

# New Syndrome Decoding Techniques for the $(n, k)$ Convolutional Codes

I. S. Reed

Department of Electrical Engineering  
University of Southern California

T. K. Truong

Communications Systems Research Section

*This paper presents a new syndrome decoding algorithm for the  $(n, k)$  convolutional codes (CC) which differs completely from an earlier syndrome decoding algorithm of Schalkwijk and Vinck. The new algorithm is based on the general solution of the syndrome equation, a linear Diophantine equation for the error polynomial vector  $E(D)$ . The set of Diophantine solutions is a coset of the CC. In this error coset a recursive, Viterbi-like algorithm is developed to find the minimum weight error vector  $\hat{E}(D)$ . An example, illustrating the new decoding algorithm, is given for the binary nonsystematic  $(3, 1)$  CC.*

## I. Introduction

In this paper, a new syndrome decoding algorithm is developed which is analogous to the syndrome decoding algorithm of block codes. This new decoding method differs completely from that invented in 1976 by Schalkwijk and Vinck (Ref. 1).

In order to develop the new syndrome decoding algorithm it is necessary to first review and further treat the algebraic nature and structure of convolutional codes (CC). This background is then used to find the general solution of the syndrome equations for the error polynomial vector  $e(D)$  where  $D$  is the unit delay operator. In particular it is shown that these syndrome equations are linear Diophantine equations over the ring of polynomials  $F[D]$  in  $D$  and with coefficients in a finite field. The method for solving linear Diophantine equations for the integers is generalized and used to solve the syndrome equations for  $e(D)$ .

The set of Diophantine solutions of the syndrome equations constitutes a coset of the convolutional code space or group. The problem of syndrome decoding is to find the minimum-weight polynomial vector  $\hat{e}(D)$  of this coset to subtract from the received polynomial vector  $z(D)$  to yield an estimate  $\hat{y}(D)$  of the transmitted polynomial vector. In order to find  $\hat{e}(D)$  efficiently, a new recursive, Viterbi-like algorithm is devised. This new syndrome decoding algorithm is presented in this paper by example.

For a fixed convolutional code the new recursive syndrome decoder for a CC appears to be comparable in complexity to the Viterbi decoder except that in the new decoder fewer comparisons are required and the control logic is considerably simpler. However, if one wishes to design a CC decoder for several different rate codes of the same constraint length, it appears that the principles of the new syndrome decoder may yield a simpler system than could be achieved using the Viterbi

technique. For variable rate communication systems that utilize convolutional codes the new syndrome decoding concept appears to have an advantage over the standard Viterbi decoding techniques. A precise quantification of this comparison is a topic for future study.

## II. Algebraic Structure of Convolutional Codes

Let  $a_0, a_1, a_2, \dots$  be any sequence of symbols from the finite field  $F = GF(q)$  of  $q$  elements. Further let  $D$  be the unit delay operator.  $D$  operates on a function  $x(n)$  of discrete time  $n$  in accordance with the definition

$$Dx(n) = x(n-1) \quad (1)$$

for all  $n$ . In terms of  $D$  the sequence  $\{a_j\}$  is conveniently represented by what is called its  $D$ -transform,

$$A(D) = a_0 + a_1 D + a_2 D^2 + \dots, \quad (2)$$

in powers of the operator  $D$ .

The input to a convolutional encoder is a set of  $k$  discrete-time input sequences. In terms of  $D$ -transforms this input is represented by the vector

$$x(D) = [x_1(D), x_2(D), \dots, x_k(D)] \quad (3)$$

where  $x_j(D) = x_{0j} + x_{1j}D + x_{2j}D^2 + \dots$  for  $(j = 1, 2, \dots, k)$  with coefficients in  $GF(q)$ . (see Refs. 2, 3).

Very simply, an encoder for a CC is some *linear* sequential circuit over the finite field  $GF(q)$  with vector input  $x(D)$  and vector output

$$y(D) = [y_1(D), y_2(D), \dots, y_n(D)] \quad (4)$$

where  $n > k$  and  $y_r(D) = y_{0r} + y_{1r}D + y_{2r}D^2 + \dots$  for  $1 \leq r \leq n$ . For the standard  $(n, k)$  convolutional code  $((n, k)$  CC) the linear relationship between the input and output is assumed to have finite memory so that it can be expressed as a matrix convolution of form

$$y(D) = x(D)G(D) \quad (5)$$

where  $G(D)$  is a  $k \times n$  matrix of polynomials of finite degree in  $D$  over  $GF(q)$ .  $G(D)$  in (5) is usually called the generating matrix of the  $(n, k)$  CC of  $k/n$  rate CC.  $G(D)$  has the specific form, as a  $k \times n$  matrix,

$$G(D) = \begin{bmatrix} g_{11}(D), & g_{12}(D), & \dots, & g_{1n}(D) \\ g_{21}(D), & g_{22}(D), & \dots, & g_{2n}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1}(D), & g_{k2}(D), & \dots, & g_{kn}(D) \end{bmatrix} \quad (6)$$

The maximum degree  $M$  of the polynomials in  $G(D)$  is called the *memory*, and the constraint length of the code is  $L = M + 1$ .

The elements of  $G(D)$  in (6) are polynomials in  $D$  over the finite field  $F = GF(q)$ . The set of all such polynomials in  $D$  over a field  $F$  is an infinite ring  $F[D]$  as well as an integral domain since it has no divisors of zero. It can also be demonstrated that  $F[D]$  is a Euclidean ring, e.g., see Ref. 4, sections 3.7 and 3.9.

If the elements of  $G(D)$  in (6) are restricted to members of field  $F$ , the rows of  $G(D)$  generate an  $n$ -dimensional vector space over field  $F$ . However, since  $G(D)$  is a  $k \times n$  matrix with elements in  $F[D]$ , the integral domain in  $D$  over  $F$ ,  $G(D)$  generates what is called a *module* over the ring  $F[D]$ . A module has the same postulates as a vector space except that its scalars are elements of a ring rather than a field.

