Sequential Syndrome Decoding of Convolutional Codes

I. S. Reed
University of Southern California

T. K. Truong
Communications Systems Research Section

This paper reviews previous studies (Refs. 1 and 2) of the algebraic structure of convolutional codes and extends those studies to apply to sequential syndrome decoding. These concepts are then used to realize by example actual sequential decoding, using the stack algorithm.

I. Introduction

In Ref. 1, a general technique was developed for finding all solutions of the syndrome equation of noncatastrophic (n, k) convolution codes (CCs). The solutions of the syndrome equation constitute the set of all possible error sequences that might have been made in transmission. In general, these solutions of the syndrome equation are either graphed on a error tree or on its more compact equivalent, an error trellis. In Ref. 2, the identity of the mathematical concept of Vinck, de Paepe, and Schalkwijk (VPS) (Ref. 3) is first generalized to all basic encoders and proved rigorously by the general technique developed in Ref. 1. This identity represents a canonical solution of the syndrome equation for all (n, k) convolutional codes with a basic encoder.

In this paper, the Fano metric for use in sequential decoding is modified so that it can be utilized to sequentially find the minimum weight error sequence in the set (or coset) of all solutions of the syndrome equation. To accomplish this, the stack algorithm is used primarily to expose the concepts of sequential syndrome decoding. Sequential syndrome decoding of CCs is illustrated in detail by an example. However, most of the present development applies as well to a Fano-like sequential syndrome decoding algorithm.

II. Fundamentals of Syndrome Decoding of Convolutional Codes

This section provides a brief review of the concepts of a CC needed for the systematic construction of an error tree or trellis of a sequential syndrome decoding. First, the input to an (n, k) CC can be represented as the D-transform

$$x(D) = \sum_{j=0}^{\infty} \mathbf{x}_j D^j \tag{1}$$

of the sequence $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots$, of k-vectors of form $\mathbf{x}_j = [x_{1j}, x_{2j}, \cdots, x_{kj}]$, where x_{ij} belong to a Galois field, F, here restricted to the binary field of two elements. Similarly, the ouptut of an (n, k) CC is the D-transform

$$y(D) = \sum_{j=0}^{\infty} \mathbf{y}_j D^j \tag{2}$$

where $\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \cdots$, constitutes a sequence of n-vectors $\mathbf{y}_j = [y_{1j}, y_{2j}, \cdots, y_{nj}]$, with $y_{ij} \in F$.

The input and output of a convolutional code are linearly related in terms of the operations of the symbol field, F. As a consequence, $y(D)$ in Eq. (2) is related, in general, to $x(D)$ in Eq. (1) by

$$y(D) = x(D)\mathbf{G(D)} \tag{3}$$

where $\mathbf{G(D)}$ is a $k \times n$ matrix of formal power series in D over the symbol field, F.

If the elements $g_{ij}(D)$ of $k \times n$ matrix

$$\mathbf{G(D)} = \begin{bmatrix} \mathbf{g}_{11}(\mathbf{D}), \cdots, \mathbf{g}_{1n}(\mathbf{D}) \\ \cdot \quad\quad \cdot \\ \cdot \quad\quad \cdot \\ \cdot \quad\quad \cdot \\ \mathbf{g}_{k1}(\mathbf{D}), \cdots, \mathbf{g}_{kn}(\mathbf{D}) \end{bmatrix} \tag{4}$$

are restricted to be polynomials, $\mathbf{G(D)}$ is the generating matrix of some (n, k) convolutional code. The maximum degree, n, of the polynomial elements of $\mathbf{G(D)}$ is called the memory delay of the code. Usually, the constraint length of the code is defined to be $K = n + 1$.

The polynomial elements of the generating matrix $\mathbf{G(D)}$ belong to the ring, $F[D]$, of all polynomials in D over the finite field, F.

$$\mathbf{G(D)} = \mathbf{A(D)}[\Gamma_k, 0]\,\mathbf{B(D)} \tag{5}$$

To avoid what is called "catastrophic error propagation," Massey and Sain (Ref. 4) proved that the right inverse $\mathbf{G^{-1}}$ of the generating matrix $\mathbf{G}$ must be feedback-free. Forney (Ref. 5) defined a *basic* encoder to be a CC that has a feedback-free inverse $\mathbf{G^{-1}}$ of generating matrix $\mathbf{G}$. Also, Forney in Ref. 5 (see also Ref. 1) showed that only basic encoders with Smith normal form

$$\mathbf{G(D)} = \mathbf{A}[I_k, 0]\,\mathbf{B} \tag{6}$$

will be considered. Where $\mathbf{A} = \mathbf{A(D)}$ is a $k \times k$ matrix with elements in $F[D]$, $\mathbf{B} = \mathbf{B(D)}$ is an $n \times n$ matrix with elements in $F[D]$, and I_k is a $k \times k$ diagonal matrix.

After transmission over a possible noisy channel, let

$$z(D) = y(D) + e(D) \tag{7}$$

be the D-transform of the received coded message, where $e(D)$ is the D-transform of the error sequence. A parity-check matrix, $\mathbf{H} = \mathbf{H(D)}$, of the generation matrix, $\mathbf{G(D)}$, is any full-rank $(n - k) \times n$ matrix with elements in $F[D]$ such that

$$\mathbf{G(D)H}^T\mathbf{(D)} = 0 \tag{8}$$

To find the parity-check matrix, $\mathbf{H}$, associated with $\mathbf{G}$, the method of Forney (Ref. 5) is used. To accomplish this, partition matrix $\mathbf{B}$ of the Smith normal form in Eq. (6) and its inverse, $\mathbf{B^{-1}}$, in the following manner. Let

$$\mathbf{G} = [\mathbf{B}_1, \mathbf{B}_2]^T \tag{9}$$

where T denotes matrix transpose; $\mathbf{B}_1$ and $\mathbf{B}_2$ are the first k rows and the last $(n - k)$ rows of $\mathbf{B}$, respectively.

