Error-Trellis Syndrome Decoding Techniques for Convolutional Codes

I. S. Reed
University of Southern California

T. K. Truong
Communication Systems Research Section

In this paper, a new error-trellis syndrome decoding technique for convolutional codes is developed. This algorithm is specialized then to the entire class of systematic convolutional codes. Finally, this algorithm is applied to the high-rate Wyner-Ash convolutional codes. A special example of the one-error-correcting Wyner-Ash code, a rate 3/4 code, is treated in this paper. The error-trellis syndrome decoding method applied to this example shows in detail how much more efficient syndrome decoding is than, say, Viterbi decoding, if applied to the same problem. For standard Viterbi decoding, 64 states would be required, whereas in the example only 7 states are needed. Also, within the 7 states required for decoding, many fewer transitions are needed between the states.

I. Introduction

This paper outlines a simplification of previous syndrome decoding methods (Refs. 1, 2) for convolutional codes (CCs). The new method involves finding minimum error paths in what is called an error tree, or its more compact equivalent, an error trellis. As will be shown, the computation of the error trellis is accomplished by finding the solution of the syndrome equations explicitly in terms of the received coded sequence. The error trellis is a graph of all path solutions of the syndrome equations. This new procedure for finding the error trellis differs from previous methods in that it does not involve an explicit computation of the syndrome.

After the error trellis has been computed, the minimum weight path in the error trellis is found by any one of many minimization techniques, including the Viterbi and sequential minimum-path-finding techniques. The minimum error path that is found by such a minimization of the path weights in the error trellis is shown to be a best estimate of the correction factor needed to correct the "noisy" message. Such a noisy message is obtained by the Massey and Sain method (Ref. 3) of applying the right inverse of the generator matrix to the received coded message.

Development of the new error trellis syndrome decoding scheme is followed by a discussion of its application to high-rate systematic convolutional codes. This application to high-rate CCs shows the real advantage of syndrome decoding over Viterbi decoding of CCs in terms of reduced complexity.

II. Syndrome Decoding With the Error Trellis

This section provides a brief development of the concepts of a convolutional code (CC) needed for systematically constructing an error trellis for minimum-error-path decoding. Here, only a brief synopsis of these concepts is given, enough to systematically construct an error tree or trellis without resorting to the intermediate step of computing the syndrome.

The inputs and outputs of an (n, k) CC can be represented, respectively, as D-transforms,

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

and

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

of the input sequence of k-vectors of form $x_j = [x_{1j}, x_{2j}, \ldots, x_{kj}]$ and the output sequence of n-vectors of form $y_j = [y_{1j}, y_{2j}, \ldots, y_{nj}]$, where x_{ij} and y_{ij} belong to a finite Galois field $F = G(q)$ usually restricted to the binary field $GF(2)$ of two elements, and D is the delay operator. The input $x(D)$ and the output $y(D)$ are linearly related by means of a $k \times n$ generator matrix $G(D)$ as follows:

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

where the elements of $G(D)$ are assumed usually to be polynomials over the finite field $GF(q)$, where q is the power of a prime integer. The maximum degree M of the polynomial elements of $G(D)$ is called the memory delay of the code, and the constraint length of the code is $k = M + 1$.

In order to avoid catastrophic error propagation, the encoder matrix $G(D)$ is assumed to be *basic* (Ref. 3). This means that the Smith normal form of $G(D)$ is

$$G = A\,[I_k, 0]\,B \tag{4}$$

where $A = A(D)$ is a $k \times k$ invertible matrix with elements in $F[D]$, the ring of polynomials in D over F, and $B = B(D)$ is an $n \times n$ invertible matrix with elements in $F[D]$. The elements of the inverses A^{-1} and B^{-1} of matrices A and B, respectively, are polynomials in $F[D]$ (Ref. 4).

By definition, the parity check matrix associated with $G = G(D)$ is any full-rank $(n - k) \times n$ matrix with polynomial elements in $F[D]$ which satisfies

$$G(D)\,H^T(D) = 0 \tag{5}$$

where T denotes matrix transpose. A modification of the method of Forney (Ref. 4) is used to find H. The method involves a partitioning of matrix B in Eq. (4), as well as its inverse B^{-1}. That is, let

$$B = \begin{bmatrix} B_1 \\ \\ B_2 \end{bmatrix} \tag{6}$$

and

$$B^{-1} = [\bar{B}_1, \bar{B}_2] \tag{7}$$

where the first k *rows* of B constitute the submatrix B_1 and the remaining $(n - k)$ rows are the matrix B_2, and where, likewise, the first k *columns* of B^{-1} constitute the submatrix B_1 and the remaining $(n - k)$ columns are the matrix $\bar{B}_2$.