To characterize the algebraic properties of different generating matrices  $G(D)$  over integral domain  $F[D]$  we follow the lead of Forney in Ref. 3. Forney bases his study of CC on the well-known invariant-factor theorem of matrices over an integral domain. The statement of this theorem is reproduced for ring  $F[D]$  as follows:

*Invariant-Factor Theorem:* If  $F[D]$  is the integral domain and  $G(D)$  is a  $k \times n$  matrix over  $F[D]$ , then  $G$  has the invariant-factor decomposition or Smith normal form<sup>1</sup>,

$$G(D) = A(D)\Gamma(D)B(D) \quad (7)$$

where  $A(D)$  is a  $k \times k$  matrix over  $F[D]$  with an inverse  $A^{-1}(D)$  with elements in  $F[D]$ ;  $B(D)$  is a  $n \times n$  matrix over  $F[D]$  also with an inverse  $B^{-1}(D)$  with elements in  $F[D]$ ; and  $\Gamma(D)$  is a  $k \times n$  matrix over  $F[D]$  of form

$$\Gamma(D) = [\Gamma_1(D), 0] \quad (8)$$

<sup>1</sup>The invariant-factor theorem was developed by the British mathematician H. J. S. Smith in the middle of the 19th century.

with 0, a  $k \times (n - k)$  matrix of zeros and  $\Gamma_1(D)$  a diagonal matrix

$$\Gamma_1(D) = \text{diag} [\gamma_1(D), \gamma_2(D), \dots, \gamma_k(D)] \quad (9)$$

The diagonal elements  $\gamma_i(D)$  in (9) for  $1 \leq i \leq k$  are elements of  $F[D]$  and are called the invariant factors of  $G(D)$ . The invariant factors are unique and can be computed as follows: Let  $\Delta_0 = 1$ . Let  $\Delta_i$  be the greatest common divisor (GCD) of all  $i \times i$  subdeterminants (minors) of  $G$ . Then  $\gamma_i(D) = \Delta_i / \Delta_{i-1}$ . If  $\gamma_{i+1}(D) \neq 0$ , then  $\gamma_i(D)$  divides  $\gamma_{i+1}(D)$  for  $i = 1, 2, \dots, k - 1$ .

Forney in Ref. 3 sketches a proof of this theorem. Other more elementary and detailed expositions of this theorem can be found in certain classic works on modern algebra, e.g., see Ref. 5, Sec. 10, Ch. III. Rather than give a proof of this theorem it is perhaps better to illustrate its use with an example.

For an example let  $F = GF(2)$  and consider the generating matrix,

$$G(D) = G = \begin{bmatrix} 1 & 1+D & 1+D \\ 1+D & D & 0 \end{bmatrix} = A \begin{bmatrix} \gamma_1(D) & 0 & 0 \\ 0 & \gamma_2(D) & 0 \end{bmatrix} B \quad (10)$$

where  $A$  is a  $2 \times 2$  matrix and  $B$  is a  $3 \times 3$  matrix, both over  $F[D]$ . It is shown in Appendix A that the invariant-factor decomposition or Smith normal form of  $G(D)$  in (10) is

$$G(D) = A \Gamma B = \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1+D & L+D \\ 0 & 1+D+D^2 & 1+D^2 \\ 0 & 1+D & D \end{bmatrix} \quad (11)$$

Let the generating matrix of a CC have the invariant-factor decomposition in (7). Then the output of the encoder in (5) can be expressed as

$$y(D) = x(D) G(D) = x(D) A(D) \Gamma(D) B(D) \quad (12)$$

where  $A(D)$  and  $B(D)$  have inverses  $A^{-1}(D)$  and  $B^{-1}(D)$ , respectively, and  $\Gamma(D)$  is the  $k \times n$  matrix defined in (8).

For the encoding operation in (12) to be useful in the context of a communications system it is desirable that the mapping of  $k$ -vectors  $x(D)$  onto  $n$ -vectors  $y(D)$  over  $F[D]$  be one-to-one and reversible. For this to be true there must exist a right inverse matrix  $G^{-1}(D)$  of matrix  $G(D)$ . If  $G^{-1}(D)$  exists, then

$$y(D) G^{-1}(D) = x(D) G(D) G^{-1}(D) = x(D) I_k = x(D) \quad (13)$$

where  $I_k$  is the  $k \times k$  identity matrix and  $x(D)$  is uniquely recoverable from the encoded message  $y(D)$ . It will be assumed henceforth that the generating matrix  $G(D)$  has a right inverse and that this inverse is realizable in finite delay time.

By the invariant-factor theorem  $G(D) = A(D) \Gamma(D) B(D)$ . Thus for  $G^{-1}(D)$  to exist uniquely the last invariant factor of  $\Gamma(D)$  must not be zero, i.e.,  $\gamma_k(D) \neq 0$ . For otherwise, if  $\gamma_k(D) = 0$ , then  $G(D)$  would have rank less than  $k$  and as a consequence  $G^{-1}(D)$  would be nonunique.

If  $\gamma_k(D) \neq 0$ , then it can be verified that

$$G^{-1}(D) = B^{-1}(D) \Gamma^*(D) A^{-1}(D) \quad (14)$$

is the right inverse of  $G(D)$  where  $\Gamma^*(D)$  is an  $n \times k$  matrix with diagonal elements  $\gamma_i^{-1}(D)$  of form

$$\Gamma^*(D) = \begin{bmatrix} \Gamma_1^*(D) \\ 0 \end{bmatrix} \quad (15)$$

with  $\Gamma_1^*(D)$  the inverse of  $\Gamma_1(D)$  in (9); i.e.,

$$\Gamma_1^*(D) = \text{diag} [\gamma_1^{-1}(D), \gamma_2^{-1}(D), \dots, \gamma_k^{-1}(D)] \quad (16)$$

If  $\deg \gamma_k(D) > 0$ , then by (16), (15) and (14) the circuit to realize (13) would not be feedback-free.

Massey and Sain (Ref. 6) proved that an inverse  $G^{-1}(D)$  or some delayed version of it must be feedback-free in order to avoid CC that give rise to catastrophic error propagation. To avoid this problem it is desirable to make  $G^{-1}(D)$  feedback-free. If  $G^{-1}(D)$  is feedback-free,  $\deg \gamma_k(D) = 0$  and  $\gamma_1(D) = \gamma_2(D), \dots, \gamma_k(D) = 1$ . Hence  $\Gamma(D)$  must have the form

$$\Gamma(D) = [I_k, 0] \quad (17)$$

where  $I_k$  denotes a  $k \times k$  identity matrix and  $0$  denotes a  $k \times (n - k)$  matrix of zeros.

The above properties needed for a useful encoder are recapitulated in the following definition of a basic encoder given by Forney (Ref. 3).

*Definition:* A basic encoder  $G(D)$  is a CC with a feedback-free right inverse  $G^{-1}(D)$ . Both  $G(D)$  and  $G^{-1}(D)$  are polynomial matrices over  $F[D]$  such that  $G(D)G^{-1}(D) = I_k$ .

