</u> exists such a code, although all the previous codes found and the new ones given in this paper satisfy these properties. We resume this fact by establishing the following:

<u>Conjecture</u> : If there exists a time invariant nonsystematic convolutional encoder with parameters K and $r = b/n$ such that $M_i(A') = M_k(A') = D^{\phi}$ for all $k \neq i$; and $M_i(A') = M^j(A') = D^{\phi}$ for all $i \neq j$ and vice-versa, with $\phi = n \cdot 2^{b-1}$ and ρ_{max} the smallest eigenvalue among all eigenvalues of augmented transition matrix A' (consequently A), then the code is <u>optimum</u>.

Only those convolutional encoders that satisfy the maximum flow and the conservation of flow property are considered, consequently being classified as "potential candidates" to be the optimum.

In order to demonstrate the potential of the maximum flow and conservation of flow property, let us take another example. Let $K = 3$ and rate $r = 1/4$. Find the optimum convolutional encoder in the class $\tilde{P}(4, \tilde{A}, 4)$ (see Figure 3). From the conjecture we have that

1) $a_7 = 0 \qquad a_2 = a_4 = a_5 = a_6 = 2$

2) $a_7 = 0 \qquad a_2 = a_5 = 3 \qquad a_6 = a_4 = 1$

The characteristic polynomial for 1) is

$$P(\rho) = \rho^3 - D^2 \rho^2 2 - D^2 \rho$$

and the eigenvalues for $D = 0.1$ are

$$\rho_1 = 0, \qquad \rho_2 = 0.105, \qquad \rho_3 = -0.095$$

The characteristic polynomial for 2) is

$$P(\rho) = \rho^3 - D^3 \rho^2 - D^3 \rho - D^2 + D^6$$

for $D = 0.1$, the eigenvalues are

$$\rho_1 = 0.2174, \qquad \rho_2 = -0.1082 + j0.3706, \qquad \rho_3 = -0.1082 - j0.3706$$

From lemma 4, the optimum encoder is such that $\rho_{max} = 0.105$. The encoder that satisfies $a_7 = 0$, $a_2 = a_4 = a_5 = a_6 = 2$ is, in octal representation, given by 5775, whereas in the literature 5777 is mentioned as the optimum in the free distance sense, [6], [20]. Although the encoder 5777 achieves the least upper bound on the minimum distance, it has poorer bit error probability than the 5775. This statement is easily checked by using the transfer function technique or by inspection of the split state diagrams

as shown in Figure 3. The bit error probabilities for A) and B) are respectively:

$$P_{bA} \leq (1/2)\ D^{10}/(1 - 2D^2)^2$$

$$P_{bB} \leq (1/2)(2D^{10} + D^{11} - D^{13} - 2D^{14} + D^{17})/(1 - D^2 - 2D^3 + D^6)^2$$

as expected $P_{bA} < P_{bB}$.

The eigenvalue problem is a very interesting approach in finding good systematic as well as nonsystematic convolutional codes for any rate $r = b/n$ and short constraint length. However, its importance lies on the theoretical rather than on the practical point of view.

4 - Convolutional Codes as a Network Flow Problem

Although the split state diagram is useful in determining a bit error bound using a transfer function approach, it is also useful in describing flow in networks as follows.

Let G be a directed graph $G = [N, \hat{A}]$, consists of a finite collection N, $N = \{1, 2, 3, \ldots, n\}$ together with a set $\hat{A}$ of the non-necessarily distinct ordered pairs (i, j), that is, $\hat{A} = \{(i, j) : i, j \in N\}$. Elements of N and $\hat{A}$ will be called states and branches, respectively.

Let us associate with each branch (i, j) of a directed graph (split state diagram) a non-negative number c_{ij}, the capacity of (i, j), to be thought of as representing the maximal branch Hamming distance of the codeword between nodes i and j.

The source, node 1, is the entry of flow into the network, and the sinkm node n, is the exist of flow from the network. Mathematically, the

branch flows x_{ij} (branch Hamming distances), are defined as a set of non-negative numbers satisfying the following constraints:

$$\sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -\phi & \text{if } j = 1 \\ 0 & \text{if } j \neq 1,n \\ \phi & \text{if } j = n \end{cases} \tag{21}$$

$$0 \leq x_{ij} \leq c_{ij} \tag{22}$$

Note that flow is conserved at every node except the source and the sink, and each branch flow x_{ij} is bounded from above by c_{ij}, the branch capacity. From the Max-Flow Min-Cut Theorem, [17], the maximal flow in a network equals the sum of the branch flows of the minimum cut set. We noticed in section 3 that if there exists a nonsystematic convolutional code satisfying the conjecture then conditions (21) and (22) are met. Hence, under this conjecture the problem of finding optimum nonsystematic convolutional codes is characterized by a maximal flow in networks. The following theorem establishes the maximum flow that optimum time invariant convolutional encoders satisfy.

Theorem 6 : For time invariant nonsystematic convolutional codes over Gf(q) with rate $r = b/n$ and constraint length K, the uniform flow, and consequently, the maximum flow is given by

$$\phi = n \cdot (q - 1) \cdot q^{b-1}$$

Proof : For rate $r = b/n$ and constraint length K, the number of states is $q^{b(K-1)}$. The number of transitions from each state is q^b. The total number of branches is given by q^{bK}. On the other hand, the encoder output has length n, and so, q^n possible output sequences.

Let $\underline{c}$ be the ratio between the number of branches and the number of output sequences. If $\underline{c}$ is greater than 1, then this number specifies how many times the output sequences will be repeated. If $\underline{c}$ is less than 1, then it will give the proportion of output sequences that are going to be used.

Let $W_T(C')$ be the total weight of the output sequances, that is,

$$W_T(C') = \sum_{m=0}^{n} m \cdot (q-1)^m \cdot \binom{n}{m} = (q-1) \cdot \left[d/dt\right]\left((1+a)^n\right)_{a=q-1}$$

$$= n \cdot (q-1) \cdot q^{n-1}$$

then $\underline{c}\, W_T(C')$ gives the total weight of the state diagram. Since there are $q^{b(K-1)}$ states, the uniform flow is

$$\phi = \left[\underline{c}\, W_T(C')\right]/q^{b(K-1)} = n \cdot (q-1)q^{b-1} \qquad \text{Q.E.D.}$$

From Theorem 6, we have the following:

Corollary 7 : For time invariant nonsystematic convolutional codes with $b = 1$ and $q = 2$, the branch codewords going to and leaving from any state, complement each other.

Proof : From Theorem 6, we have that the flow at any state is $\phi = n2^{b-1}$. For rate $r = 1/n$, $\phi = n$. Let $d_H(j, i)$ be the Hamming distance between branch codewords j and i, then

$$d_H(j, i) = w(j + i) = n$$

which implies that $w(j) + w(i) = n$, thus $w(j) = n - w(i)$ Q.E.D.

Therefore, the search for the optimum convolutional code for any K and rate $r = b/n$ reduces to the following problem:

Given a split state diagram where all branch weights are lower bounded by zero and upper bounded by c_{ij}, (i, j)'s maximum branch Hamming distance. Find the branch flows, branch distances, such that the minimum distance from the zero state back to itself is maximized subject to the maximum flow ϕ.

A network with these characteristics is called a bounded network. Hence, we have transformed the problem of finding good convolutional codes by using a heuristic technique, into a well structured problem of finding flow in network.

Inherent to this transformation, there are some combinatorial problems that in the majority of the cases can be very easily solved, making possible determination of convolutional codes by hand calculation.

In order to speed up the process, the minimum distance of the code is valuable in the exclusion of "potential candidates" from the class of good codes.

The following Theorem establishes an upper bound on the minimum distance of any convolutional code (time invariant, periodically time varying, time varying) as well as of binary nonlinear trellis codes recurrently generated, with parameters K and $r = b/n$, $\gcd(b, n) = 1$, which is very good for short constraint length. It generalizes Heller's upper bound [7], for convolutional codes with constraint length K and rate $r = 1/n$. In particular we have

Theorem 8 : For any convolutional code with constraint length K and rate $r = b/n$, $\gcd(b, n) = 1$, the minimum distance is upper bounded by

$$d_{min} \leq \min_{p \geq 1} \left\{\left[2^{p-1}/2^p - 1\right] \cdot (n/b)(p + b(K-1))\right\}$$

Proof : It is known that a terminated binary convolutional code with M information bits is a group code with an $M \times n \cdot (M+(K-1))$ dimensional generator matrix for rate $r = 1/n$. For rate $r = b/n$, $\gcd(b, n) = 1$, since we have b parallel K shift registers, in order to terminate a code we need to insert $b(K-1)$ known digits. The total length of the information bits is bM, and so, the generator matrix has now bM rows and $(n/b) \cdot [bM - b(K-1)]$ columns.