Since B times its inverse B^{-1} is the $n \times n$ identity matrix, the following identities evidently hold:

$$\left.\begin{aligned} B_1 \bar{B}_1 &= I_k \\ B_1 \bar{B}_2 &= 0 \\ B_2 \bar{B}_1 &= 0 \\ B_2 \bar{B}_2 &= I_{n-k} \end{aligned}\right\} \tag{8}$$

In terms of the partition in Eq. (7), the Forney parity-check matrix is defined by

$$H = \bar{B}_2^T \tag{9}$$

It is readily verified using Eq. (4) and the identities of Eq. (8) that Eq. (9), in fact, satisfies Eq. (5), the requirement for H to be a parity-check matrix. It should be noted that the parity-check matrix is not unique. For example, it can be shown that $H = C\,B_2^T$ is a parity-check matrix where C is any $(n - k) \times (n - k)$ invertible matrix with elements in $F[D]$.

For an input message $x(D)$, as defined in Eq. (1), the encoded message or code sequence is $y(D)$ as generated by Eq. (3). Suppose that $y = y(D)$ is transmitted and $z = z(D)$ is received. Then, the transmitted and received sequences are related by

$$z(D) = y(D) + e(D) \tag{10}$$

where $e(D)$ is the D-transform of the error sequence. The *syndrome* of the received code $z(D)$ is

$$s(D) = z(D) \times H^T(D) \tag{11}$$

If $y(D)$ in Eq. (3) is substituted in Eq. (10), then the syndrome, computed in Eq. (11), satisfies, by Eq. (5),

$$\begin{aligned} s &= z H^T \\ &= (x G + e) H^T \\ &= e H^T \end{aligned} \tag{12}$$

This is the syndrome equation for the error sequence $e = e(D)$. The syndrome equation, Eq. (12), shows that the syndrome computed in Eq. (11) is functionally independent of the original transmitted code $y(D)$ as well as the original message $x(D)$.

The problem of syndrome decoding of convolutional codes is, as for block codes, to solve the syndrome equation, Eq. (12), for the set of all possible solutions $e = e(D)$. It has been shown (Ref. 1) that this set of solutions is a coset of the set of all codewords.

To explicitly solve the syndrome equation, Eq. (12), substitute H as given by Eq. (9) in Eq. (12), thereby obtaining

$$s = e\bar{B}_2 = eB^{-1}\begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} \tag{13}$$

where I_{n-k} is the identify matrix of $(n - k)$ rows. In Eq. (13), let

$$\epsilon = eB^{-1} \tag{14}$$

so that Eq. (13) becomes the simple equation

$$s = \epsilon \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} \tag{15}$$

where $s = [s_1, s_2, \ldots, s_{n-k}]$ and $\epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]$. The *general* solution of Eq. (15) over the ring $F[D]$ is given evidently by

$$\begin{aligned} [\epsilon_1, \epsilon_2, \ldots, \epsilon_k] &= [\tau_1, \tau_2, \ldots, \tau_k] \equiv \tau \\ [\epsilon_{k+1}, \epsilon_{k+2}, \ldots, \epsilon_n] &= [s_1, s_2, \ldots, s_{n-k}] \\ &= s \end{aligned} \tag{16}$$

where $\tau_j = \tau_j(D)$ are *arbitrary* elements in $F[D]$. Thus, more compactly, the general solution of Eq. (14) is

$$\begin{aligned} \epsilon &= [\tau, s] \\ &= eB^{-1} \end{aligned} \tag{17}$$

where τ, as in Eq. (16), is an arbitrary k-vector with elements in the ring $F[D]$. Finally, a multiplication of both sides of Eq. (17) by B yields

$$\begin{aligned} e &= \epsilon B \\ &= [\tau, s]\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \\ &= \tau B_1 + sB_2 \end{aligned} \tag{18}$$

in terms of submatrices B_1 and B_2 in Eq. (6) as the most general solution of the syndrome equation, Eq. (12).