Similarly, let

$$\mathbf{B^{-1}} = [\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2] \tag{10}$$

where $\bar{\mathbf{B}}_1$ and $\bar{\mathbf{B}}_2$ are the first k columns and the last $(n - k)$ columns of $\mathbf{B^{-1}}$, respectively. Then, since $\mathbf{BB^{-1}} = \mathbf{I}_n$, the following identities hold:

$$\begin{aligned} \mathbf{B}_1\bar{\mathbf{B}}_1 &= \mathbf{I}_k, \quad & \mathbf{B}_1\bar{\mathbf{B}}_2 &= 0 \\ \mathbf{B}_2\bar{\mathbf{B}}_1 &= 0, \quad & \mathbf{B}_2\bar{\mathbf{B}}_2 &= \mathbf{I}_{n-k} \end{aligned} \tag{11}$$

In terms of the above matrix partitions, the Forney parity-check matrix is defined by

$$\mathbf{H} = \bar{\mathbf{B}}_2^T \tag{12}$$

where "T" denotes matrix transpose. That $\mathbf{H}$, defined by Eq. (12), is a parity-check matrix satisfying Eq. (8) and is verified by substituting for $\mathbf{G}$ its Smith normal form and by partitioning $\mathbf{B}$ as given in Eq. (9) (see Ref. 1 for more details).

The syndrome $s(D)$ of the received sequence, corresponding to $z(D)$ in Eq. (7), is computed by

$$s = s(D) = z(D)\mathbf{H}^T\mathbf{(D)} \tag{13}$$

where $\mathbf{H(D)}$ is the parity-check matrix in Eq. (12). A substitution of $z(D)$, which appears in Eq. (7), into Eq. (13) yields

$$s(D) = e(D)\mathbf{H}^T\mathbf{(D)} \tag{14}$$

as the syndrome, computed by Eq. (13), but now in terms of only $e(D)$, the error sequence. This result shows that the

syndrome $s(D)$ is completely independent of the transmitted code sequence, $y(D)$.

Syndrome decoding of convolutional codes depends first, as it does of block codes, on solving the syndrome equation, Eq. (14), for all possible error sequence $e(D) \in F[D]$ that might have given rise to the syndrome $s(D)$ as computed by Eq. (17). Next, that error sequence, $\hat{e}(D)$, is chosen from that set of solutions of Eq. (14) that maximizes the likelihood of being close to the actual error sequence.

The general solution of the syndrome equation, Eq. (14), is given in Eq. (23) of Ref. 2. That is,

$$e = t\mathbf{G} + s\mathbf{B}_2 \tag{15}$$

where $\mathbf{B} = (\mathbf{H}^{-1})^T$, $\mathbf{G}$ is the $k \times n$ generating matrix, $\mathbf{B}_2$ is computed by Eq. (9), s is the $(n-k)$ component computed by Eq. (13), and t is an arbitrary k vector with elements in $F[D]$.

In the next section, the solution $e(D)$ of the syndrome equation, Eq. (14), is shown to graph on a rooted tree or trellis as a function of all possible binary sequences corresponding to the arbitrary D-transform $t(D)$. Such an error tree or trellis is used then to illustrate the sequential syndrome decoding algorithm for CCs using a modified Fano metric, μ_F, developed in Appendix A.

Now let $\hat{e} = \hat{e}(D)$ be the D-transform of the error sequence found by maximizing μ_F over all subsequences of the error trellis generated by the particular sequential syndrome decoding algorithm developed in Section III. Also, let $\hat{t} = \hat{t}(D)$ be the D-transform of the binary sequence path in the error trellis along which sequence $\hat{e}$ was found. By Eq. (23), $\hat{t}$ and $\hat{e}$ are related, in fact, by

$$\hat{e} = \hat{t}\mathbf{G} + s\mathbf{B}_2 \tag{16}$$

Next, note, by the Smith normal form in Eq. (6) of a generating matrix $\mathbf{G}$, that

$$\mathbf{G}^{-1} = \mathbf{B}^{-1}\begin{bmatrix}\mathbf{I}_k\\ 0\end{bmatrix}\mathbf{A}^{-1} \tag{17}$$

is the *right* inverse of $\mathbf{G}$. Hence, multiplying both sides of Eq. (16) yields, by Eq. (11), the identity,

$$\begin{aligned}\hat{e}\mathbf{G}^{-1} &= \hat{t} + s \cdot \mathbf{B}_2 \cdot \mathbf{G}^{-1}\\ &= \hat{t} + s\mathbf{B}_2[\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2]\begin{bmatrix}\mathbf{I}_k\\ 0\end{bmatrix}\mathbf{A}^{-1}\\ &= \hat{t} + s[0, \mathbf{I}_{n-k}]\begin{matrix}\mathbf{I}_k\\ 0\end{matrix}\mathbf{A}^{-1} = \hat{t}\end{aligned} \tag{18}$$

between $\hat{e}$ and $\hat{t}$.

The sequential syndrome decoding algorithm produces the D-transform $\hat{e}$ of the most likely estimate of the actual error sequence. Hence, by Eq. (7), a subtraction of $\hat{e}$ from z yields an estimate, $\hat{y}$, of the actual transmitted sequence y. But, $\hat{y}$, in turn, is generated by an estimated message sequence $\hat{x}$. That is,

$$\hat{y} = \hat{x}\mathbf{G} = z + \hat{e} \tag{19}$$

Multiplying both sides of the latter equality by $\mathbf{G}^{-1}$ produces, by Eq. (18),

$$\hat{x} = z\mathbf{G}^{-1} + \hat{e}\mathbf{G}^{-1} = z\mathbf{G}^{-1} + \hat{t} \tag{20}$$

This identity shows that $\hat{t}$, computed by the sequential syndrome algorithm, is a *correction factor* to the standard technique for recovering the message from its coded form in the noiseless case.