For binary group codes, the Hamming distance between codewords is equivalent to the weights of the non zero codewords. Hence, if all 2^{bM} codewords are arranged as rows of a matrix, then any column, excluding the all zero, has half ones and half zeros, [18]. Thus, the total weight of the 2^{bM} codewords is upper bounded by

$$W_T \leq 2^{bM-1} (n/b)(bM + b(K-1))$$

Since $2^{bM} - 1$ codewords are non zero and their minimum weight must be less than or equal to their average weight, the minimum distance satisfies

$$d_{min} \leq [2^{bM-1}/2^{bM} - 1] (n/b)(bM + b(K-1))$$

Since this bound holds for any bM, it also holds for $p < bM$. What we really want is the least upper bound. This is achieved if we minimize the right hand side with respect to p, thus

$$d_{min} \leq \min_{p \geq 1} [2^{p-1}/2^{p} - 1] (n/b)(p + b(K-1)) = d_{bound}$$

where $[a]$, means the largest integer less than or equal to a. Q.E.D.

The improved upper bound developed by Odenwalder [2], when d_{bound} is odd and $r = 1/n$, is also applicable to rates $r = b/n$. It can be shown that the upper bound is given by

$$d_{bound} \leq [2^{c-1}/2^{c} - 1] \{(n/b)(c + b(K-1)) + 2^{1-c} - 1\} \qquad (23)$$

If d_{bound} is odd and (23) is not satisfied, then d_{bound} can be decreased by 1.

The following examples demonstrate the approach so far. Let $K = 3$ and rate $r = 1/7$. Find the optimum convolutional encoder. By Theorem 6, $\Phi = n2^{b-1} = 7$. Among all possible noncatastrophic configurations, we have the following split state diagrams, Figure 4.

By Theorem 8, $d_{min} \leq 18$, and by inspection, configuration C is the optimum. Let us use the approach presented in section 3. The characteristic polynomials and its maximum eigenvalues for $D = 0.1$ are

a) $\rho^3 - D^6 \rho^2 - D^6 \rho - D^2 + D^{12}$; $\rho = 0.215$

b) $\rho^3 - D^5 \rho^2 - D^5 \rho - D^4 + D^{10}$; $\rho = 0.04646$

c) $\rho^3 - D^4 \rho^2 - D^4 \rho - D^6 + D^8$; $\rho = 0.0134$

d) $\rho^3 - D^3 \rho^2 - D^3 \rho + D^6 - D^8$; $\rho = 0.031638$

e) $\rho^3 - D \rho^2 - D\rho - D^{12} + D^2$; $\rho = 0.316666$

f) $\rho^3 - D^2 \rho^2 - D^2 \rho - D^{10} + D^4$; $\rho = 0.1$

Hence, by lemma 4, we choose c), which is the same answer as the previous procedure. Note that when $a_2 = 7$ and $a_5 = 0$ we have a catastrophic code. For the sake of simplicity, let us take $K = 2$ and rate $r = 2/3$. Find the optimum binary convolutional encoder. By Theorem 6, $\Phi = 12/2 = 6$. The split diagram is shown in Figure 5.

So, $a_1 + a_2 + a_3 = 6$ and $a_4 + a_5 + a_6 = 6$, but a_i are the Hamming distances of the branch codewords. Thus, we have to find values for a_i such that: 1) the flow is 6 at the zero state, and conservation of flow holds for all remaining states; 2) the minimum distance is maximum; 3) the state diagram's weight is 24.

The weight of a split state diagram generated by an encoder with constraint length K and rate $r = b/n$ is given by the sum of all branch weights. From Theorem 6, we have that the flow going in and out of every state is $\Phi = n \cdot (q-1) \cdot q^{b-1}$, since there are $q^{b(K-1)}$ states, $W_T = a_i = \Phi \cdot q^{b(K-1)}$. Thus, we have proved:

Lemma 9 : The total weight of a state diagram generated by a convolutional encoder with parameters K and $r = b/n$, $\gcd(b, n) = 1$, is given by

$$W_T = \Phi \cdot q^{b(K-1)} = n \cdot (q-1) \cdot q^{bK-1}$$

Let $\underline{u} = (u_1, u_2, u_3, \ldots)$ be the data input sequence that goes into the encoder shift registers, with u_i having length b and G_i are the $b \times n$ submatrices of the generator matrix G of the convolutional code.

The output codewords are given by $\underline{x} = \underline{u} \cdot G$. Hence, the combinatorial problem is to find the rows of G_i, $i = 0, 1, 2, \ldots, K-1$, such that conditions 1), 2), and 3) are satisfied. By Theorem 8, $d_{min} \leq 4$, for $d_{min} = 4$, all possible G_i give catastrophic codes, then $d_{min} = 3$ is the optimum value for this class. Section 5, gives more details about this. One possible solution of the combinatorial problem is:

$$G_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad G_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Therefore, maximum flow and conservation of flow are important properties with respect to finding good codes by limiting the search to a set $\tilde{S}$ whose codes have maximum flow Φ. The next step in this process is to select those codes in $\tilde{S}$ whose minimum distance (possibly) attains the least

upper bound given in Theorem 8. Let us call this new set $\tilde{N}$. Thus, $\tilde{N}$ contains codes which are optimum under the free distance criterion. Now, in order to select the optimum code under the bit error probability criterion a search procedure is necessary. Before leaving this section, we have the following:

Definition 10 : A convolutional code is said to be optimum under the bit error probability if and only if the coefficients of its transfer function is the lowest among all coefficients of the transfer functions in the ensemble of codes.

5 - Selection of Good Convolutional Encoders

It has been shown in sections 3 and 4 that for all partitions $\tilde{P}(\tilde{b}, \tilde{A}, \tilde{h})$ there exists an optimum encoder. It is also shown that when the parameters are K and $r = 1/n$ and $\tilde{b} = \tilde{h} = n$, partition $\tilde{P}(n, \tilde{A}, n)$, the bit error probability is upper bounded by the lowest value when ρ_{max} is selected as the minimum in the set of all maximum eigenvalues. When this happens, there is a conservation of flow at every single state but the "zero" state in the split state diagram. Although maximum flow and conservation of flow are important properties, they only provide the search in a set whose cardinality is smaller than that of the original problem. For long constraint length, this new set has a considerable number of elements.

The least upper bound on the free distance is another parameter that must be used in the process, which will reduce the search many folds. Therefore, maximum flow and consequently conservation of flow, and free distance are the only parameters that have to be examined.

Consider a split state diagram of a convolutional encoder with

rate $r = b/n$ and $b \geq 2$. The number of transitions from any but the zero state is 2^b. For the zero state this number is $2^b - 1$. Let d_p be the ordered set of all $2^b - 1$ accumulated branch Hamming distances (K branches long) that leave and return the zero state (see Figure 6), that is

$$d_p = \{d_{p1}, d_{p2}, d_{p3}, \ldots, d_{p2^{b}-1}\} \tag{24}$$

with $d_{pi} = w_i + \tilde{w}_i$ and w_i, $\tilde{w}_i$ the Hamming weights of the codewords going from state zero to state i and from state i to state zero respectively. It is natural to expect that

$$d_{free} \geq \min \{d_p\} \tag{25}$$

The optimality criterion to be adopted when comparing ordered sets is of the lexicographical type, that is, given two ordered sets X and Y

$$X = \{X_1, X_2, X_3, \ldots, X_M\}$$

$$Y = \{Y_1, Y_2, Y_3, \ldots, Y_M\}$$

$$X > Y \quad \text{if} \quad \begin{cases} X_i = Y_i & \text{for} \quad 1 \leq i \leq p \\ X_{p+1} > Y_{p+1} \end{cases} \tag{26}$$

Note that this approach does not imply that the encoder is optimum in the bit werror probability sense, since the selected candidates belong to $\tilde{S}$. So far, nothing was said about how to find the values d_{pi} for a given constraint length K and rate $r = b/n$.

From Theorem 6, we know that the maximum flow with $q = 2$ is $\Phi = n2^{b-1}$ which equals the sum of all Hamming distances going in or out of a state. Since we have an encoder with K storage elements, K Φ is an upper

bound on the weighted sum of all d_{pi} belonging to the K branch long paths. But this is a linear programming problem stated as follows:

Problem I - Given

$$\sum_{i=1}^{B} a_i d_{pi} \leq K \cdot \Phi \tag{27}$$

subject to

$$\sum_{i=1}^{B} a_i = 2^b - 1$$

with $d_{p1} = d_{free}$, $d_{p2} = d_{free} + 1$, and so on with B is a constant. Find the lexographically minimum B-tuple element $\hat{a}_i = (a_{i1}, a_{i2}, \ldots, a_{iB})$ satisfying (27).