It was shown in detail by Forney (Ref. 3) and briefly above that an encoder is basic if and only if it is polynomial over  $F[D]$  and has all its invariant factors equal to one. Henceforth in this paper only basic encoders for a CC will be treated so that by (17) and the invariant-factor theorem the generating matrix for an  $(n, k)$  CC has the form

$$G(D) = A(D) [I_k, 0] B(D) \quad (18)$$

where  $I_k$  is the  $k \times k$  identity matrix.

Forney [Ref. 3, Appendix I] exhibits a parity-check matrix  $H(D)$  over  $F[D]$  for a generating matrix which is equivalent to  $G(D)$  in (18). It is shown here that this parity-check matrix is, in fact, the parity-check matrix of all generating matrices equivalent to (18) in the sense of Forney.

To treat the parity-check matrix the Euclidean ring  $F[D]$  is extended to the field  $F(D)$  of quotients or rational functions of polynomials in  $F[D]$ ; e.g., see Ref. 5, Sec. 3.8. In terms of sequences field  $F(D)$  is in one-to-one correspondence with the field  $S$  of all possible infinite sequences that can be generated by an impulse passed through all finite memory linear circuits with or without feedback.

Let  $g^{(1)}(D), g^{(2)}(D), \dots, g^{(k)}(D)$  be the  $k$  rows of matrix  $G(D)$  in (18). Since  $G(D)$  has rank  $k$ ,

$$V = \left\{ \sum_{i=1}^k \alpha_i(D) g^{(i)}(D) \mid \alpha_i(D) \in F(D) \right\}$$

is a  $k$ -dimensional vector space with respect to the field  $F(D)$  of scalars of its equivalent, to the field  $S$  of sequences. The null space  $V^\perp$  of  $V$  is a vector space with field  $F(D)$  of scalars of dimension  $n - k$  (see [Ref. 7, Sec. 3.2]).

Let  $H(D)$  be a matrix with coefficients in  $F[D] \subset F(D)$  of rank  $n - k$  which has its row space equal to  $V^\perp$ . Evidently any set of basis vectors in  $V^\perp$  can be used to find  $H(D)$  with components in  $F[D]$ . The rows of  $H(D)$ , found by this

means, constitute a basis for  $V^\perp$ . Thus  $V$  is a null space of  $V^\perp$  if and only if for any vector,  $y(D) \in V$ ,

$$y(D)H^T(D) = 0 \quad (19)$$

Since  $g^{(j)}(D) \in V$  for  $1 \leq j \leq k$  condition (19) implies,

$$G(D)H^T(D) = 0 \quad (20)$$

An explicit method due to Forney (Ref. 3, Appendix I) for constructing a parity matrix  $H(D)$  of all generating matrices  $G(D)$  with form (18) is now given. The coefficients of the parity-check matrix  $H(D)$ , developed by this technique, are polynomials in  $D$ , i.e., elements from the Euclidean ring  $F[D]$ .

Let the first  $k$  rows of matrix  $G$  in (18) be a submatrix  $B_1$  and the last  $(n - k)$  rows of  $B$  be a submatrix  $B_2$ . Then  $B$  is the matrix

$$B = [B_1, B_2]^T \quad (21)$$

in terms of submatrices  $B_1$  and  $B_2$ . Similarly denote the first  $k$  columns of the inverse matrix  $B^{-1}$  of  $B$  by  $B'_1$  and denote the last  $(n - k)$  columns of  $B^{-1}$  by  $B'_2$ . Then the inverse of  $B$  is the matrix

$$B^{-1} = [B'_1, B'_2] \quad (22)$$

Multiplying (21) and (22) yields

$$BB^{-1} = \begin{bmatrix} B_1 B'_1 & B_1 B'_2 \\ B_2 B'_1 & B_2 B'_2 \end{bmatrix} = \begin{bmatrix} I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} \quad (23)$$

and the matrix identities,

$$B_1 B'_1 = I_k, \quad B_1 B'_2 = 0; \quad (24)$$

$$B_2 B'_1 = 0, \quad B_2 B'_2 = I_{n-k}.$$

Let

$$H(D) = (B'_2)^T, \quad (25)$$

where  $T$  denotes transpose and  $B'_2$  is defined in (22), be a candidate for the parity-check matrix of  $G(D)$  in (18). Multi-

plying  $G(D)$  in (18) by the transpose of  $H(D)$  in (25) produces by (21) and (24),

$$\begin{aligned} G(D)H^T(D) &= A [I_k, 0] [B_1, B_2]^T B_2' \\ &= [A, 0] [0, I_{n-k}]^T = 0 \end{aligned} \quad (26)$$

Since the  $n - k$  rows of  $H(D)$  are the  $n - k$  columns of the invertible matrix  $B^{-1}(D)$ , these rows must be linearly independent. Thus  $H(D)$ , as defined in (25), has rank  $(n - k)$  and is a valid parity-check matrix for the general basic encoder  $G(D)$ , given in (18).  $H(D)$  is clearly not unique since a matrix of form  $H_1(D) = H(D)C(D)$ , where  $C(D)$  is a nonsingular matrix with elements in  $F[D]$ , is also a parity check matrix.

A parity-check matrix  $H(D)$ , associated with the general  $k \times n$  generator matrix  $G(D)$  of a basic encoder, is given by (25). However, for two special cases, the systematic  $(n, k)$  CC and the nonsystematic  $(n, 1)$  CC, a parity-check matrix can be found more directly. For example, for the systematic  $(n, k)$  CC the parity-check matrix has form  $G(D) = [I_k, P(D)]$  where  $I_k$  is the  $k \times k$  identity matrix and  $P(D)$  is a  $k \times (n - k)$  matrix of polynomials over  $F = GF(q)$ . In this case it is readily verified (Ref. 7) that  $H(D) = [-P^T(D), I_{n-k}]$  is a parity-check matrix.

The generator matrix for the nonsystematic  $(n, 1)$  CC has by (6) the form

$$G(D) = [g_1(D), g_2(D), \dots, g_n(D)] \quad (27)$$

where for simplicity the first subscript of  $g_{1j}(D)$  has been dropped. Condition (20) for the  $(1 \times n)$  generating matrix  $G(D)$  in (27) is easily shown to be satisfied by the  $(n - 1) \times n$  matrix

$$H(D) = [L(D), R(D)] \quad (28)$$

where  $L(D) = [g_2(D), g_3(D), \dots, g_n(D)]^T$  and  $R(D) = \text{diag} [g_1(D), g_2(D), \dots, g_n(D)]$ , so that it is a parity-check matrix of the  $(n, 1)$  CC generated by  $G(D)$  in (27).