III. Sequential Syndrome Decoding with a Stack Algorithm

In this section, the details of a sequential syndrome decoding algorithm are developed by an example of the basic encoder. This encoder is the same as the (3, 1) CC described (Ref. 6, Chapter 12) in the development of the standard stack algorithm for the sequential decoding of convolutional codes. For expository and comparative purposes, some of the parameters in the example (Ref. 6, Chapter 12) are used also in this paper.

To illustrate a sequential syndrome decoding with a stack algorithm, consider the generating matrix

$$\mathbf{G(D)} = [1 + \mathbf{D}, 1 + \mathbf{D}^2, 1 + \mathbf{D} + \mathbf{D}^2] \tag{21}$$

of a (3, 1) CC. Using elementary column transformations, the Smith normal form of $\mathbf{G}$ in Eq. (21) is

$$\mathbf{G} = [1\ 0\ 0]\,\mathbf{B(D)} \tag{22a}$$

where

$$\mathbf{B(D)} = \begin{bmatrix}1 + \mathbf{D}, & 1 + \mathbf{D}^2, & 1 + \mathbf{D} + \mathbf{D}^2\\ 0 & 1 & 1\\ 1 & 0 & 1\end{bmatrix} \tag{22b}$$

With the same elementary transformations needed to obtain Eq. (22a) applied in reverse order, the inverse of $\mathbf{B}(\mathbf{D})$ is obtained as

$$\mathbf{B}^{-1}(\mathbf{D}) = \begin{bmatrix} 1, & 1+\mathbf{D}^2, & \mathbf{D} \\ 1, & \mathbf{D}^2, & 1+\mathbf{D} \\ 1, & 1+\mathbf{D}^2, & 1+\mathbf{D} \end{bmatrix} \tag{23}$$

Also from Eqs. (22b) and (9), one observes

$$\mathbf{B}_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Hence, the solution of the syndrome equation, Eq. (14), for this example is, by Eq. (15),

$$e = [e_1, e_2, e_3] = t[1+\mathbf{D}, 1+\mathbf{D}^2, 1+\mathbf{D}+\mathbf{D}^2] + [s_1, s_2]\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{24}$$

where $t \in \mathbf{F}[\mathbf{D}]$ and the syndrome $s = [s_1, s_2]$ is computed from Eqs. (13), (12), and (23) by

$$s = [s_1, s_2] = zH^T = [z_1, z_2, z_3]\begin{bmatrix} 1+\mathbf{D}^2, & \mathbf{D}^2, & 1+\mathbf{D}^2 \\ \mathbf{D}, & 1+\mathbf{D}, & 1+\mathbf{D} \end{bmatrix}^T \tag{25}$$

Thus, by Eqs. (24) and (25), the two components of the syndrome are

$$\begin{aligned} s_1 &= (1+\mathbf{D}^2)z_1 + \mathbf{D}^2 z_2 + (1+\mathbf{D}^2)z_3 \\ s_2 &= \mathbf{D}z_1 + (1+\mathbf{D})z_2 + (1+\mathbf{D})z_3 \end{aligned} \tag{26}$$

and the three components of the solution of the syndrome equation are

$$\begin{aligned} e_1 &= (1+\mathbf{D})t + s_2 \\ e_2 &= (1+\mathbf{D}^2)t + s_1 \\ e_3 &= (1+\mathbf{D}+\mathbf{D}^2)t + s_1 + s_2 \end{aligned} \tag{27}$$

For this example, let the input sequence be

$$x = [1\ 1\ 0\ 0\ 1] \tag{28a}$$

corresponding to its D-transform.

$$x = 1 + \mathbf{D} + \mathbf{D}^4 \tag{28b}$$

By Eqs. (3) and (21), the transmitted codeword is

$$y = [y_1, y_2, y_3] = [(1+\mathbf{D})x, (1+\mathbf{D}^2)x, (1+\mathbf{D}+\mathbf{D}^2)x] \tag{29}$$

The encoded transmission is obtained by substituting Eq. (28b) into Eq. (29) with the following calculations:

x:	1	1	0	0	1		x:	1	1	0	0	1	
$\mathbf{D}x$:		1	1	0	0	1	$\mathbf{D}^2x$:			1	1	0	0	1
y_1:	1	0	1	1	1	1	y_2:	1	1	1	1	1	0	1

and

x:	1	1	0	0	1	
$\mathbf{D}x$:		1	1	0	0	1
$\mathbf{D}^2x$:			1	1	0	0	1
y_3:	1	0	0	1	1	1	1

Hence,

$$\begin{aligned} y = {} & [1\ 1\ 1] + [0\ 1\ 0]\mathbf{D} + [1\ 1\ 0]\mathbf{D}^2 \\ & + [0\ 1\ 1]\mathbf{D}^3 + [1\ 1\ 1]\mathbf{D}^4 + [1\ 0\ 1]\mathbf{D}^5 \\ & + [0\ 1\ 1]\mathbf{D}^6 \end{aligned} \tag{30a}$$

is the D-transform of the encoded sequence

$$y = [1\ 1\ 1, 0\ 1\ 0, 1\ 1\ 0, 0\ 1\ 1, 1\ 1\ 1, 1\ 0\ 1, 0\ 1\ 1] \tag{30b}$$

Note that there is a one-to-one correspondence between sequences and their D-transforms. The final encoded sequence is obtained by multiplexing sequences y_1, y_2, and y_3. This example of encoding illustrates the point that a D-transform x operated on by $\mathbf{D}^k$, i.e., $\mathbf{D}^k x$, can be used interchangeably with sequence x, shifted right k times.

The circuit diagram for the (3, 1) CC encoder in Eq. (29) is shown in Fig. 1. This circuit is a linear finite-state machine, where the states of the machine are the four binary states of register (R_1, R_2). The state diagram of this machine is given in Fig. 2, where a dashed line corresponds to the input $x = 1$ and a bold line corresponds to the input $x = 0$. Finally, Fig. 3 is a graph of all possible inputs, next-state nodes, and outputs of

this state diagram as a function of discrete time. This is the trellis diagram of the (3, 1) CC encoder where the initial state is (0 0).