Once the $\hat{a}_i$'s are known, the next step is to solve a combinatorial problem as follows: Find b x n submatrices G_i, $i = 0, 1, 2, \ldots, K-1$ with elements in GF(2) such that the linear combination of its rows satisfies the solution of Problem I. If this is not possible we have to accommodate values for $\hat{a}_i$ up to the point where the combinatorial problem matches the new solution of Problem I. When this happens, we have found the encoder that will be lexicographically optimum. This procedure does not eliminate the possibility of finding catastrophic codes. For short constraint length, it is easier to check if the proper values found for G_i results in a catastrophic code by a simple rule of linear independence among the rows of G_i, but for long constraint length an exhaustive search should be used.

Let us demonstrate the procedure by an example. Consider the case where K=2 and r=2/5. We know from Theorem 6 with q=2 that $\Phi = n.2^{b-1} = 5.2 = 10$. From Theorem 8, $d_{min} \leq 6$, then Problem I is established as follows

A) $\sum_{i=1}^{B} a_i d_{pi} \leq K \cdot \phi$ $\qquad a_1 d_{p1} + ... + a_B d_{pB} \leq 20$

B) $\sum_{i=1}^{B} a_i = 2^b - 1$ $\qquad a_1 + a_2 + ... + a_B = 3$

with $d_{p1} = 6$, $d_{p2} = 7$, $d_{p3} = 8$, and so on. From A) and B) we have

1) $a_1 = 1$ $\quad a_2 = 2$; $\quad$ 2) $a_1 = 2$ $\quad a_3 = 1$

Thus, $\hat{a}_1 = (1, 2)$ and $\hat{a}_2 = (2, 1)$ corresponding to $d_p = \{6, 7\}$ and $d_p = \{6, 8\}$ respectively. Clearly, the solution is 1). For it is the lexographically minimum 2-tuple. Therefore,

$$d_p = \{6, 7\}$$

The combinatorial problem is to find b x n matrices G_o and G_1 such that d_p is achieved. But d_{pi}, $i = 1, 2, 3$ is the sum of the branch Hamming weights leaving from and merging to the zero state (see Figure 7). Let w_i and $\tilde{w}_i$ with $1 \leq i \leq 3$, be the Hamming weights of all combinations of the rows of G_o and G_1 respectively. Clearly,

$$d_{p1} = w_1 + \tilde{w}_1, \quad d_{p2} = w_2 + \tilde{w}_2, \quad d_{p3} = w_3 + \tilde{w}_3$$

One possible solution is

$w_1 = 4$ $\quad \tilde{w}_1 = 2$

$w_2 = 3$ $\quad \tilde{w}_2 = 4$

$w_3 = 3$ $\quad \tilde{w}_3 = 4$

therefore,

$$G_o = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \qquad G_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

On the other hand, if after a number of trials there is no matching between solutions of the combinatorial problem and Problem I, the next step is to decrease by one the free distance's original value and start again the search process up to the point that a matching occurs.

6 - Search Procedure

In order to select the optimum encoder in the set $\bar{S}$ a search procedure is necessary. The search procedure used is as follows: the generating function T(z) is a polynomial in D (pairwise error probability) and z (a decoding error). The exponent of D and z specify the accumulated Hamming distance and the number of information bit errors respectively. In general, T(z) is of the form

$$T(z) = c_1 D^{a1} z^{b1} + c_2 D^{a2} z^{b2} + c_3 D^{a3} z^{b3} + ... \qquad (28)$$

where c_i, b_i, a_i, $i = 1, 2, 3, ...$ are integer valued constants.

For 2^b-ary trees the first branch from the root node has $(2^b - 1)$ branches and from each one of them follows 2^b branches. Let us consider a binary rooted tree with depth K (K branches long) as shown in Figure 8.

This tree characterizes error events when all zeros data path is assumed correct. Thus, the first branch from the root node is due to a decoded data error as "1". All subsequent branches are associated with correct decoded data "0", which moves up, and decoded data error "1" which

moves down. Whenever a decoded data error "1" occurs a variable z is appended to it.

From this characterization and since P_b is the derivative of T(z) with respect to z, the key idea behind the search procedure is to minimize the coefficients of $[d/dz]T(z)$ while maximizing the exponents of D. In general, we have to place the value of $a(K) = d_{free}$ for the shortest path (the K branches long path) when K is odd, and $a(K) = d_{free} + 1$ when K is even. Note that a(K) represents the number of tap connections among modulo 2 adders and shift registers. By a judicious choice of the branch Hamming distances along this path such that its sum equal a(K) and the maximum flow property, the taps connecting the modulo 2 adders and shift registers are set up.

The next step is to determine the branch flow value(s) for the $(K + j + 1)$ branches long path, $a(K + j + 1)$, and so, we have the following conditions: initially, set $j = 0$

A) For K odd [K even], if $a(K + j + 1)$ is smaller than $a(K + j)$ [go to C], rearrange the tap connections up to the point where its value(s) at least equal $a(k + j)$

B) If $a(K + j + 1)$ is equal or greater than $a(K + j)$, find the flow value(s) for the $(K + j + 2)$ branches long path

C) Compare $a(K + j + 2)$ with $a(K + j + 1)$, if smaller, rearrange the tap connections such that the previous value(s) do not violate the key idea behind the search procedure. If greater or equal, update j by setting $j = j + 1$ and go to C. Repeat this process until all tree branches with depth K are all checked.

This algorithm provides an easy way of finding convolutional encoders by hand calculation for any rate $r = 1/n$ and constraint length $K \leq 7$. For $K \geq 8$, a computer-aided search has to be employed using the above algorithm. These codes are shown in Tables 1 to 7. We have included new codes in addition to the ones found in [15]. Extension to long constraint length is only a matter of computer facilities.

For rates $r = b/n$, it is possible to solve for $b(K - 1) < 6$ by hand calculation. For $b(K - 1) \geq 6$ a computer-aided search is unavoidable, due to the exponential growth of the number of states. These codes are shown in Tables 8 to 11. For rates $r = 2/3$ and $3/4$, all codes shown in the respective Tables are better, in the bit error probability sense, than those in [6].

7 - Unit-Memory Byte Oriented Binary Convolutional Codes

Lee [9], has shown that convolutional codes with unity-memory, $K = 2$, and rate $r = b/n$, n a multiple of b, always achieve the largest free distance among all codes with equivalent rate $r = 1/n$ and same number of states $2^{b(K-1)}$. In other words, multiplicity of primitive rates has free distance given by

$$d_{free} \geq d_{bound}$$

where d_{bound} is the least upper bound on d_{min} of convolutional codes with constraint length K' and rate $r = b'/n'$, $\gcd(b', n') = 1$.

Using the network flow approach to this problem, an upper bound on the minimum, distance of convolutional codes with $\gcd(b, n) \geq 2$ is given. It is summarized in the following lemma.

Lemma 11 : For (K-1)-memory byte oriented convolutional codes with rate r = b/n, n a multiple of b, the minimum distance is upper bounded by

$$d_{min} \leq \left[(2^{b-1}/2^b - 1) \cdot n\right] \cdot K = d'_{bound}$$

where $[x]$, means the largest integer less than or equal to x.

Proof : From Theorem 6 with $q = 2$, we know that $\Phi = n\ 2^{b-1}$. This is the uniform, and consequently, the maximum flow attainable, which is the same as the total weight or distance of a block code with 2^b codewords each with Hamming weight n/2.
Since there are $2^b - 1$ nonzero codewords, the minimum distance of this block code is less than or equal to its average distance. Thus,

$$d \leq \left[(2^b/2^b - 1) \cdot (n/2)\right]$$

Since we have K memory elements, the minimum distance is

$$d_{min} \leq \left[(2^{b-1}/2^b - 1) \cdot n\right] \cdot K \qquad \text{Q.E.D.}$$

Due to the transcedental form of the least upper bound on the minimum distance of (K-1)-memory convolutional codes, and the equivalent convolutional codes with gcd(b', n') = 1, the only feasible way of showing that the free distance of convolutional codes is lower and upper bounded by d_{bound} (Theorem 8) and d'_{bound} (Lemma 11) is graphically. Therefore, the free distance is

$$d_{bound} \leq d_{free} \leq d'_{bound} \qquad (29)$$

$$\min_{p \geq 1} \left[(2^{p-1}/2^p - 1) \cdot (n'/b') \cdot (p + b' \cdot (K' - 1))\right] \leq d_{free} \leq \left[(2^{b-1}/2^b - 1) \cdot n\right] \cdot K$$

where gcd(b', n') = 1, and gcd(b, n) $\geq$ 2.