To illustrate how to use (25) to compute a parity-check matrix consider again the example of a generating matrix, given in (10). By (A-2) in Appendix A the matrix  $B^{-1}$  for this  $G$  is

$$B^{-1} = \begin{bmatrix} 1 & 1+D & D+D^2 \\ 0 & D & 1+D^2 \\ 0 & 1+D & 1+D+D^2 \end{bmatrix} \quad (29)$$

so that by (22) and (25)

$$H(D) = (B_2')^T = [D+D^2, 1+D^2, 1+D+D^2], \quad (30)$$

the transpose of the last column of  $B^{-1}$  in (29). It is easily verified that (30) satisfies (20), i.e.,  $G(D)H^T(D) = 0$ , and that  $H(D)$  is of rank one. Hence  $H(D)$  in (30) is the parity-check matrix of the  $(3, 2)$  CC with generating matrix  $G(D)$  in (10).

In the next section the parity-check matrix will be used to obtain the syndrome of the received CC. A new decoding algorithm will then be presented by example.

### III. Solutions of the Syndrome Equation for Convolutional Codes

Let  $y(D)$  be transmitted CC in accordance with (5) where  $G(D)$  is the generating matrix for a basic encoder. Next let  $z(D) = y(D) + e(D)$  be the received code possibly corrupted by an error or noise sequence  $e(D)$ . The syndrome  $s(D)$  of  $z(D)$  is defined by

$$s(D) = z(D)H^T(D) \quad (31)$$

where  $H(D)$  is syndrome (28) for the basic encoder.

Substituting (5) in syndrome (31) yields

$$\begin{aligned} s(D) &= (y(D) + e(D))H^T(D) \\ &= e(D)H^T(D) + x(D)G(D)H^T(D) \end{aligned} \quad (32)$$

But by construction (20) is satisfied; i.e.,  $G(D)H^T(D) = 0$ . Hence the last term of (32) vanishes and

$$s(D) = e(D)H^T(D) \quad (33)$$

is the syndrome in terms of an error sequence or polynomial  $e(D)$ . The syndrome for a basic  $(n, k)$  CC is by (33) totally independent of the transmitted coded message  $y(D)$ . Syndromes for block group codes also have this property so that

one might suspect that it is possible to use syndromes to decode CC in a manner similar to that used for block codes.

The first step towards achieving this goal for CC is to find the general solution for  $e(D)$  of the syndrome equation (33), assuming that  $s(D)$  is computed by (31). That is, given  $s(D)$  by (31), solve for the set of all solutions  $e(D)$  of the syndrome equation (33).

To find the general solution of (33) again use is made of the important invariant-factor theorem of the last section. This theorem is applied to the transpose of the parity-check matrix,  $H^T(D)$  in (33). By construction the rank of  $H(D)$  is  $n - k$ , the maximum possible rank. By the invariant-factor theorem

$$H(D) = R^T(D) [\Lambda, 0] L^T(D) \quad (34)$$

where  $L(D)$  and  $R(D)$  are invertible  $n \times n$  and  $(n-k) \times (n-k)$  matrices over  $F(D)$ ,

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n-k}) \quad (35)$$

and "0" denotes a  $k \times (n-k)$  matrix of zeros. The  $\lambda_j$ 's in (35) are the invariant factors defined as follows: Let  $\delta_0 = 1$ , let  $\delta_j$  be the GCD of all  $j \times j$  minors of  $H(D)$ . Then  $\lambda_j = \delta_j / \delta_{j-1}$  and  $\lambda_j$  divides  $\lambda_{j+1}$  for  $j = 1, 2, \dots, n-k-1$ . Hence the transpose of  $H(D)$  in (34) has the Smith normal form

$$H^T(D) = L(D) [\Lambda, 0]^T R(D) \quad (36)$$

A lemma, due to Forney (Ref. 3, Appendix I), is used to evaluate the diagonal matrix  $\Lambda$  of invariant factors in (36). In the present terminology this lemma is stated as follows:

*Lemma (Forney):* The  $(n-k) \times (n-k)$  minors of  $H(D)$  are equal up to scalar field elements in  $F = GF(q)$  to the  $k \times k$  minors of  $G(D)$ .

Since the basic encoder has a generating matrix  $G(D)$  of form (18), the  $k \times k$  minors of  $G(D)$  have a GCD equal to 1. Hence by Forney's lemma the  $(n-k) \times (n-k)$  minors of  $H(D)$  also have a GCD equal to 1. Thus  $1 = \delta_{n-k} = \delta_{n-k-1} = \dots = \delta_1$  and the invariant factors of  $H^T(D)$  are all 1. Therefore,  $\Lambda = I_{n-k}$  and by (36)

$$H^T(D) = L(D) [I_{n-k}, 0]^T R(D) \quad (37)$$

is the Smith normal form of the transpose of a parity-check matrix for a basic encoder.

To solve for  $e(D)$  first substitute expression (37) for  $H^T(D)$  into (33). This yields

$$s(D) = e(D) L(D) [I_{n-k}, 0]^T R(D) \quad (38)$$

Next multiply both sides of (38) by  $R^{-1}(D)$  to obtain

$$\begin{aligned} \sigma(D) &= s(D) R^{-1}(D) = [e(D) L(D)] [I_{n-k}, 0]^T = \\ &e(D) [I_{n-k}, 0]^T \end{aligned} \quad (39)$$

where

$$\sigma(D) = s(D) R^{-1}(D) = [\sigma_1(D), \dots, \sigma_{n-k}(D)] \quad (40)$$

is an  $(n-k)$ -component transformed vector of  $S(D)$  and

$$\epsilon(D) = e(D) L(D) = [\epsilon_1(D), \dots, \epsilon_n(D)] \quad (41)$$

is an  $n$ -component transformed vector of the unknown polynomial error vector  $e(D)$ .

The component-by-component solution of (39) is obtained by equating components of the equation

$$\epsilon_j(D) = \sigma_j(D) \quad \text{for } 1 \leq j \leq n-k \quad (42a)$$

and

$$\epsilon_j(D) = t_{j-n+k}(D) \quad \text{for } n-k+1 \leq j \leq n \quad (42b)$$

where  $t_i(D)$  for  $1 \leq i \leq k$  is an arbitrary polynomial in the Euclidean ring  $F[D]$ .

Substituting (42a) and (42b) into the right side of (41) and solving for  $e(D)$  yields finally

$$\begin{aligned} e(D) &= [e_1(D), \dots, e_n(D)] \\ &= [\sigma_1(D), \dots, \sigma_{n-k}(D), t_1(D), \dots, t_k(D)] \\ &L^{-1}(D) \end{aligned} \quad (43)$$

as the general solution of the syndrome equation (33) in terms of the  $n-k$  components of the transformed syndrome  $\sigma(D)$  in (39) and  $k$  arbitrary polynomial parameters  $t_j(D)$  of  $F[D]$  for  $1 \leq j \leq k$ .