Let the received D-transform of code be after transmission of y in Eq. (30a),

$$z = [z_1, z_2, z_3] = [1\ 1\ 0] + [1\ 1\ 0]\mathbf{D} + [1\ 1\ 0]\mathbf{D}^2 + [1\ 1\ 1]\mathbf{D}^3 + [0\ 1\ 1]\mathbf{D}^4 + [1\ 0\ 1]\mathbf{D}^5 + [0\ 0\ 1]\mathbf{D}^6 \quad (31a)$$

so that

$$z_1 = [1\ 1\ 1\ 1\ 0\ 1\ 0], \qquad z_2 = [1\ 1\ 1\ 1\ 1\ 0\ 0] \quad (31b)$$

and

$$z_3 = [0\ 0\ 0\ 1\ 1\ 1\ 1] \quad (31b)$$

The syndrome $s = [s_1, s_2]$ for this received convolutional code is given in Eq. (26). Using Eq. (31b) in Eq. (26) yields

$$\begin{array}{rl}
z_1: & 1\ 1\ 1\ 1\ 0\ 1\ 0 \\
\mathbf{D}^2 z_1: & \quad 1\ 1\ 1\ 1\ 0\ 1\ 0 \\
\mathbf{D}^2 z_2: & \quad 1\ 1\ 1\ 1\ 1\ 0\ 0 \\
z_3: & 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
\mathbf{D}^2 z_3: & \quad 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
\hline
s_1 = & [1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1] \\
\mathbf{D} z_1: & \quad 1\ 1\ 1\ 1\ 0\ 1\ 0 \\
z_2: & 1\ 1\ 1\ 1\ 1\ 0\ 0 \\
\mathbf{D} z_2: & \quad 1\ 1\ 1\ 1\ 1\ 0\ 0 \\
z_3: & 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
\mathbf{D} z_3: & \quad 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
\hline
s_2 = & [1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0]
\end{array}$$

Hence, the D-transform of syndrome is

$$s = [s_1, s_2] = [1\ 1] + [1\ 1]\mathbf{D} + [1\ 1]\mathbf{D}^2 + [1\ 1]\mathbf{D}^4 + [1\ 1]\mathbf{D}^5 + [1\ 1]\mathbf{D}^6 + [0\ 1]\mathbf{D}^7 + [1\ 0]\mathbf{D}^8 \quad (32a)$$

or, as a sequence, the syndrome is

$$s = [1\ 1, 1\ 1, 1\ 1, 0\ 0, 1\ 1, 1\ 1, 1\ 1, 0\ 1, 1\ 0] \quad (32b)$$

The syndrome $s(D)$ for this example is given above, in Eq. (32). Given $s(D)$, an error trellis or tree can now be constructed from the solution in Eq. (27) of the syndrome equation, Eq. (32). The error trellis for the solution, Eq. (27), of the syndrome equation is shown in Fig. 4, where $s = [s_1, s_2]$ is the syndrome sequence found in Eq. (32). By Eq. (27) and Fig. 4, the error trellis is constructed exactly like the trellis for the encoder in Fig. 3, except that input is $t(D)$ instead of message $x(D)$ and the output is the code $t\mathbf{G}$ *plus a correction factor*,

$$S\mathbf{B}_2 = [s_1, s_2, s_1 + s_2]$$

instead of the code only.

To illustrate how $[e_1, e_2, e_3]$ is computed at each branch of the trellis, suppose one is at node or state $b = 10$ at time 1 and let $t = 0$. The output at this branch of the code trellis in Fig. 3 is $[1\ 0\ 1]$. The syndrome at this branch is $[s_1, s_2] = [1\ 1]$, so that the correction factor is

$$[s_2, s_1, s_1 + s_2] = [1\ 1\ 0]$$

Hence, the output of the error trellis at this branch is

$$[e_1, e_2, e_3] = [1\ 0\ 1] + [1\ 1\ 0] = [0\ 1\ 1]$$

All other outputs of the error trellis are computed in precisely the same manner.

The error trellis in Fig. 4 will be used next to demonstrate sequential syndrome decoding using the stack algorithm. The metric used for this purpose is the modification of the Fano metric developed in Appendix A. Let $P(e_{ij}) = Prob[E_{ij} = e_{ij}]$ and $P(Z_{ij}) = Prob[Z_{ij} = z_{ij}]$ be, respectively, the probability that E_{ij}, the bit of the i-th coordinate of the error sequence e at time j, equals e_{ij} (0 or 1) and the probability that Z_{ij}, the bit of the i-th coordinate of the received sequence z at time j, equals z_{ij} (0 or 1). By Eq. (A-16), the Fano metric for some received sequence of an (n, k) CC of depth L is

$$\mu_F = \sum_{j=1}^{L}\sum_{i=1}^{n} \lambda(e_{ij}) \quad (33)$$

where

$$\lambda(e_{ij}) = \log_2 \frac{P(e_{ij})}{P(z_{ij})} - R \quad (34)$$

is the incremental change in the Fano metric at time stage i and coordinate j. The rate for this example in Eq. (34) is one-third. Also, let $P(e = 1) = 0.1$ and $P(e = 0) = 0.9$. Since

$$P(z_{ij}) = P(z_{ij}|\mu_{ij} = 0)P(\mu_{ij} = 0) + P(z_{ij}|\mu_{ij} = 1)P(\mu_{ij} = 0)$$
$$= \tfrac{1}{2}$$

for $z_{ij} = (0, 1)$, Eq. (34) becomes $\lambda(e_{ij}) = \log_2(2 \times 0.9) - 1/3 = 0.52$ if $e = 0$ and $\lambda(e_{ij}) = \log_2(2 \times 0.1) - 1/3 = 2.65$ if $e_{ij} = 1$. For simplicity, this increment of the Fano metric is often scaled so that its values are approximately integers. For the present case, if $\sigma(e_{ij}) = \lambda(e)/0.52$ then approximately

$$\sigma(e_{ij}) = 1 \quad \text{if} \quad e_{ij} = 0$$
$$= -5 \quad \text{if} \quad e_{ij} = 1 \qquad (35)$$

is the scaled version of the modification of Fano's metric for this example.