This simple upper bound on d_{free} is very good for short multiplicity of primitive rates. Using it, together with the selection criterion of section 5, we were able to find good unity-memory binary convolutional codes that have better performance, both in bit and byte error probabilities, than some of the codes presented in [9], [19].

These codes are shown in Table 12, with respectives rates, achievable minimum distance (bits), $d_{ach.}$, minimum distance of the equivalent convolutional code, $d_{eq.}$, and the encoder's octal representation.

8 - Conclusion

The network flow approach enabled us to identify and to put in the same context the problem of finding good nonsystematic convolutional codes as a maximal flow problem. From this characterization, we were able to extend the cardinality of the set of known good codes with short constraint length K and rate r = b/n with gcd(b, n) $\geq$ 1 where the unavoidable search was reduced many times.

Table 1

Constraint Length - K=3

rate		d_{free} ach.	d_{free} bound	octal representation
(1)	1/2	55	5	57
(1)	1/3	88	8	577
(2)	1/4	10	10	5577
(3)	1/5	13	13	55777
(3)	1/6	16	16	557777
(3)	1/7	18	18	5557777
(3)	1/8	21	21	55577777
	1/9	24	24	555777777
	1/10	26	26	5555777777
	1/11	29	29	55557777777
	1/12	32	32	555577777777
	1/13	34	34	5555577777777
	1/14	37	37	55555777777777
	1/15	40	40	555557777777777
	1/16	42	42	5555557777777777

(1) - codes found by Odenwalder, [2].

(2) - code found by Larsen [3] was 5777.

(3) - codes found independently by Daut et al., [10].

Table 2

Constraint Length - K=4

rate		d_{free} ach.	d_{free} bound	octal representation
(1)	1/2	6	6	15 17
(1)	1/3	10	10	13 15 17
(2)	1/4	13	13	13 15 15 17
(3)	1/5	16	16	13 15 15 17 17
(3)	1/6	20	20	13 13 15 15 17 17
(3)	1/7	23	23	13 13 15 15 15 17 17
(3)	1/8	26	26	13 13 15 15 15 17 17 17
	1/9	30	30	13 13 13 15 15 15 17 17 17
	1/10	33	33	13 13 13 15 15 15 15 17 17 17
	1/11	36	36	13 13 13 15 15 15 15 17 17 17 17
	1/12	40	40	13 13 13 13 15 15 15 15 17 17 17 17
	1/13	43	43	13 13 13 13 15 15 15 15 15 17 17 17 17
	1/14	46	46	13 13 13 13 15 15 15 15 15 17 17 17 17 17
	1/15	50	50	13 13 13 13 13 15 15 15 15 15 17 17 17 17 17
	1/16	53	53	13 13 13 13 13 15 15 15 15 15 15 17 17 17 17 17

(1) - codes found by Odenwalder, [2].

(2) - code found by Larsen, [3].

(3) - codes found independently by Daut et al., [10].

Table 3

Constraint Length = K = 4

	rate	d_{free} ach.	d_{free} bound	octal representation
(1)	1/2	7	8	23 35
(1)	1/3	12	12	25 33 37
(2)	1/4	16	16	25 27 33 37
(3)	1/5	20	20	25 27 33 35 37
(3)	1/6	24	24	25 27 33 35 35 37
(3)	1/7	28	28	25 27 27 33 35 35 37
(3)	1/8	32	32	25 25 27 33 33 35 37 37
	1/9	36	36	25 25 27 33 33 35 35 37 37
	1/10	40	40	25 25 25 33 33 33 35 37 37 37
	1/11	44	44	25 25 25 27 33 33 33 35 37 37 37
	1/12	48	48	25 25 25 27 33 33 33 35 35 37 37 37
	1/13	52	52	25 25 25 27 27 33 33 33 35 35 37 37 37
	1/14	56	56	25 25 25 27 27 33 33 33 35 35 35 37 37 37
	1/15	60	60	25 25 25 25 27 33 33 33 33 35 35 37 37 37 37
	1/16	64	64	25 25 25 25 27 33 33 33 33 35 35 35 37 37 37 37

(1) - codes found by Odenwalder, [2].

(2) - code found by Larsen, [3].

(3) - codes found independently by Daut et al., [10].

Table 4

Constraint Length = K = 6

	rate	d_{free} ach.	d_{free} bound	octal representation
(1)	1/2	8	8	53 75
(1)	1/3	13	13	47 53 75
(2)	1/4	18	18	53 67 71 75
(3)	1/5	22	22	51 55 67 73 77
(3)	1/6	27	27	47 55 57 65 73 75
(3)	1/7	32	32	47 53 57 65 67 75 75
(3)	1/8	36	36	47 51 57 57 65 67 73 75
	1/9	41	41	51 57 57 65 65 67 71 73 77
	1/10	45	45	45 47 53 55 65 67 73 73 75 77
	1/11	50	50	47 53 57 57 65 65 65 71 73 75 77
	1/12	54	54	51 53 55 57 65 65 65 67 71 73 77 77
	1/13	59	59	47 51 53 53 57 65 65 67 67 73 75 75 77
	1/14	63	63	47 51 53 57 57 63 65 65 65 67 73 75 75 77
	1/15	68	68	47 51 53 57 57 63 65 65 65 67 73 75 75 75 77
	1/16	72	72	47 53 57 57 63 65 65 65 67 67 71 71 73 75 75 75

(1) - codes found by Odenwalder, [2].

(2) - code found by Larsen, [3].

(3) - codes found independently by Daut et al., [10].

Table 5

Constraint Length - K = 7

	rate	d_{free} ach.	d_{free} bound	octal representation
(1)	1/2	10	10	133 171
(1)	1/3	15	15	133 145 175
(2)	1/4	20	20	135 135 147 163
(3)	1/5	25	25	131 135 135 145 175
(3)	1/6	30	30	135 135 137 151 163 173
(3)	1/7	36	36	135 135 137 145 147 165 173
(3)	1/8	40	40	111 135 135 137 147 153 165 173
	1/9	46	46	117 123 127 137 153 155 165 171 175
	1/10	51	51	115 115 127 133 137 145 157 165 173 175
	1/11	56	56	115 115 125 127 137 151 157 163 173 173 175
	1/12	61	61	115 117 125 127 133 135 153 157 167 171 171 175
	1/13	66	66	115 117 125 127 133 135 153 155 157 167 171 171 175
	1/14	72	72	115 117 125 127 133 135 153 155 157 167 171 171 175 175
	1/15	76	76	117 127 127 131 131 135 135 153 153 155 157 171 171 175 175
	1/16	82	82	117 127 127 131 131 135 135 137 153 153 155 157 171 171 175 175

(1) - codes found by Odenwalder, [2].

(2) code found by Larsen, [3].

(3) - codes found independently by Daut et al., [10].

Table 6

Constraint Length - K = 8

	rate	d_{free} ach.	d_{free} bound	octal representation
(1)	1/2	10	11	247 371
(1)	1/3	16	16	225 331 367
(2)	1/4	22	22	235 275 313 357
(3)	1/5	28	28	233 257 271 323 357
(3)	1/6	34	34	235 253 313 331 357 375
(3)	1/7	40	40	235 253 275 313 331 357 375
(3)	1/8	45	45	235 253 275 275 313 331 357 371
	1/9	51	51	251 265 267 273 311 337 337 337 345
	1/10	56	56	225 225 273 275 275 313 317 357 363 365
	1/11	62	62	225 257 267 277 277 315 327 331 351 355 363
	1/12	68	68	225 257 267 277 277 315 327 331 345 351 355 363
	1/13	74	74	225 257 257 267 277 277 315 327 331 345 351 355 363
	1/14	80	80	225 257 257 267 267 277 277 315 327 331 345 351 355 363
	1/15	85	85	225 235 257 257 267 267 277 277 315 327 331 345 351 355 363
	1/16	91	91	231 255 257 257 267 267 275 277 277 315 327 331 345 351 355 363

(1) - codes found by Odenwalder, [2].

(2) - code found by Larsen, [3].

(3) - codes found independently by Daut et al., [10].