The general solution, Eq. (18), of the syndrome equation can be expressed in a number of different forms. For example, it can be put into canonical form originally found heuristically by Vinck, De Paepe, and Schalkwijk (Ref. 4). Towards this end, note from the identities in Eqs. (8) and (9) that B_2^T is the left inverse, denoted by H^{-1}, of the parity-check matrix H. Hence,

$$B_2 = (H^{-1})^T \tag{19}$$

Next, note from the Smith normal form in Eq. (4) of a basic encoder that

$$\begin{aligned} A^{-1}G &= [I_k, 0]\,B \\ &= B_1 \end{aligned} \tag{20}$$

A substitution of B_1 in Eq. (20) and B_2 in Eq. (19) into Eq. (18) yields

$$e = \tau A^{-1}G + s(H^{-1})^T \tag{21}$$

Since τ is an arbitrary k-vector of elements in $F[D]$,

$$t = \tau A^{-1} \tag{22}$$

is also an arbitrary vector of polynomials in $F[D]$. Finally, substituting t in Eq. (22) into Eq. (21) yields,

$$e = tG + s(H^{-1})^T \tag{23}$$

as the general solution of the syndrome equation, Eq. (12), where G is the $k \times n$ generator matrix, H^{-1} is the left inverse of the parity-check matrix, s is the $(n - k)$ component syndrome computed by Eq. (11), and t is an arbitrary k-vector with elements in $F[D]$. The above proof is a simplification of a more general version, given in Ref. 2, of the Vinck, de Paepe, and Schalkwijk identity heuristically established in Ref. 4. Herein, it is desired to put Eq. (23) in a form which makes it possible in the syndrome decoding process to bypass the explicit computation of the syndrome $s(D)$.

Towards this end, substitute Eq. (19) into Eq. (23) and, by Eqs. (9) and (11), the quantity $z\bar{B}_2$ for the syndrome s. These substitutions yield

$$e = tG + z(\bar{B}_2 B_2) \qquad (24)$$

in terms of received sequence z as the general solution of the syndrome equation.

In Eq. (24), let R be the $n \times n$ matrix $B_2 B_2$, since B_2 and $\bar{B}_2$ have ranks $(n - k)$, it can be shown that the matrix $R = \bar{B}_2 B_2$, where B_2 and $\bar{B}_2$ are defined in Eqs. (6) and (7), respectively, also has rank $(n - k)$. Substituting R into Eq. (24) yields

$$e = tG + zR \qquad (25)$$

as the general solution of the syndrome equation. Here, R is the $n \times n$, rank $(n - k)$ matrix

$$R = \bar{B}_2 B_2 \qquad (26)$$

t is an arbitrary k-vector of elements in $F[D]$, and z is the D-transform of the received sequence.

Let $z(D)$ be any finite-length received sequence. By the maximum likelihood principle, the most likely error sequence is the one with minimum Hamming weight. Given $z(D)$, the sequence $e(D)$ with minimum Hamming weight is found by minimizing the weight of the right side of Eq. (25) over all polynomials $t(D)$ in $F[D]$. That is,

$$\min \| e \| = \min \| tG + zR \|, \qquad t \in F[D] \qquad (27)$$

where $z = z(D)$ is the D-transform or polynomial of any finite-length received sequence and $\| x \|$ denotes the Hamming weight or "norm" of an element $x = x(D)$ in $F[D]$.

The minimization required in Eq. (27) is analogous to certain optimum nulling techniques in control theory. The sequence $r(D) = z(D)\,R(D)$ is the error sequence for $t(D) = 0$. What one attempts to do in Eq. (27) is to find that sequence $\hat{t}$ which, when encoded as $\hat{t}G$ and subtracted from $r(D)$, yields the sequence $\hat{e}$ of minimum Hamming weight. That is, if $\hat{t} = \hat{t}(D)$ is the D-transform for which $\| e \| = \| tG + zR \|$ is minimum, then•

$$\hat{e} = \hat{t}G + zR \qquad (28)$$

is the D-transform of the minimum-weight-possible error sequence.

By Eq. (4), the right inverse G^{-1} of the generating matrix G is

$$G^{-1} = B^{-1} \begin{bmatrix} I_k \\ \\ 0 \end{bmatrix} A^{-1} \qquad (29)$$

This is verified by multiplying G in Eq. (4) on the right by G^{-1} in Eq. (29). Multiplying both sides of Eq. (28) on the right by G^{-1} in Eq. (29) yields, by Eqs. (7) and (8), the identity

$$\begin{aligned} \hat{e}G^{-1} &= [\hat{t}G + z\bar{B}_2 B_2]\; G^{-1} \\ &= \hat{t} + z\bar{B}_2 B_2 \, [\bar{B}_1, B_2] \begin{bmatrix} I_k \\ \\ 0 \end{bmatrix} A^{-1} \\ &= \hat{t} + z\bar{B}_2 \, [0, I_{n-k}] \begin{bmatrix} I_k \\ \\ 0 \end{bmatrix} A^{-1} \\ &= \hat{t} \end{aligned} \qquad (30)$$