The stack algorithm for sequential syndrome decoding assumes that paths in the error trellis with their associated metrics are stored in a stack. The entries in the stack are ordered by the Fano metric. The path with the largest metric is placed on top and all other paths are listed in the order of decreasing metric. In detail, the stack algorithm for sequential syndrome decoding has the following steps:

(1) Load stack with starting node or state of error trellis or tree; the starting metric is zero.

(2) Compute metric of successors of path in the top of the stack.

(3) Delete path in the top of the stack.

(4) Insert the new paths in the stack, and resort paths in the stack in the order of decreasing metric.

(5) If the path at the top of stack ends at a terminating node, stop; otherwise, return to Step 2, above.

Next, the stack algorithm is illustrated below for the error trellis in Fig. 4.

In the following example of sequential syndrome decoding using the stack algorithm, each entry of a stack contains three times: a partial path in the error trellis, the path metric, and the Hamming weight of path. With this stack structure, the steps of the stack algorithm for the error trellis in Fig. 4 and the metric in Eq. (35) appear as follows:

Step 1

1 (-3) (1)
0 (-9) (2)

Step 2

1 1 (- 6) (2)
0 (- 9) (2)
1 0 (-12) (3)

Step 3

1 1 0 (- 3) (2)
0 (- 9) (2)
1 0 (-12) (3)
1 1 1 (-21) (5)

Step 4

1 1 0 1 (- 6) (3)
0 (- 9) (2)
1 0 (-12) (3)
1 1 0 0 (-12) (4)
1 1 1 (-21) (5)

Step 5

1 1 0 1 1 (- 9) (5)
0 (- 9) (2)
1 0 (-12) (3)
1 1 0 0 (-12) (4)
1 1 0 1 0 (-15) (6)
1 1 1 (-21) (5)

Step 6

1 1 0 1 1 0 (- 6) (4)
0 (- 9) (2)
1 0 (-12) (3)
1 1 0 0 (-12) (4)
1 1 0 1 0 (-15) (6)
1 1 1 (-21) (5)
1 1 0 1 1 1 (-24) (7)

Step 7

1 1 0 1 1 0 1 (- 9) (5)
0 (- 9) (2)
1 0 (-12) (3)
1 1 0 0 (-12) (4)
1 1 0 1 1 0 0 (-15) (6)
1 1 0 1 0 (-15) (6)
1 1 1 (-21) (5)
1 1 0 1 1 1 (-24) (7)

Step 8

1 1 0 1 1 0 1 0 (- 6) (5)
0 (- 9) (2)
1 0 (-12) (3)
1 1 0 0 (-12) (4)
1 1 0 1 1 0 0 (-15) (6)
1 1 0 1 0 (-15) (6)
1 1 1 (-21) (5)
1 1 0 1 1 0 1 1 (-24) (8)
1 1 0 1 1 1 (-24) (7)

Step 9

1 1 0 1 1 0 1 0 0 (- 3) (5)
0 (- 9) (2)
1 0 (-12) (3)
1 1 0 0 (-12) (4)
1 1 0 1 1 0 0 (-15) (6)
1 1 0 1 0 (-15) (6)
1 1 0 1 1 0 1 0 1 (-21) (8)
1 1 1 (-21) (5)
1 1 0 1 1 0 1 1 (-24) (8)
1 1 0 1 1 1 (-24) (7)

The above stack algorithm shows that the best estimate of path t is

$$\hat{t} = [1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0] \qquad (36)$$

The error sequence along this path is the estimate

$$\hat{e} = [0\ 0\ 1, 1\ 0\ 0, 0\ 0\ 0, 1\ 0\ 0,$$
$$1\ 0\ 0, 0\ 0\ 0, 0\ 1\ 0, 0\ 0\ 0, 0\ 0\ 0] \quad (37)$$

If $\hat{e}$ were added to sequence z in Eq. (31a), the result would be the original coded sequence y in Eq. (30). However, it is more efficient to recover an estimate $\hat{x}$ of the original message by using Eq. (20).

The right inverse of the generating matrix for the example in Eq. (21) is, by Eqs. (22), (23), and (17)

$$\mathbf{G}^{-1} = \begin{bmatrix} 1, & 1+\mathbf{D}^2, & \mathbf{D} \\ 1, & \mathbf{D}^2, & 1+\mathbf{D} \\ 1, & 1+\mathbf{D}^2, & 1+\mathbf{D} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (38)$$

Hence, the estimate $\hat{x}$ of the original message is, by Eq. (20), for the example

$$\hat{x} = [z_1, z_2, z_3] \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \hat{t}$$

$$= z_1 + z_2 + z_3 + t \quad (39)$$

where $\hat{t}$ is the message correction factor obtained from Eq. (36) and the sequential syndrome decoding algorithm is developed below.

A substitution of Eqs. (31) and (36) into Eq. (39) yields the computation of x as follows:

$$\begin{array}{rl} z_1\text{:} & 1\ 1\ 1\ 1\ 0\ 1\ 0 \\ z_2\text{:} & 1\ 1\ 1\ 1\ 1\ 0\ 0 \\ z_3\text{:} & 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ t\text{:} & 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0 \\ \hline x = & [1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0] \end{array}$$

Hence,

$$\hat{x} = 1 + \mathbf{D} + \mathbf{D}^4 \quad (40)$$

is the estimate of the original message x. Since this agrees with Eq. (28), the sequential syndrome decoding algorithm correctly decoded z as given in Eq. (31).