Some examples of the above technique for solving the linear Diophantine equations of the syndrome equation are now presented. Consider first the generating matrix

$$G(D) = [1 + D^2, 1 + D + D^2, 1 + D + D^2] \quad (44)$$

of a (3, 1) CC with constraint length 3. Using the Forney method developed in the last section a parity-check matrix for  $G(D)$  is readily calculated to be

$$H(D) = \begin{bmatrix} 1 + D + D^2, & 1 + D^2, & 0 \\ 1 + D + D^2, & D^2, & 1 \end{bmatrix} \quad (45)$$

The Smith normal form for  $H^T(D)$  is given by

$$H^T(D) = L \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T \quad (46)$$

where the inverse of  $L$  is

$$L^{-1} = \begin{bmatrix} D & 1 + D & D \\ 0 & 0 & 1 \\ 1 + D^2 & 1 + D + D^2 & 1 + D + D^2 \end{bmatrix} \quad (47)$$

Substituting (45) and (31) the syndrome is  $s(D) = z(D)H^T(D)$ . Hence by (33) and (46) the syndrome equation to solve is

$$\begin{aligned} s(D) &= [s_1(D), s_2(D)] = ([e_1(D), e_2(D), e_3(D)] L) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T \\ &= [\epsilon_1(D), \epsilon_2(D), \epsilon_3(D)] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T \end{aligned} \quad (48)$$

By (42a, b) the solution of (48) for  $\epsilon_j(D)$  is

$$\epsilon_1(D) = s_1(D), \epsilon_2(D) = s_2(D), \epsilon_3(D) = t(D) \quad (49)$$

where  $t(D)$  is an arbitrary element of  $F[D]$ . Finally the  $e_j(D)$  in terms of the solutions (49) is, using (47),  $[e_1(D), e_2(D), e_3(D)] = [s_1(D), s_2(D), t(D)] L^{-1}$ . This yields

$$e_1(D) = Ds_1(D) + (1 + D^2)t(D)$$

$$e_2(D) = (1 + D)s_1(D) + (1 + D + D^2)t(D) \quad (50)$$

$$e_3(D) = DS_1(D) + s_2(D) + (1 + D + D^2)t(D)$$

as the general solution of the syndrome equation (31) for the (3, 1) encoder in (44).

Another example of the general method for finding the general solutions of the syndrome equation is given in Appendix B for the (3, 2) CC with generating matrix (10). It is shown in the next section by an example how to use the solutions of the syndrome equation to perform optimum syndrome decoding.

#### IV. Syndrome Decoding of $(n, k)$ CC

Syndrome decoding of an  $(n, k)$  CC involves finding a maximum likelihood estimate (MLE)  $\hat{e}(D)$  of the actual error sequence in the coset, determined by (43), of all possible solutions of the syndrome equation (31). In order to accomplish this, both the weight or distance between codewords of a sequence and the type of channel need to be defined. For an  $(n, k)$  CC a possible error sequence is of form  $e(D) = [e_1(D), e_2(D), \dots, e_n(D)]$  where  $e_j(D)$  for  $1 \leq j \leq n$  are finite degree polynomials over  $GF(q)$ . The usual weight for a discretized channel is the Hamming weight. The Hamming weight of  $e(D)$  is

$$W_H[e(D)] = \sum_{j=1}^n W_H[e_j(D)] \quad (51)$$

where  $W_H[e_j(D)]$ , the Hamming weight of  $e_j(D)$ , is the number of nonzero coefficients of  $e_j(D)$ . It is convenient for this weight to assume that the channel can be approximated by a  $q$ -ary channel (see Ref. 2, Sec. 7.2).

If in (3)  $\deg[x_j(D)] \leq L - 1$  for  $1 \leq j \leq k$ , codeword  $y(D) = (y_1(D), y_2(D), \dots, y_n(D))$  is said to be the  $L$ th truncation of an  $(n, k)$  CC (see Ref. 2, p. 203). In this case

$$\deg[y_i(D)] \leq M + L - 1 \quad \text{for } 1 \leq i \leq n \quad (52)$$

where  $M$  is the memory. Hence an  $L$  truncated  $(n, k)$  CC can be considered to be a block code where each word has length  $n(L + M)$ . Hence for a truncated  $(n, k)$  CC the MLE of an error vector is what it would be for a linear block code. For a truncated  $(n, k)$  CC transmitted over a  $q$ -ary symmetric channel the MLE of  $e(D)$  is any vector  $e(D)$  of form (43) such that

$$W_H[\hat{e}(D)] = \min_{t_1(D), \dots, t_k(D)} (W_H[[\sigma_1(D), \dots, \sigma_{n-k}(D), t_1(D), \dots, t_k(D)]L^{-1}(D)]) \quad (53)$$

The above procedure for finding the MLE  $\hat{e}(D)$  or the error vector, needed to correct a codeword, is equivalent to the usual technique for correcting block codes, e.g., see Ref. 7, Sec. 7.5. A recursive technique is developed now by example to perform the minimization required in (53). The iterative minimization procedure, needed to efficiently find  $\hat{e}(D)$ , is a Viterbi-like or dynamic programming type of algorithm.

As an example of the new syndrome decoding algorithm consider the (3, 1) CC with the generating matrix in (44). If (44) is substituted in (4), then

$$[y_1, y_2, y_3] = [x + D^2x, x + Dx + D^2x, x + Dx + D^2x] \quad (54)$$

is the output of the encoder. Assume the input sequence is

$$x = [0 \ 1 \ 0 \ 0 \ 1 \ 0].$$

Then by (54),

$$y_1 = [0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0],$$

$$y_2 = [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0],$$

and

$$y_3 = [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0]$$

are the three components of the transmitted sequence. Let the corresponding three received sequences be

$$\begin{aligned} z_1 &= [1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0], \\ z_2 &= [0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0]; \\ z_3 &= [0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0] \end{aligned} \quad (55)$$

Next substitute the parity-check matrix (45) of the (3, 1) CC in (31) to obtain

$$\begin{aligned} s_1 &= (1 + D + D^2)z_1 + (1 + D^2)z_2 \\ s_2 &= (1 + D + D^2)z_1 + D^2z_2 + z_3 \end{aligned} \quad (56)$$

as the syndrome of the code. A calculation of the syndromes in (56) in terms of the received sequence  $[z_1, z_2, z_3]$  in (55) yields the syndrome sequences

$$\begin{aligned} s_1 &= [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0], \\ s_2 &= [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0] \end{aligned} \quad (57)$$

for this example. Given the syndrome sequences (57) it is desired now to solve syndrome equation (33) for the vector sequence  $e(D)$ .