By Eq. (10), the subtraction of $\hat{e}$ from z produces a best estimate $\hat{y}$ of the transmitted code, i.e.,

$$\hat{y} = z - \hat{e} \qquad (31)$$

The best estimate $\hat{y}$ of the code, if multiplied on the right by G, yields

$$\hat{x} = \hat{y}G^{-1} \qquad (32)$$

which is the best estimate of the original message. Hence, substituting Eq. (31) in Eq. (32) and using Eq. (30) produces

$$\begin{aligned} \hat{x} &= (z - \hat{e})\,G^{-1} \\ &= zG^{-1} - \hat{t} \end{aligned} \qquad (33)$$

This important identity shows that $\hat{t} = \hat{t}(D)$, obtained by the minimization in Eq. (27), is a *correction factor* to the standard method of recovering the message from $z = z(D)$ if z were noise-free.

In the following section, the techniques of performing the minimization in Eq. (27) for finding $\hat{e}$ and $\hat{t}$ are discussed. Among these methods are the Viterbi dynamic programming algorithm and some of the sequential decoding techniques. Then, the syndrome-decoding algorithm described above is applied to systematic high-rate CCs and, in particular, to the one-error-correcting CC developed originally by Wyner and Ash (Ref. 5).

III. Syndrome Decoding of Systematic Convolutional Codes

The results of the preceding section are now applied to systematic convolutional codes. The generator matrix for a systematic CC has form

$$G(D) = [I_k, P(D)] \tag{34}$$

where I_k is the $k \times k$ identity matrix and $P(D)$ is a $k \times (n-k)$ matrix of polynomials over $GF(q)$ in the delay operator D. Again, as in the general case, the maximum degree M of the polynomials in $P(D)$ is called the memory of the code and $K = M + 1$ is the constraint length.

A parity-check matrix associated with $G(D)$ in Eq. (34) is the $(n-k) \times n$ matrix,

$$H(D) = [-P^T(D), I_{n-k}] \tag{35}$$

This follows from the fact that $H(D)$ has rank $n - k$ and that it satisfies Eq. (5).

The Smith formal form of Eq. (34) is, by Eq. (4),

$$\begin{aligned} G &= A\ [I_k, 0]\ B \\ &= [I_k, 0]\begin{bmatrix} I_k, P \\ 0, I_{n-k}\end{bmatrix} \end{aligned} \tag{36}$$

where $P = P(D)$, the matrix of polynomials in the generator matrix $G(D)$ in Eq. (34). Hence, for a systematic code, $A = I_k$ and

$$B = \begin{bmatrix} I_k, P \\ 0, I_{n-k}\end{bmatrix} \tag{37}$$

Because of the triangular form of B, the inverse is readily found to be

$$B^{-1} = \begin{bmatrix} I_k, -P \\ 0, I_{n-k}\end{bmatrix} \tag{38}$$

which actually equals B when the field of coefficients is the binary field $GF(2)$.

The partitions, given in Eqs. (6) and (7), of B and B^{-1}, respectively, are, for a systematic CC,

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

where

$$\begin{aligned} B_1 &= [I_k, P(D)] \\ B_2 &= [0, I_{n-k}] \end{aligned} \tag{39}$$

and

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

where

$$\begin{aligned} \bar{B}_1 &= \begin{bmatrix} I_k \\ 0 \end{bmatrix} \\ \bar{B}_2 &= \begin{bmatrix} -P \\ I_{n-k} \end{bmatrix} \end{aligned} \tag{40}$$

Note that the parity-check matrix found by Eq. (9) from $\bar{B}_2$ in Eq. (40) actually equals the parity-check matrix found already in Eq. (35) by satisfying Eq. (5). As a consequence, for a systematic CC, the syndrome s in Eq. (12) is

$$\begin{aligned} s &= z H^T \\ &= z\begin{bmatrix} -P \\ I_{n-k} \end{bmatrix} \\ &= [z_m, z_p]\begin{bmatrix} -P \\ I_{n-k} \end{bmatrix} \\ &= -z_m(D)P(D) + z_p(D) \end{aligned} \tag{41}$$

where $z_m(D)$ is the message code vector of k components, possibly corrupted by noise, and $z_p(D)$ is an $(n - k)$ component vector of parity symbols, also possibly changed by channel noise.