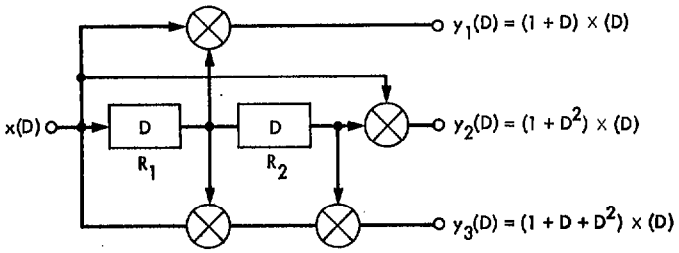

Fig. 1. Encoder circuit for (3, 1) convolutional code

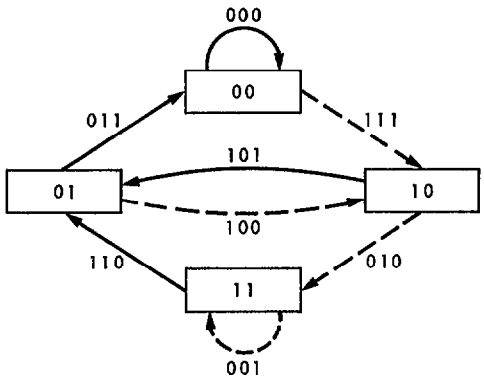

Fig. 2. State diagram for (3, 1) convolutional code (dashed lines correspond to $x = 1$; solid lines correspond to $x = 0$)

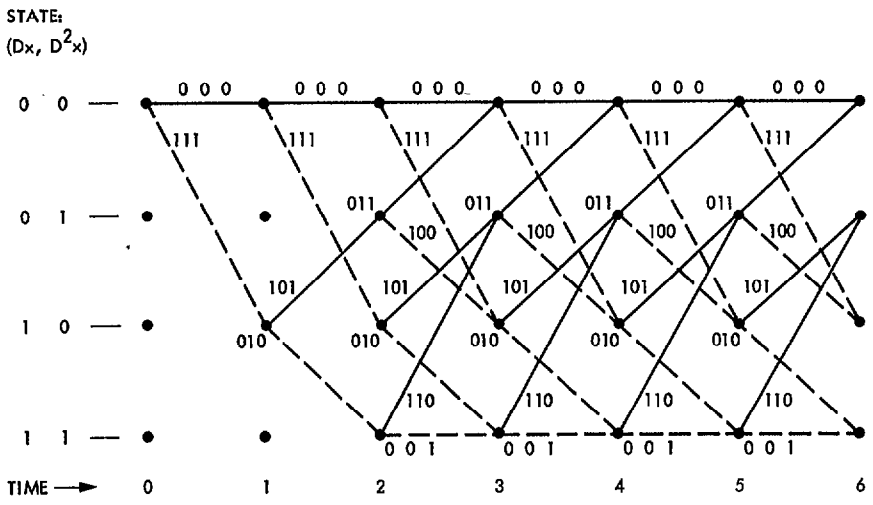

Fig. 3. Trellis diagram for encoder (3, 1) convolutional code (dashed lines correspond to $x = 1$; solid lines correspond to $x = 0$)

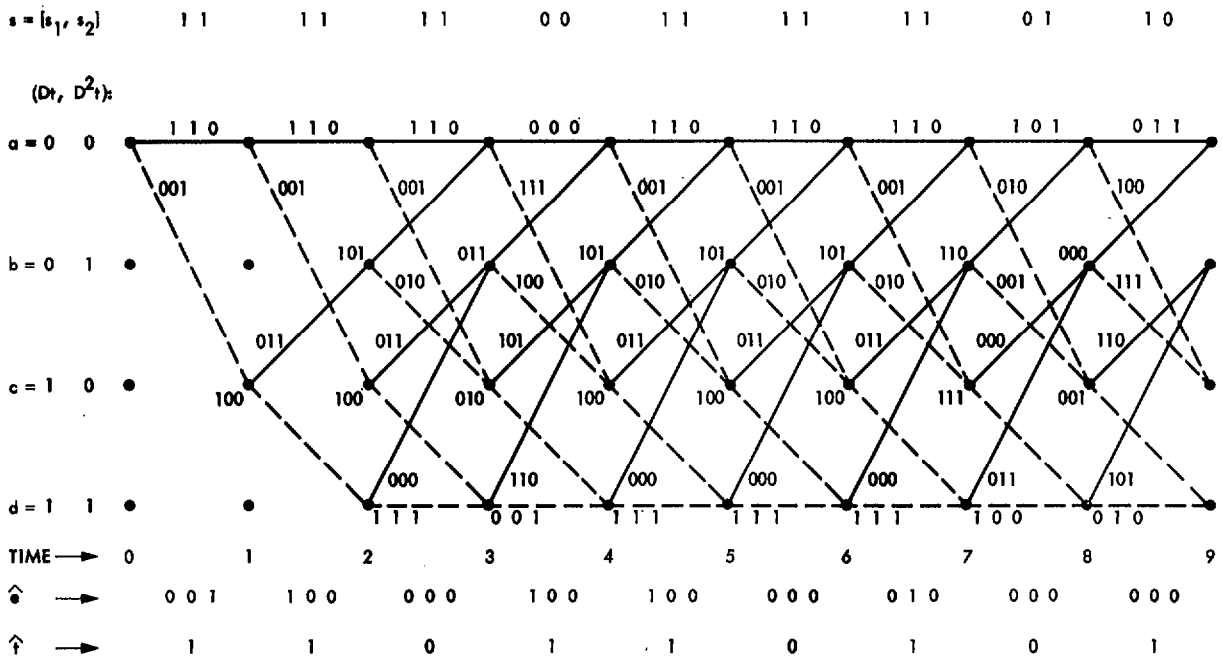

Fig. 4. Error trellis of $e = [e_1, e_2, e_3]$ as a function of t, state (Dt, D^2t), syndrome s, and stage k. Estimates of error sequence $\hat{e}$ and path $\hat{t}$ are shown below the trellis

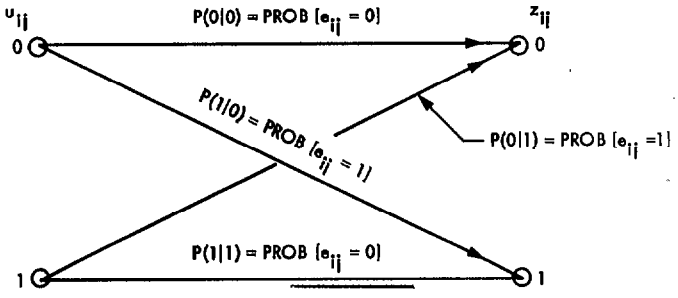

Fig. 5. Relationship of transition and error probabilities

Appendix A

The Modified Fano Metric for Sequential Syndrome Decoding of Binary Convolutional Codes

In Section II, the general technique is developed for finding all solutions of the syndrome equation of all noncatastrophic (n, k) CCs. In general, these solutions can be graphed on an error tree in precisely the same manner that these solutions were shown to graph on a trellis diagram. In fact, the trellis is just another way of representing the *graph* of a linear tree and vice versa.