The explicit general solutions of the syndrome equation in (33) for the components of  $e(D)$  were found in (50) of the last section. These solutions are explicitly

$$\begin{aligned} e_1 &= Ds_1 + t + D^2t \\ e_2 &= s_1 + Ds_1 + t + Dt + D^2t \\ e_3 &= Ds_1 + s_2 + t + Dt + D^2t \end{aligned} \quad (58)$$

where  $t$  is an arbitrary polynomial in  $F[D]$  and where for simplicity in notation the functional dependence on  $D$  of such functions as  $s_1(D)$ ,  $t(D)$ , etc., is deleted. Note that physically the functions  $Ds_1$ ,  $D^2t$ , etc., in (57) can be interpreted as the function  $s_1(D)$ , delayed by one time unit, and the function  $t(D)$ , delayed by two time units, respectively.

The problem now is to find from (58) and the given syndrome sequences (57) the maximum likelihood estimate  $\hat{e}$  of  $e = [e_1, e_2, e_3]$ . As in the Viterbi decoding algorithm  $\hat{e}$  is found iteratively or sequentially with the aid of a trellis diagram (see Ref. 8). In the present case, the underlying trellis diagram is "universal" in the sense that it is identical for all  $(n, k)$  CC of fixed  $k$  and constraint length  $L$ .

For the present example the states of the trellis diagram are equivalent to the states of the shift register needed in (58) to store sequentially the delayed versions  $Dt(D)$  and  $D^2t(D)$  of the arbitrary function  $t(D)$  on  $F$ . The block diagram of a shift register to hold  $Dt$  and  $D^2t$ , the function  $t$ , delayed by one time unit and two time units, respectively, is shown in Fig. 1. The state table of the shift register in Fig. 2, when conceived to be a sequential circuit, is given in Fig. 3. Finally, in Fig. 3 the trellis diagram of the state table in Fig. 2 is presented. A solid-line transition in Fig. 3 corresponds to the input  $t(D) = 0$ ; a dashed-line transition corresponds to the input  $t(D) = 1$ .

The new Viterbi-like syndrome decoding algorithm is illustrated by example in Fig. 4. The digits of the syndrome

sequences  $s_1$  and  $s_2$  computed in (57) are placed immediately over the corresponding transition paths of the trellis. The vectors  $[e_1, e_2, e_3]$  are computed at each stage from Eqs. (58), using the syndrome data for  $s_1, Ds_1$  and  $s_2$  and data  $t, Dt$  and  $D^2t$ , depending on the particular path taken.

To illustrate the above procedure, suppose that the algorithm has reached stage 2 at state  $d=11$ . At stage 2 the required values of the syndrome sequence needed in (58) are

$$s_1 = 0, \quad Ds_1 = 1 \quad \text{and} \quad s_2 = 1. \quad (59)$$

Since the previous two values of  $t$  leading to state  $d$  at stage 2 are 1,

$$Dt = 1 \quad \text{and} \quad D^2t = 1. \quad (60)$$

If the  $t=0$  branch is taken in the trellis from stage 2 and state  $d$ , a substitution of Eqs. (59), (60) and  $t=1$  into (58) yields  $[e_1, e_2, e_3] = [0, 1, 0]$  as the values of  $e$  along that segment of the path which has attained stage 3 at state  $b$ .

After stage 2 in Fig. 3 there are always two possible transitions leading to a given node in the trellis. The transition chosen is the one of minimum weight. This is precisely the technique of dynamic programming to determine a minimum weight path. A similar method is used in the Viterbi-algorithm to find a transmitted codeword that is closest in Hamming weight to the received codeword.

The trellis diagram shown in Fig. 4 is completed by the above illustration and the dynamic programming rules for choosing the "survivor" segment of path. At state 9 the minimum weight path in the trellis diagram of Fig. 4 is clearly  $acdbcbaaaa$ . The branches of this path yield  $\hat{e}(D) = [\hat{e}_1(D), \hat{e}_2(D), \hat{e}_3(D)] = [1, D^2, D^3]$  as the estimate of error vector  $e(D)$ . Subtracting these estimates of the error from the components of  $z$  in (55) produces

$$\begin{aligned} \hat{y}_1 &= [0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0] \\ \hat{y}_2 &= [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] \\ \hat{y}_3 &= [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] \end{aligned} \quad (61)$$

The Smith normal form of the (3, 1) CC generating matrix  $G(D)$  in (44) is  $G(D) = [1 \ 0 \ 0]B$  where

$$B^{-1} = \begin{bmatrix} 1+D & 1+D+D^2 & 1+D+D^2 \\ D & 1+D^2 & D^2 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus the estimate  $\hat{x}(D)$  of message in terms of the estimate of the transmitted codeword  $\hat{y}(D)$  is  $\hat{y} = \hat{x}G = \hat{x}[1 \ 0 \ 0]B$ . Solving this relationship for  $\hat{x}$ ,

$$\begin{aligned} \hat{x} &= \hat{y}G^{-1} = \hat{y}B^{-1}[1,0,0]^T \\ &= [\hat{y}_1, \hat{y}_2, \hat{y}_3] \begin{bmatrix} 1+D & 1+D+D^2 & 1+D+D^2 \\ D & 1+D^2 & D^2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= (1+D)\hat{y}_1 + D\hat{y}_2 \end{aligned} \quad (62)$$

since by (14)  $G^{-1} = B^{-1}[1, 0, 0]^T$  is the right inverse of  $G$ . A substitution of the estimates of  $\hat{y}_1$  and  $\hat{y}_2$  in (61) into (62) yields finally  $\hat{x} = [0 \ 1 \ 0 \ 0 \ 1 \ 0]$  as the estimate of the original message.

In the above example the new syndrome decoding algorithm produces the original message. However, if the number of errors exceeds the free distance  $d_f$  within any interval less than some multiple of the constant length, there may exist two or more paths of the same minimum error weight. In such a case a decoding failure and an erasure should be declared.