Table 7

Constraint Length - K = 9

	rate	d_{free} ach.	d_{free} bound	octal representation
(1)	1/2	12	12	561 753
(2)	1/3	18	18	557 663 711
(2)	1/4	24	24	463 535 733 745
	1/5	31	31	467 531 535 675 747
	1/6	37	37	475 545 553 677 711 727
	1/7	44	44	457 463 525 673 737 751 755
	1/8	50	50	513 553 567 625 647 671 717 775
	1/9	56	56	471 515 527 537 653 661 673 747 775
	1/10	62	62	467 537 547 571 625 653 677 711 725 773

(1) - code found by Odenwalder, [2].

(2) - codes found by Larsen, [3].

Table 8

Constraint Length - K = 2

	rate	d_{free} ach.	d_{free} bound	octal representation
(4)	2/3	3	4	Go - 36 G1 - 25
(3)	2/5	6	6	Go - 31 26 G1 - 25 33
(3)	2/7	9	9	Go - 147 131 G1 - 066 171
	2/9	12	12	Go - 733 547 G1 - 714 473
	2/11	14	14	Go - 3163 1755 G1 - 1077 3760
(4)	3/4	4	4	Go - 03 05 16 G1 - 11 12 14
(3)	3/5	5	5	Go - 23 25 32 G1 - 30 06 15
(3)	3/7	8	8	Go - 170 036 063 G1 - 125 071 162
(3)	3/8	8	8	Go - 360 074 227 G1 - 146 073 303
	3/10	11	11	Go - 1740 0374 0547 G1 - 0770 0037 1313
	3/11	12	12	Go - 3434 0770 2133 G1 - 2525 2752 1361
(3)	4/5	3	4	Go - 16 22 02 13 G1 - 03 06 14 30
(3)	4/7	6	6	Go - 170 036 115 055 G1 - 140 070 056 041
	4/9	8	8	Go - 740 146 264 453 G1 - 170 147 716 444
	4/11	11	11	Go - 3740 0374 0707 1625 G1 - 3700 0370 0516 1225

Table 9

Constraint Length - K = 3

	rate	d_{free} ach.	d_{free} bound	octal representation
(4)	2/3	5	5	Go - 36 G1 - 27 G2 - 65
(3)	2/5	10	10	Go - 16 23 G1 - 27 31 G2 - 25 33
(3)	2/7	14	14	Go - 147 171 G1 - 066 115 G2 - 155 172
	2/9	18	18	Go - 733 547 G1 - 714 473 G2 - 157 723
	2/11	22	22	Go - 3163 1755 G1 - 1077 3760 G2 - 3617 3163
(4)	3/4	6	6	Go - 11 05 03 G1 - 11 11 16 G2 - 05 12 06
	3/5	7	8	Go - 23 35 22 G1 - 25 06 16 G2 - 25 31 36
	3/7	12	12	Go - 131 036 063 G1 - 125 071 162 G2 - 056 033 126
	3/8	13	13	Go - 360 074 227 G1 - 146 073 303 G2 - 370 066 033
	3/10	16	16	Go - 1740 0374 0547 G1 - 0770 0037 1313 G2 - 0525 0553 1660
	3/11	18	18	Go - 3434 0770 2133 G1 - 2525 2752 1361 G2 - 3314 0476 2547

Table 10

Constraint Length - K = 4

	rate	d_{free} ach.	d_{free} bound	octal representation
(4)	2/3	7	7	Go - 36 G1 - 65 G2 - 25 G3 - 63
(3)	2/5	12	12	Go - 16 23 G1 - 27 31 G2 - 25 33 G3 - 34 07
(3)	2/7	18	18	Go - 147 170 G1 - 126 055 G2 - 171 147 G3 - 131 156
	2/9	23	23	Go - 732 547 G1 - 714 477 G2 - 167 732 G3 - 473 335
	2/11	28	28	Go - 3163 1755 G1 - 1137 1762 G2 - 1656 3563 G3 - 2746 1437
(4)	3/4	8	8	Go - 11 05 03 G1 - 03 07 13 G2 - 06 01 10 G3 - 06 14 11
	3/5	10	11	Go - 23 35 24 G1 - 15 06 32 G2 - 25 31 36 G3 - 21 12 34
	3/7	14	15	Go - 130 036 063 G1 - 127 071 162 G2 - 057 033 147 G3 - 114 121 052

rate	d_{free} ach.	d_{free} bound	octal representation
3/8	17	17	Go - 344 076 227 G1 - 146 073 303 G2 - 330 063 033 G3 - 313 125 346
3/10	22	22	Go - 1740 0370 1167 G1 - 0770 0037 1313 G2 - 0525 0553 1662 G3 - 1272 0317 0546
3/11	24	24	Go - 1524 0372 3113 G1 - 2275 1646 3115 G2 - 1742 0672 2457 G3 - 1343 2336 1534

Table 11

Constraint Length = K = 5

rate	d_{free} ach.	d_{free} bound	octal representation
(4) 2/3	8	8	Go - 53 G1 - 62 G2 - 37 G3 - 16 G4 - 52
2/5	14	15	Go - 32 15 G1 - 13 37 G2 - 34 22 G3 - 15 31 G4 - 26 13
2/7	21	21	Go - 113 076 G1 - 073 055 G2 - 136 145 G3 - 131 156 G4 - 174 073
2/9	27	27	Go - 536 647 G1 - 554 333 G2 - 374 672 G3 - 433 735 G4 - 762 175
2/11	34	34	Go - 3056 1771 G1 - 3227 1555 G2 - 0337 2742 G3 - 3456 3563 G4 - 3346 2437

(4) - codes found by Paaske, [6].

(3) - codes found independently by Daut et al., [10].

Table 12

Constraint Length - K = 2

	rate	d_{free} ach.	d_{free} bound	octal representation
(5)	2/4	5	5	Go - 14 07 G1 - 16 03
(5)	2/6	8	8	Go - 74 17 G1 - 36 47
(5)	2/8	10	10	Go - 370 037 G1 - 174 237
	2/10	13	13	Go - 1760 0177 G1 - 0774 1037
	2/12	16	16	Go - 7760 0377 G1 - 1774 3761
(5)	3/6	6	6	Go - 70 36 13 G1 - 34 07 62
	3/9	10	10	Go - 760 076 651 G1 - 370 742 147
	3/12	13	13	Go - 7740 7036 3147 G1 - 4614 7660 0367
(6)	4/8	8	7	Go - 160 074 063 252 G1 - 265 232 170 116

(5) - octal representation were not shown in [9].

(6) - code found by Lee [9], and Lauer [19], has poorer bit error probability than the one shown.

REFERENCES

[1] G.D. Forney, "Convolutional Codes I: Algebraic Structure," IEEE Trans. Inform. Theory, Vol. IT-16, pp. 720-738, November 1970.

[2] J.P. Odenwalder, "Optimal Decoding of Convolutional Codes," Ph.D. Dissertation, University of California, Los Angeles, 1970.

[3] K.J. Larsen, "Short Convolutional Codes with Maximum Free Distance for Rates 1/2, 1/3, 1/4," IEEE Trans. Inform. Theory, Vol. IT-19, pp. 371-372, May 1973.

[4] R. Johannesson and E. Paaske, "Further Results on Binary Convolutional Codes with an Optimum Distance Profile," IEEE Trans. Inform. Theory, Vol. IT-24, pp. 264-268, March 1978.

[5] L.R. Bahl, and F. Jelinek, "Rate 1/2 Convolutional Codes with Complementary Generators," IEEE Trans. Inform. Theory, Vol. IT-17, pp. 718-727, November 1971.

[6] E. Paaske, "Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 683-689, September 1974.

[7] J.B. Cain, G.C. Clark, Jr., and J.M. Geist, "Punctured Convolutional Codes of Rate (n-1)/n and Simplified Maximum Likelihood Decoding," IEEE Trans. Inform. Theory, Vol. IT-25, pp. 97-100, January 1979.

[8] J.L. Massey, and D.J. Costello, Jr., "Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications," IEEE Trans. Comm. Technol., Vol. COM-19, pp. 806-813, October 1971.

[9] L.N. Lee, "Short Unity-Memory Byte-Oriented Binary Convolutional Codes having Maximal Free Distance," IEEE Trans. Inform. Theory, Vol. IT-22, pp. 349-352, May 1976.

[10] D.G. Daut, J.W. Modestino, and L.D. Wismer, "New Short Constraint Length Convolutional Codes Constructions for Selected Rational Rates," IEEE Trans. Inform. Theory, Vol. IT-28, pp. 794-800, September 1982.

[11] H. Kwakernaak, and R. Sivan, Linear Optimal Control Systems, McGraw Hill, 1972.

[12] J.S. Meditch, Stochastic Optimal Linear Control, McGraw-Hill, 1969.

[13] A.J. Viterbi, and J.K. Omura, Principles of Digital Communication and Coding, McGraw-Hill, 1979.