The problem of a sequential syndrome decoding algorithm is to find, sequentially, at some desired tree depth L, the path or error sequence that has minimum Hamming weight. At any stage of such a sequential syndrome decoding algorithm, one must utilize a weight or metric to order the different partial paths the algorithm has already generated and considered in the error tree. In this appendix, the standard Fano metric is modified into a metric suitable for the syndrome decoding of convolutional codes.

To find the Fano metric, consider the code tree of a binary (n, k) CC of depth L. The transmitted code y_L is a code of L blocks of n binary bits of form

$$y_L = (y_{11}, y_{12}, \cdots, y_{1n}, y_{21}, y_{22}, \cdots, y_{2n}, \cdots, y_{L1}, y_{L2}, \cdots, y_{Ln})$$

where $x_{ij} = 0$ or 1 for $i = 1, 2, \cdots, L$ and $j = 1, 2, \cdots, n$. If such a code is sent through a binary symmetric channel, the received code z_L is of the same form as the transmitted code, namely,

$$z_L = (z_{11}, z_{12}, \cdots, z_{1n}, z_{21}, z_{22}, \cdots, z_{2n}, \cdots, z_{L1}, z_{L2}, \cdots, z_{Ln}) \tag{A-1}$$

An algorithm of a sequential decoder examines a set S_M of, say M, subsequences or subcodes of the code tree of depth L. This set of subcodes has the form

$$S_M = \left\{u_{k_1}, u_{k_2}, \cdots, u_{k_M}\right\}$$

where the subsequence u_m of the (n, k) CC code tree is of the form

$$u_m = \left(u_{11}, u_{12}, \cdots, u_{1n}, u_{21}, u_{22}, \cdots, u_{2n}, \cdots, u_{k_m 1}, u_{k_m 2}, \cdots, u_{k_m n}\right) \tag{A-2}$$

where $m = 1, 2, \cdots, M$.

The problem of the decoder is to find the sequence u_m in the set S_M that is most likely to compare with the received codeword z_L. To accomplish this comparison process with probability methods, one imagines, following Massey (Ref. 7), that each one of the partial codes u_m in S_M of depth k_m, for $1 \leqslant k_m \leqslant L$, is randomly extended to a depth L in the code tree. Such an extension is called a random tail. Explicitly, the extended codes for u_{ϱ_m} have the form

$$\hat{y}_L = (u_m, t_m) = (\hat{y}_{11}, \hat{y}_{12}, \cdots, \hat{y}_{1n}, \hat{y}_{21}, \hat{y}_{22}, \cdots, \hat{y}_{2n}, \cdots, \hat{y}_{L1}, \hat{y}_{L2}, \cdots, \hat{y}_{Ln})$$

where the "tail" is the binary vector,

$$t_m = \left(t_{11}, t_{12}, \cdots, t_{1n}, t_{21}, t_{22}, \cdots, t_{2n}, \cdots, t_{L-k_m,1}, t_{L-k_m,2}, \cdots, t_{L-k_m,n}\right) \tag{A-3}$$

chosen randomly from the code tree. Let t_m be chosen independently of u_m and y_L, in accordance with some arbitrary probability distribution $P(t_m)$.

The joint probability of transmitting u_m, augmenting u_m with the random tail t_m, and receiving y_L is

$$P(\hat{y}_L, z_L) = P(u_m, t_m, z_L) = P(u_m) P(t_m | u_m) P(z_L | u_m, t_m) = P(u_m) P(t_m) P(z_L | \hat{y}_L) \tag{A-4}$$

The last equality follows from the independence of t_m and u_m. Assume that the channel is a discrete memoryless channel (DMC) (Ref. 8). Then,

$$P(z_L | \hat{y}_L) = \prod_{j=1}^{n} \prod_{i=1}^{L} P(z_{ij} | \hat{y}_{ij})$$

$$= \prod_{j=1}^{n} \left[\prod_{i=1}^{k_m} P(z_{ij} | u_{ij}) \prod_{\ell=1}^{L-k_m} P\left(z_{\ell+k_m,j} \,|\, t_{\ell j}\right) \right] \tag{A-5}$$

But, the independence of y_L and t_m implies

$$P\left(z_{\ell+k_m,j} \,|\, t_{\ell j}\right) = P\left(z_{\ell+k_m,j}\right) \tag{A-6}$$

Hence, substituting Eqs. (B-5) and (B-7) into Eq. (4) yields

$$P(u_m, t_m, z_L) = P(u_m) P(t_m) \prod_{j=1}^{n} \times \left[\prod_{i=1}^{k_m} P(z_{ij} \,|\, u_{ij}) \prod_{\ell=1}^{L-k_m} P\left(z_{\ell+k_m,j}\right) \right] \tag{A-7}$$

The marginal joint distribution of u_m and y_L is obtained by summing $P(u_m, t_m, z_L)$ in Eq. (A-7) over all possible random tails t_m. Since

$$\sum_{t_m} P(t_m) = 1$$

this distribution is

$$P(u_m, z_L) = P(u_m) \prod_{j=1}^{n} \times \left[\prod_{i=1}^{k_m} P(z_{ij} \,|\, u_{ij}) \prod_{\ell=1}^{L-k_m} P\left(z_{\ell+k_m,j}\right) \right] \tag{A-8}$$

where $P(u_{ij})$ is the probability of bit y_{ij} at the output of the binary symmetric channel.