## V. Conclusions

In this paper a new syndrome decoding algorithm for an  $(n, k)$  convolutional code with a basic encoder is found. To accomplish this, the method of Forney is extended and used to find the syndrome  $H(D)$  of any basic encoder with generating matrix  $G(D)$ . Next, a general method based on the Smith normal form for matrices over the Euclidean ring  $F[D]$  is used to solve the syndrome equation for all possible error vectors  $e(D)$ . This general solution of the syndrome equation for an  $(n, k)$  code is shown in (43) to have the explicit form,  $e(D) = [\sigma_1(D), \dots, \sigma_{n-k}(D), t_1(D), \dots, t_k(D)]L^{-1}(D)$  where  $t_i(D)$  for  $1 \leq i \leq k$  is an arbitrary polynomial in the  $F[D]$  and  $\sigma_j(D)$  for  $1 \leq j \leq n-k$  are computable linear functions of the  $(n-k)$  syndromes  $s_1(D), \dots, s_{n-k}(D)$ .

To complete the new syndrome decoding algorithm, a Viterbi-like algorithm is developed to find the minimum Hamming-weight error-vector  $\hat{e}(D)$  of all solutions of the syndrome equation.  $\hat{e}(D)$  is the maximum likelihood estimate (MLE) of  $e(D)$  within the coset of all solutions of the syndrome equation. The estimate  $\hat{e}(D)$  is subtracted from the received codeword  $z(D)$  to obtain the MLE  $\hat{y}(D)$  of the transmitted codeword. Finally, the method of Forney (Ref. 3) and Massey and Sain (Ref. 7) is extended and applied to the basic

encoder to find the inverse circuit needed to obtain the decoded message  $\hat{x}(D)$  from  $\hat{y}(D)$ .

The new syndrome decoder for the  $(n, k)$  CC developed herein is comparable in complexity to the Viterbi decoder. The control logic and the computations associated with the

trellis are somewhat simpler in the new decoder. An advantage of the new encoder is the ease with which codes of the same constraint length, but with different rates, can be switched. Detailed comparisons of the new syndrome decoder with the Viterbi decoder for both low and high rate codes are clearly topics for future study.

## Acknowledgement

The authors wish to thank Charles Wang of JPL for the suggestions he made during the preparation of this paper.

## References

1. Schalkwijk, J. P. M., and Vinck, A. J., "Syndrome Decoding of Binary Rate- $\frac{1}{2}$  Convolutional Codes," *IEEE Trans. Comm.*, COM-24, 1976, 977-985.
2. McEliece, R. J., *The Theory of Information and Coding*, Reading, Mass.: Addison-Wesley Publishing Co., 1974.
3. Forney, G. D., "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Inform.*, IT-16, 1970, 720-738.
4. Herstein, I. M., *Topics in Algebra*, Xerox College Publishing, Lexington, Mass., 1975.
5. Albert, A. A., *Modern Higher Algebra*, University of Chicago Press, Chicago, Ill., 1947.
6. Massey, J. L., and Sain, M. K., "Inverses of Linear Circuits," *IEEE Trans. Comput.*, C-17, 1968, 330-337.
7. Peterson, W. W., and Weldon, E. J., Jr., *Error Correcting Codes*, 2nd Ed., Cambridge, Mass., MIT Press, 1972.
8. Viterbi, A., and Omura, J., *Digital Communication and Coding*, New York: McGraw-Hill Book Co., 1978.

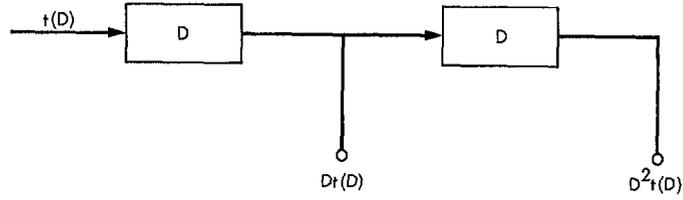


Fig. 1. Shift register to generate delayed versions of  $t(D)$

	$Dt$	$D^2t$	$t$	
			0	1
a =	0	0	0 0	1 0
b =	0	1	0 0	1 0
c =	1	0	0 1	1 1
d =	1	1	0 1	1 1

Fig. 2. State table of shift register for  $t(D)$

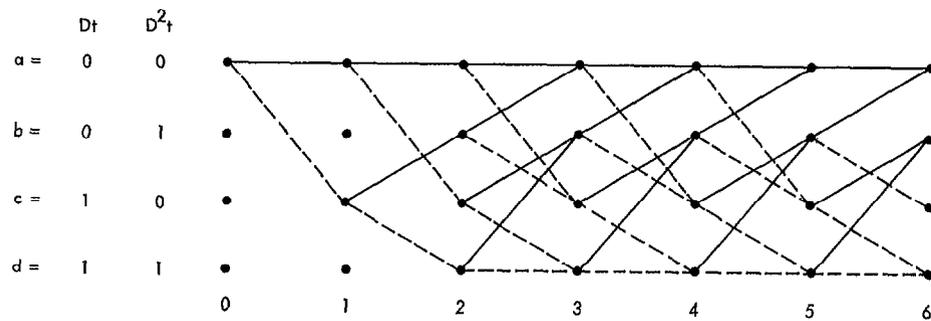
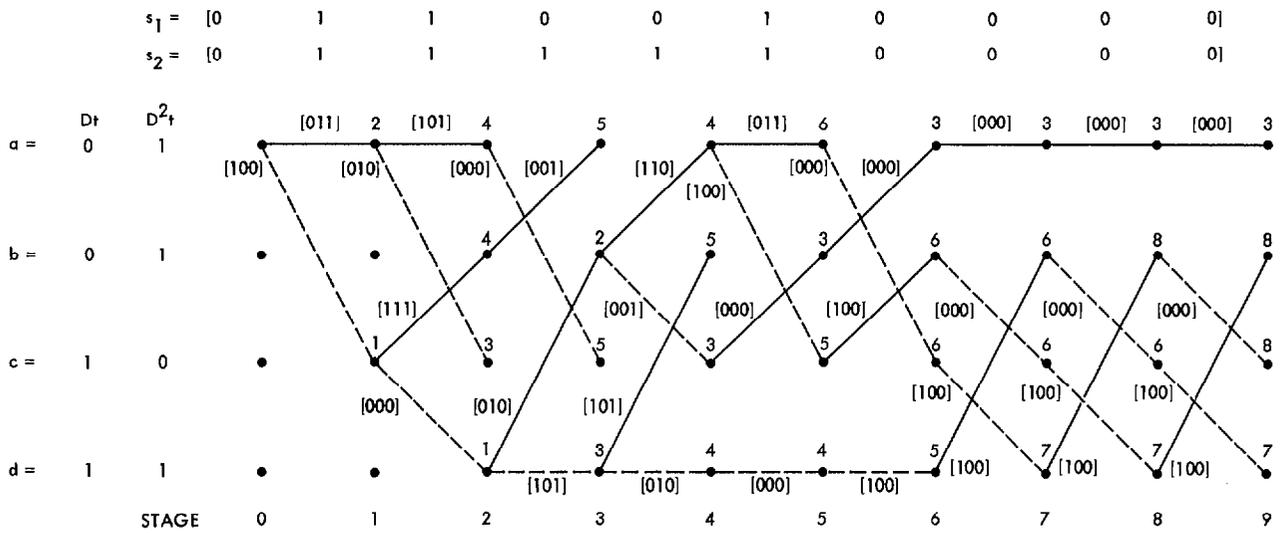