[14] E.I. Jury, Inners and Stability of Dynamic Systems, John Wiley & Sons, 1974.

[15] R. Palazzo, Jr., "Analysis of Periodic Linear and Nonlinear Trellis Codes," Ph.D. Dissertation, University of California, Los Angeles, 1984.

[16] J.A. Heller, "Short Constraint Length Convolutional Codes," Jet Propulsion Lab., California Inst. Technol., Space Programs Summary 37-54, Vol. 3, pp. 171-177, Oct./Nov. 1968.

[17] L.R. Ford, and D.R. Fulkerson, Flow in Networks, Princeton University Press (Princeton, N.J.), 1962.

[18] W.W. Peterson, and E.J. Weldon, Jr., Error Correcting Codes, MIT Press, Cambridge, Mass., 1972.

[19] G.S. Lauer, "Some Optimal Partial Unit-Memory Codes," IEEE Trans. Inform. Theory, Vol. IT-25, pp. 240-243, March 1979.

[20] S. Lin, and D.J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice Hall, 1983.

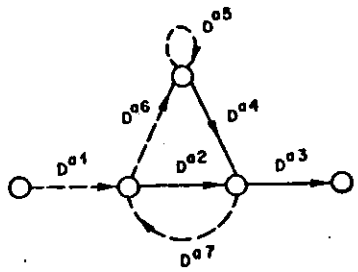

Figure 1 - Split state diagram.

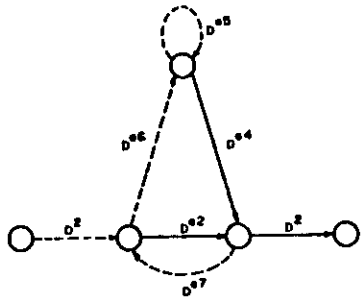

Figure 2 - Partition $\tilde{p}(2, \tilde{A}, 2)$.

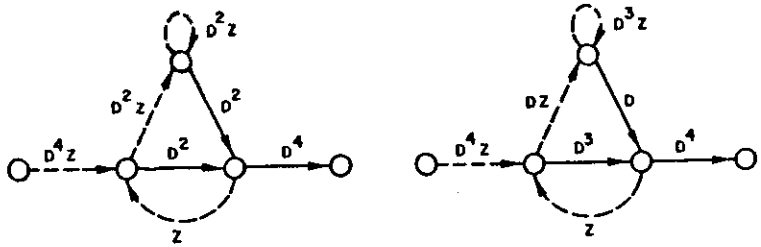

Figure 3 - Split state diagram for k = 3 and r = 1/4.

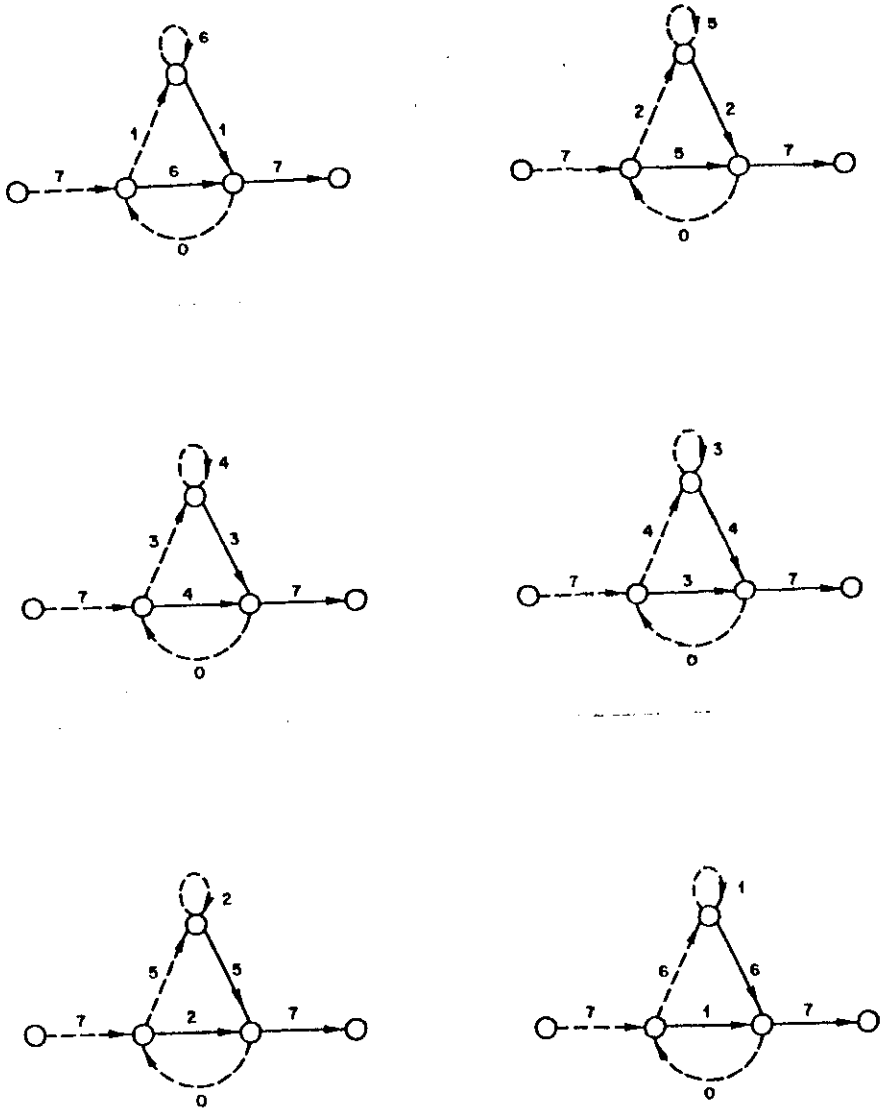

Figure 4 - Split state diagrams for k = 3 and r = 1/7.

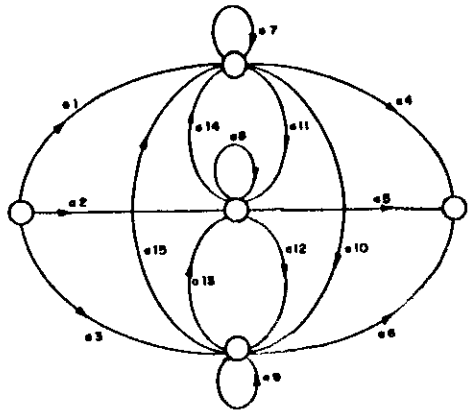

Figure 5 - Split state diagram for k = 2 and r = 2/3.

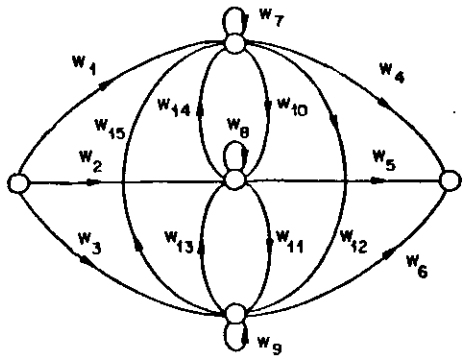

Figure 6 - Split state diagram for k = 2 and r = b/n.

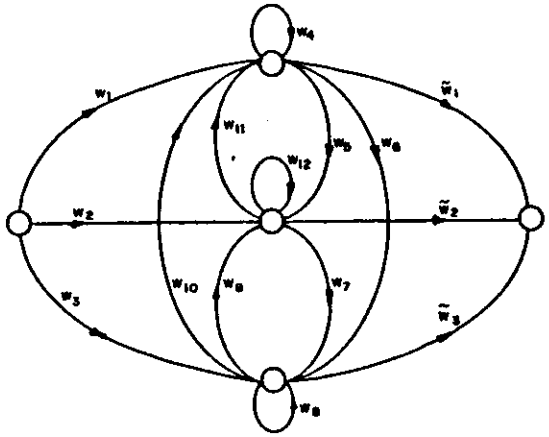

Figure 7 - Split state diagram for k = 2 and r = 2/5.

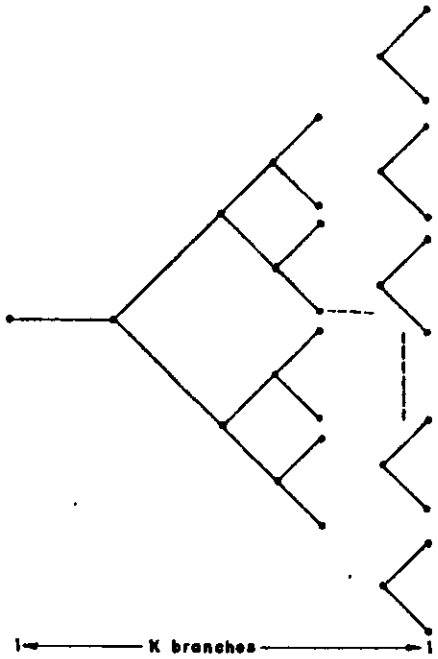

Figure 8 - Error event binary tree.