To compute the probability $P(u_m)$ that the partial code u_m in set S_M was transmitted, assume that the information symbols are independent and equally likely, i.e., the probability an information symbol is one equals $P = 1/2$. By Eq. (A-2), the number of information symbols used to generate the partial codeword u_m is

$$N_m = Rnk_m = \left(\frac{k}{n}\right) nk_m = k \cdot k_m \tag{A-9}$$

since the dimensionless code rate is $R = k/n$. Let i_m be the number of ones in the information sequence that generated the partial codeword u_m. Then,

$$P(u_m) = p^{i_m} q^{N_m - i_m} = p^{i_m} q^{Rnk_m - i_m} = 2^{-Rnk_m} \tag{A-10}$$

By Eqs. (A-8) and (A-10), and the rules for conditional probability,

$$P(u_m | z_L) = \frac{2^{-Rnk_m} \prod_{j=1}^{n} \left[\prod_{i=1}^{k_m} P(z_{ij} | u_{ij}) \prod_{\ell=1}^{L-k_m} P\left(z_{\ell+k_m,j}\right) \right]}{P(z_L)} \tag{A-11}$$

is the conditional probability that the partial codeword u_m was transmitted, given that y_L was received. The minimum error decoding (MED) principle (Ref. 8) says to choose that path u_m in set S_M that maximizes $P(u_m | z_L)$ in Eq. (A-11). Since $P(y_L)$ is constant over this maximization, this is equivalent to the choice m in set $\{1, 2, \cdots, M\}$, which maximizes

$$\psi(m) = 2^{-Rnk_m} \prod_{j=1}^{m} \left[\prod_{i=1}^{k_m} P(z_{ij} | u_{ij}) \prod_{\ell=1}^{L-k_m} P\left(z_{\ell+k_m,j}\right) \right] \tag{A-12}$$

Finally, since the term

$$\prod_{i=1}^{n} \prod_{j=1}^{L} P(z_{ij})$$

is a constant with respect to the maximization of $\psi(m)$, this maximization is equivalent to the maximization with respect to m of the ratio

$$\phi(m) = \psi(m) \Big/ \prod_{j=1}^{n} \prod_{i=1}^{L} P(z_{ij})$$

or its logarithm,

$$\mu_F(m) = \sum_{i=1}^{k_m} \sum_{j=1}^{n} \left[\log \frac{P(z_{ij} | u_{ij})}{P(z_{ij})} - R \right] \tag{A-13}$$

$\mu_F(m)$ in Eq. (A-13) is the standard Fano metric for the sequential decoding of an (n, k) CC.

The above derivation of the Fano metric is more directly applicable to (n, k) CC than the classical proof due to Massey (Ref. 7). It is also somewhat more precise than the recent derivation given by Clark and Cain (Ref. 9), upon which the present derivation is based.

As in Eq. (A-2), let u_{ij} denote a particular bit of some subsequence u_m in S_M of possible transmitted messages in the code tree of an (n, k) CC, and let z_{ij} denote the corresponding received bit. Then, z_{ij} is related to bit u_{ij} by

$$z_{ij} = u_{ij} + e_{ij}$$

where, solving for e_{ij},

$$e_{ij} = z_{ij} + u_{ij} \qquad \text{(A-14)}$$

is the difference, or error, between the possible transmitted bit u_{ij} and the received bit z_{ij}, and addition is modulo 2 or addition of the field $F_2 = \{0, 1\}$ of two elements.

But, by the Shannon transition diagram in Fig. 5, the conditional probabilities $P(z_{ij}|u_{ij})$ are in all cases given by

$$P(z_{ij}|u_{ij}) = \text{Prob}\,[e_{ij} = u_{ij} + z_{ij}] \equiv P(e_{ij}) \qquad \text{(A-15)}$$

A substitution of Eq. (A-15) into Eq. (A-13) yields

$$\mu_F(m) = \sum_{i=1}^{k_m} \sum_{j=1}^{n} \left[\log \frac{P(e_{ij})}{P(z_{ij})} - R \right] \qquad \text{(A-16)}$$

as the desired modification of the Fano metric.

References

1. Reed, I. S., T. K. Truong, "New Syndrome Decoding Techniques for the (n, k) Convolutional Codes," *Proceedings IEEE*, Vol. 131, Pt. F, No. 4, July 1984.
2. Reed, I. S., T. K. Truong, "Error Trellis Syndrome Decoding Techniques for Convolutional Codes," *TDA Progress Report 42-78*, Jet Propulsion Laboratory, Pasadena, California, April-June 1984.
3. Vinck, A. J., A. J. P. de Paepe, and J. P. M. Schalkwijk, "A Class of Binary Rate One-Half Convolutional Codes that Allows an Improved Stack Decoder," *IEEE Transactions on Information Theory*, IT-26, No. 4, 1980, pp. 389-392.
4. Massey, J. L., and M. K. Sain, "Inverses of Linear Circuits," *IEEE Transactions on Comput.*, C-17, 1968, pp. 330-337.
5. Forney, G. D., "Convolutional Codes: Algebraic Structures," *IEEE Transactions on Information Theory*, IT-6, 1970, pp. 720-738.
6. Lin, S., and D. J. Costello, Jr., *Error Control Coding*, Prentice-Hall, New Jersey, 1983.
7. Massey, J. L., "Variable-Length Codes and the Fano Metric," *IEEE Transactions on Information Theory*, IT-18, January 1972, pp. 190-198.
8. Gallager, R. G., *Information Theory and Reliable Communications*, John Wiley and Sons, Inc., New York, 1968.
9. Clark, G. C., Jr., and J. B. Cain, *Error-Correcting Codes for Digital Communications*, Plenum Press, New York, 1981.