Fig. 3. Trellis diagram of shift register for  $t(D)$



$\hat{e} = [100 \ 000 \ 010 \ 001 \ 000 \ 000 \ 000 \ 000 \ 000]$   
 EACH BRANCH OF THE TRELLIS IS LABELED WITH  $[e_1, e_2, e_3]$  WHERE  $e_1 = Ds_1 + Dt$ ,  
 $e_2 = s_1 + Ds_1 + Dt + D^2t$ , AND  $e_3 = Ds_1 + s_2 + Dt + D^2t$ . EACH NODE AT STAGE  $k$  IS  
 LABELED WITH  $W_H[e_1^{(k)}(D), e_2^{(k)}(D), e_3^{(k)}(D)]$  WHERE  $e_i^{(k)}(D)$  IS POLYNOMIAL  $e_i(D)$   
 TRUNCATED AT DEGREE  $k$

**Fig. 4. A new Viterbi-like syndrome decoding algorithm for (3,1) CC**

## Appendix A

In this appendix an example of the invariant-factor decomposition for a generating matrix is given. It is shown that matrices  $A$  and  $B$  in (10) can be obtained by elementary row and column operations, respectively. The elementary operations are of three types (see, for example, Ref. 5):

- (1) The interchange of any two rows (columns) is called an elementary operation of type 1.
- (2) Let any row (column) be multiplied by an element of  $F[D]$ . The addition of this result to any other row (column) is called an elementary operation of type 2.
- (3) The multiplication of any row (column) by a nonzero scalar of  $F[D]$ , i.e., a nonzero element of  $F$  is called an elementary operation of type 3.

One procedure for finding  $A$  and  $B$  in (10) is to express (10) in the form,

$$\begin{bmatrix} 1 & 1+D & 1+D \\ 1+D & D & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} G \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A-1})$$

and reduce the left side to the form of  $\Gamma$  in (10) by elementary transformation. To put zeros in the second column of the first row multiply the first column by  $1+D$  and add the result to the second column of the matrix on the left. This same transformation is performed at the same time on the  $3 \times 3$  identity matrix on the right side of the equation. The result is

$$\begin{bmatrix} 1 & 0 & 1+D \\ 1+D & 1+D+D^2 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} G(D) \begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where on the right a new  $3 \times 3$  identity matrix multiplies the elementary transformation matrix

$$\begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

of type 2. Proceeding step by step in this fashion it is readily verified that the left side of (A-1) reduces finally to

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} G(D) \begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1+D \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & D \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} G(D) \begin{bmatrix} 1 & 1+D & D+D^2 \\ 0 & D & 1+D^2 \\ 0 & 1+D & 1+D+D^2 \end{bmatrix}$$

$$= A^{-1}(D) G(D) B^{-1}(D) \quad (\text{A-2})$$

For  $F = GF(2)$  it is easy to verify that an elementary matrix  $E$  over  $F[D]$  is its own inverse, i.e.,  $E^{-1} = E$  for an elementary matrix of types 1, 2 or 3. Thus solving for  $G$  in (A-2) yields

$$G(D) = \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & D \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1+D \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1+D & 1+D \\ 0 & 1+D+D^2 & 1+D^2 \\ 0 & 1+D & D \end{bmatrix}$$

$$= A(D) \Gamma(D) B(D) \quad (\text{A-3})$$

as the invariant-factor decomposition of  $G(D)$  in (A-1).

## Appendix B

In this appendix an example for finding the general solution of syndrome equation for the (3, 2) CC is given. Consider the parity-check matrix,  $H(D) = [D + D^2, 1 + D^2, 1 + D + D^2]$ , found in (30) for the (3, 2) CC with generating matrix (10). A diagonalization of  $H^T(D)$  yields for this example

$$L^{-1}(D) = \begin{bmatrix} 0 & 1+D & D \\ 1 & D+D^3 & D^2+D^3 \\ 0 & 1+D+D^2 & 1+D^2 \end{bmatrix} \quad (\text{B-1})$$

where  $L(D)$  is the left invertible matrix of the Smith normal form in (37) for  $H^T(D)$ . In this case the syndrome equation to solve for  $e(D)$  is

$$\begin{aligned} s(D) &= [e_1(D), e_2(D), e_3(D)] L [1, 0, 0]^T \\ &= [\epsilon_1(D), \epsilon_2(D), \epsilon_3(D)] [1, 0, 0]^T \end{aligned} \quad (\text{B-2})$$

The solution in (B-2) for  $e(D)$  is

$$\epsilon_1(D) = s(D), \quad \epsilon_2(D) = t_1(D), \quad \epsilon_3(D) = t_2(D) \quad (\text{B-3})$$

where  $t_1(D)$  and  $t_2(D)$  are arbitrary elements of parameters of  $F[D]$ . Solving  $e(D)L = \epsilon(D)$  for  $e(D)$  in terms of the solution (B-3) for  $\epsilon(D)$  is, using (B-1)

$$[e_1(D), e_2(D), e_3(D)] = [s(D), t_1(D), t_2(D)]$$

$$\times \begin{bmatrix} 0 & 1+D & D \\ 1 & D+D^3 & D^2+D^3 \\ 0 & 1+D+D^2 & 1+D^2 \end{bmatrix}$$

which yields, upon an equating of coefficients,

$$\begin{aligned} e_1(D) &= t_1(D) \\ e_2(D) &= (1+D)s(D) + (D+D^3)t_1(D) + (1+D+D^2)t_2(D) \\ e_3(D) &= Ds(D) + (D^2+D^3)t_1(D) + (1+D^2)t_2(D) \end{aligned} \quad (\text{B-4})$$

as the general solution of the syndrome equation (31) for the (3, 2) encoder in terms of two arbitrary parameters  $t_1(D)$  and  $t_2(D)$  in  $F[D]$ . It is a straightforward exercise to verify that (B-4) satisfies (31).