9 - Distance Properties of Convolutional Codes

The performance of convolutional codes is related to the decoding algorithm and with the distance properties of the code. In order to be able to evaluate the performance of such codes, we are going to introduce the most frequently used distance measures. They are the free distance, d_{∞}, the column distance function d_i, and the minimum distance d_{min}.

Among these distances, the free distance is the most important one. It is defined as

$$d_{\infty} = \min \{ d(v', v'') : u' \neq u'' \}$$

where v' and v'' are the coded sequences corresponding to the information sequences u' and u'', respectively. Thus, the free distance is the minimum distance between any v' and v''. Since convolutional codes are linear, we have that

$$d_{\infty} = \min \{ w(v' \oplus v'') : u' \neq u'' \}$$

$$= \min \{ w(v) : u \neq 0 \}$$

$$= \min \{ w(u.G) : u \neq 0 \}$$

From this, it follows that d_{∞} is the minimum weight coded sequence of any length resulting from pairwise comparison between any path in the trellis that leaves and comes back to the zero state.

Regarding the column distance function, let v[o,i] be the truncated coded sequence, that is,

$$v[o.i] = (vo, v1, v2, ..., vi)$$

and u[o,i] the corresponding information sequence, that is,

$$u[o,i] = (uo, u1, u2, \ldots, ui)$$

The column distance function of order i, d_i , is defined as

$$d_i = \min \{ d(v'[o,i], v''[o,i]) : u'[o] \neq u''[o] \}$$

$$= \min \{ w(v[o,i]) : u[o] \neq 0 \}$$

Thus, d_i is the minimum weight of the coded sequence of length (i+1) with u[o] ≠ 0. On the other hand, if we take into consideration the generator matrix of the code, then

$$v[o,i] = u[o,i].G[o,i]$$

$$G[o,i] = \begin{bmatrix} Go & G1 & \ldots\ldots & Gi \\ & Go & \ldots\ldots & Gi-1 \\ & & Go \ldots & Gi-2 \\ & & & \ldots\ldots \\ & & & Go \end{bmatrix} \quad \text{for } i \leq m$$

or

$$G[o,i] = \begin{bmatrix} Go & G1 & G2 & \ldots\ldots & Gm & & \\ & Go & G1 & & Gm-1 & Gm & \\ & & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \\ & & & Go & G1 \ldots\ldots & \ldots\ldots & Gm \\ & & & & \ldots\ldots & \ldots\ldots & \\ & & & & & & Go \end{bmatrix}$$

for i > m. Therefore,

$$d_i = \min \{ w(u[o,i].G[o,i]) : u[o] \neq 0 \} \qquad (9.1)$$

will depend only on the (n+i) columns of G. This is the reason it is called column distance function. It should be emphasized that d_i does not decrease with i.

The column distance function has two cases which deserve be taken into consideration. They are respectively the case when i = m and when i goes to infinity. When i = m, d_i is called the minimum distance of the convolutional code and its usual representation is d_{min} . Looking at (9.1) one sees that d_{min} is the coded sequence with minimum weight over the first constraint length.

On the other hand, when i goes to infinity, $\lim_{i \to \infty} d_i$ is the minimum weight of the coded sequence when the first block of the information sequence is not zero. Therefore, we have

$$d_\infty = \lim_{i \to \infty} d_i$$

this means that eventually d_i will achieve d_∞ and will not increase anymore.

Linear Unit-Memory Codes over GF(q)

10. INTRODUCTION

The problem of finding good specific time invariant convolutional codes has received a lot of attention by many researchers. From these investigations a number of clever algorithms were proposed. However, a general method for solving this problem has not been presented yet.

In order to shed some light into it, the purpose of this paper aims at showing from a combinatorial point of view the equivalence of the problem of finding good linear unit-memory codes (UM) and consequently good specific nonsystematic convolutional codes over GF(q), with q a prime or power of a prime, with the problem of solving a knapsack. It is known that this problem is nondeterministic polynomial (NP) complete, [3], and so is presumed to be hard, at least in the worst case. Consequently, justifying why a complete answer to the original problem has not been provided.

11 PRELIMINARIES

Let $\underline{x}_t$ be a b-dimensional data input vector, let $\underline{y}_t$ be an n-dimensional encoded vector over GF(q) defined as follows

$$\underline{x}_t = (x_{t1}, x_{t2}, x_{t3}, \ldots, x_{tb})$$

and

$$\underline{y}_t = (y_{t1}, y_{t2}, y_{t3}, \ldots, y_{tn})$$

respectively.

Let $G_0(t)$ and $G_1(t)$ be b x n time varying matrices with elements in GF(q). A (b,n) unit memory code is defined by an encoding rule of the form

$$\underline{y}_t = \underline{x}_t \cdot G_0(t) + \underline{x}_{t-1} \cdot G_1(t) \qquad t \geq 0 \text{ and } \underline{x}_{-1} = \underline{0}$$

where $\underline{0}$ is the all zeros row matrix, and all operations are also in GF(q). If $G_j(t) = G_j$ for all t, and j, then the code is said to be time invariant. If $G_j(t) = G_j(t+T)$ for all j and some positive T, then the code is said to be periodically time varying. To show the equivalence it is sufficient that the code be time invariant, so this will be the assumption from now on.

We denote the encoder of a UM code with rate r = b/n and memory ν by b parallel $(\nu+1)$-stage shift registers. Let d_∞ be the free distance, that is, the smallest Hamming weight between pairwise output sequences resulting from distinct input sequences. Equivalently, the unrestricted minimum distance.

It has been shown in [2] that in general optimum specific nonsystematic convolutional codes and optimum linear unit-memory codes satisfy the maximum flow and conservation of flow properties when we look upon convolutional codes as a combinatorial optimization problem. Here by optimum it is meant a code whose weight enumerator function has the least number of codewords attaining d_∞. However, if such codes have the same number of codewords for the same d_∞, then the code whose next term of the weight enumerator function has the least number of codewords with distance $d_\infty + 1$ is considered the optimum. If there is a tie with the coefficients of the term $d_\infty + 1$, a comparison is made with the coefficients of the term for $d_\infty + 2$, and so on. By maximum flow, Φ, it is meant the sum of the branch Hamming weights leaving from or going into a state, in a split state diagram representation, satisfying

$$\Phi = n(q-1)q^{b-1} \qquad (1)$$

Note that (1) represents the total weight of a q-ary block code with block length n and q^b codewords. By conservation of flow it is meant that the flow ϕ going into and leaving any state, but the zero state, in a split state diagram representation, are equal. Equivalently, maximum flow and conservation of flow properties imply that any column of G_o and G_1 can not be all zeros.

12. EQUIVALENCE TO THE KNAPSACK PROBLEM

Since linear $(b\nu, n\nu)$ UM codes are equivalent to (b, n, ν) convolutional codes, [1], with at least the same free distance, we have that UM codes form a special class of codes by itself.

Let the tap connections between the shift-registers and mod q adders be arbitrary for a given rate $r = b/n$. This generates a class of linear UM codes where each code can be represented by a split state diagram as shown in Fig. 1.

Let $\underline{d}$ be the lexographic representation of the branch Hamming weights w_{oi} and w_{io} from the zero state to state i, and from state i to state zero, respectively, (see Fig. 1), that is,

$$\underline{d} = \{d_1, d_2, d_3, \ldots, d_{q^b-1}\}$$

with

$$d_i = w_{oi} + w_{io} \quad , \qquad 1 \leq i \leq q^b - 1$$

The problem here is to find each one of the w_{oi} and w_{io}. Consequently, knowing w_{oi} and w_{io} allow us to determine the matrices G_o and G_1 respectively. Hence, the only parameter we have to know is the free distance of the code due to the fact that in general $d_i = d_\infty + i - 1$ for $1 \leq i \leq q^b - 1$. Thus, the next two Theorems establishes upper bounds on the free distance of convolutional and unit-memory codes which will be useful in setting up $\underline{d}$.

Theorem 1 : For any convolutional code over GF(q) with memory ν and rate $r = b/n$, $\gcd(b, n) = 1$, the minimum distance is upper bounded by

$$d_{min}/(n/b) \leq \min_{p \geq 1} \left[(q-1) \cdot (q^{p-1}/q^p - 1) \cdot (p + b \cdot \nu)\right] \qquad (2)$$

where p is an integer, and $[a]$ means the largest integer less than or equal to a.

Proof : It is known that a terminated q-ary convolutional code with W information symbols is a group code with an $W \times n \cdot (W + \nu)$ dimensional generator matrix for rate $r = 1/n$. For rate $r = b/n$, $\gcd(b,n) = 1$, since we have b parallel $(\nu+1)$ stage shift registers, in order to terminate a code we need to insert $b \cdot \nu$ known digits. The total length of the information bits is $b \cdot W$, and so, the generator matrix has now $b \cdot W$ rows and $(n/b) \cdot [b \cdot W - b \cdot \nu]$ columns.

For q-ary group codes, the Hamming distance between codewords is equivalent to the weight of the nonzero codewords. Hence, if all $q^{b.W}$ codewords are arranged as rows of a matrix then the total weight, W_T, of the $q^{b.W}$ codewords is upper bounded by

$$W_T \leq (q-1)q^{b.W-1} \cdot (n/b) \cdot (b \cdot W + b \cdot \nu)$$

Since $q^{b.W} - 1$ codewords are nonzero and their minimum weight must

he less than or equal to their average weight, the minimum distance satisfies

$$d_{min} \leq (q-1) \cdot (q^{b \cdot W-1}/q^{b \cdot W}-1) \cdot (n/b) \cdot (b \cdot W + b \cdot \nu)$$

Since this bound holds for any $b \cdot W$, it also holds for $p < b \cdot W$. What we really want is the least upper bound. This is achieved if we minimize the right hand side with respect to p. Thus,

$$d_{min} \leq \min_{p \geq 1} \left[(q-1) \cdot (q^{p-1}/q^{p}-1) \cdot (n/b) \cdot (p + b \cdot \nu)\right] \qquad \text{Q.E.D.}$$

Theorem 2 : For ν-memory byte oriented convolutional codes over GF(q) with rate $r = b/n$, $\gcd(b,n) \geq 2$, the minimum distance is upper bounded by

$$d_{min} \leq \left[(q-1) \cdot (q^{b-1}/q^{b}-1) \cdot n\right] \cdot (\nu+1) \qquad (3)$$

Proof : From (1), the maximum flow is $\Phi = n(q-1)q^{b-1}$. But this is the total weight of a block code with q^b codewords each one with an average Hamming weight $(q-1) \cdot (n/q)$.

Since there are $q^b - 1$ nonzero codewords, the minimum distance of this block code is less than or equal to its average distance. Thus,

$$d \leq \left[(q-1) \cdot (q^{b}/q^{b}-1) \cdot (n/q)\right]$$

Since we have ν memory elements, the minimum distance is

$$d_{min} \leq \left[(q-1) \cdot (q^{b-1}/q^{b}-1) \cdot n\right] \cdot (\nu+1) \qquad \text{Q.E.D.}$$

It should be noted that the least upper bound on (2) or (3) is

the unrestricted minimum distance, that is, d_∞. Since the set $\underline{d}$ is lexographical, we have $d_1 = d_\infty$ and in general $d_i = d_\infty + i - 1$ for $1 \leq i \leq q^b - 1$, where d_∞ is obtained from (2) or (3) depending upon the case in consideration. From the maximum flow property, equation (1) and Figure 1, we have that

$$\sum_{i=1}^{q^b-1} a_i \cdot d_i = (\nu+1) \cdot \Phi \qquad (4)$$

where $\nu = 1$ for unit-memory codes, and a_i accounts for the number of times the value d_i appears. Using a vector representation for the elements of (4), we have

$$\underline{a} * \underline{d} = \Phi \qquad (5)$$

where $*$ represents dot product. However, for a linear UM code with fixed rate $r = b/n$, Φ and $\underline{d}$ are known by using (1) and {(2) or (3)}, respectively. Therefore, solving (5) is to find $\underline{a}$. Clearly, (5) is a mathematical characterization of a Knapsack Problem. Therefore, finding good UM codes is equivalent to solving a knapsack.

If the a_i's are known so are the G_o and G_1 matrices, since this translates the fact that we know the Hamming weights of the rows of each matrix. Therefore, solving (5) for linear UM codes with large values of b, the data input length, is equivalent to finding among the at most q^b possible subsets of solutions the ones which will lead to the good codes. This exponential form with the data input length of the possible subsets of solutions is the computational complexity of the best known algorithms for solving knapsack problems.

These subsets are defined as $A_i = \{(w_{o1}, w_{o2}, \ldots, w_{oq^b-1})$,

$(w_{1o}, w_{2o}, \ldots, w_{q^b-1o})\}$, where w_{oj} and w_{jo} are the branch Hamming weights from the zero state to the jth state and the branch Hamming weights from the jth state back to the zero state, respectively, for $1 \leq i \leq q^b$. Note that in each subset there are $(q^b - 1)$ equivalent codes.

13. EXAMPLES

Let us consider an example of a UM code over GF(2) with rate $r = 2/4$, where (5) is easily solved. This code is equivalent to a convolutional code with memory $\nu = 2$ and rate $r = 1/2$. Finding a good UM code is equivalent to solving (5) for a_i, that is,

$$a_1 \cdot d_1 + a_2 \cdot d_2 + a_3 \cdot d_3 = 16 \qquad (6)$$

From (2) or (3), this UM code has $d_\infty = 5$, thus $d_1 = 5$, $d_2 = 6$, and $d_3 = 7$. Clearly, the solution of (6) is $a_1 = 2$ and $a_2 = 1$.

The 4 possible subsets of solutions which satisfy (6) are $A_1 = \{(4,3,1), (1,3,4)\}$, $A_2 = \{(4,2,2), (1,3,4)\}$, $A_3 = \{(4,2,2), (2,3,3)\}$, and $A_4 = \{(2,3,3), (3,2,3)\}$. Although A_3 and A_4 subsets lead to codes with the same d_∞, A_4 is the optimum. Therefore, one possible solution for G_o and G_1 is given by

$$G_o = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \qquad G_1 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

where "1" implies that there is a tap connecting a shift register to a modulo 2 sum, and "0" otherwise. Since the encoder is represented by 2 parallel 2-shift registers, we have that the first and second rows of

$G_o(G_1)$ represent tap connections between the first (second) stage shift registers in the parallel arrangement to the modulo 2 adders.

As another example, consider the UM code with rate $r = 18/36$. This code is equivalent to a convolutional code with memory $\nu = 18$ and rate $r = 1/2$. Finding a good UM code is equivalent to solving the following equation.

$$a_1 \cdot d_1 + \ldots + a_{262,143} \cdot d_{262,143} = 9,437,184$$

for a_i, with $d_1 = 23$, $d_2 = 24$, up to $d_{262,143} = 262,165$. Thus, solving the above equation is equivalent to finding the solution(s) among the at most 2^{18} possible subsets of solutions. In general, for small values of b ($b \leq 4$) the knapsack is easily solved and consequently good UM codes can be found by hand calculation.

14. CONCLUSION

We have shown from the combinatorial point of view that the problem of finding good convolutional codes is equivalent to the problem of solving a knapsack. The difficulty of this problem lies entirely on the values of the data input length, b, since this knapsack has at most q^b possible subsets of solutions.

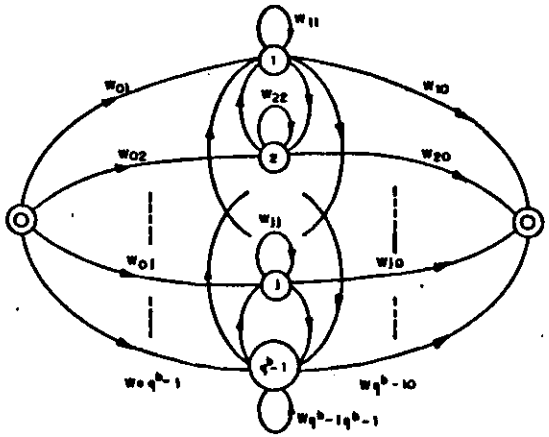

Figure 1 - Split state diagram for UM codes with fixed r = b/n.

REFERENCES

[1] L.N. Lee, "Short, unit-memory, byte-oriented, binary convolutional codes having maximal free distance," IEEE Trans. Inform. Theory, Vol. IT-22, pp. 349-352, May 1976.

[2] R. Palazzo, Jr., "Analysis of periodic linear and nonlinear trellis codes" Ph.D. Dissertation, University of California, Los Angeles, 1984.

[3] M.R. Garey, and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness San Francisco: W.H. Freeman, 1979.