Next, by Eqs. (39) and (40), the matrix R in Eq. (26) is given by

$$\begin{aligned} R &= \bar{B}_2 B_2 \\ &= \begin{bmatrix} -P \\ I_{n-k} \end{bmatrix} [0, I_{n-k}] \\ &= \begin{bmatrix} 0, -P \\ 0, I_{n-k} \end{bmatrix} \end{aligned} \tag{42}$$

Thus, for a systematic CC, the general solution, Eq. (25), of the syndrome equation, Eq. (12), is, by substituting Eqs. (34) and (42) into Eq. (25),

$$\begin{aligned} e(D) &= t\,G + z\,R \\ &= t\,[I_k, P] + z \begin{bmatrix} 0, -P \\ 0, I_{n-k} \end{bmatrix} \\ &= [t\,I_k,\ t\,P(D)] + \left[0, z \begin{bmatrix} -P \\ I_{n-k} \end{bmatrix} \right] \\ &= [t(D), (t(D) - z_m(D))\,P(D) + z_p(D)] \end{aligned} \tag{43}$$

where $z_m(D)$ is the received message sequence "in the clear," $z_p(D)$ is the received parity sequence of the CC, and $t(D)$ is an element of $F[D]$. By Eq. (41), the above general solution, Eq. (43), of the syndrome equation for a systematic CC can be expressed in the alternate form

$$e(D) = [t(D), t(D)\,P(D) + s(D)] \tag{44}$$

where $s(D)$ is the syndrome, computed by Eq. (41) in terms of $z_m(D)$ and $z_p(D)$.

The "best" correction factor $\hat{t}(D)$ for all systematic CCs is found, as in Eq. (27), by minimizing the Hamming weight of $e(D)$, given in Eqs. (43) of (44). For low-rate systematic CCs, this minimization can be taken over $F[D]$, whereas for high-rate systematic CCs, this minimization need only be accomplished over a small subset, call it E, of $F[D]$, defined by error-bound constraints of the particular CC. This latter fact for high-rate systematic CCs will be demonstrated for the one-error correcting Wyner-Ash CC (Ref. 6). It is the very small size of the set E compared to the set $F[D]$ which makes syndrome decoding more efficient than the classical maximum likelihood method for decoding CCs.

Let $\hat{e}$ denote the error sequence of the solution, Eq. (44), of minimum Hamming weight, and let $\hat{t}$ be the element $t(D) \in F(D)$, for which the Hamming weight of $e(D)$ in Eq. (43) or Eq. (44) is minimum. Then, by Eqs. (43) and (44), as in Eq. (28), $\hat{e}$ and $\hat{t}$ are related by

$$\begin{aligned} \hat{e} &= [\hat{t}, (\hat{t} - z_m)\,P + z_p] \\ &= [\hat{t}, \hat{t}\,P + s] \end{aligned} \tag{45}$$

By Eqs. (29), (36), and (38), the right inverse of the generator matrix G in Eq. (34) is

$$\begin{aligned} G^{-1} &= B^{-1} \begin{bmatrix} I_k \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} I_k, & -P \\ 0, & I_{n-k} \end{bmatrix} \begin{bmatrix} I_k \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} I_k \\ 0 \end{bmatrix} \end{aligned} \tag{46}$$

Hence, by Eqs. (45) and (46), the relation

$$\hat{e}\,G^{-1} = \hat{t}$$

given in Eq. (30), also holds for systematic CCs. Again, the subtraction of $\hat{e}$ from z produces

$$\hat{y} = z - \hat{e}$$

as the best estimate of transmitted code, so that

$$\begin{aligned}\hat{x} &= \hat{y}\,G^{-1}\\ &= (z-\hat{e})\,G^{-1}\\ &= z\,G^{-1}-\hat{t}\\ &= [z_m, z_p]\begin{bmatrix} I_k \\ 0 \end{bmatrix}\hat{t}\\ &= z_m - \hat{t} \qquad (47)\end{aligned}$$

as the best estimate of the received message in terms of z_m, the received message "in the clear," and the correction factor, $\hat{t}$.

IV. Error Trellis Syndrome Decoding of Wyner-Ash Convolutional Code

The Wyner-Ash one-error correction codes were first defined in Ref. 5. A more modern and understandable development can be found in Blahut's recent book (Ref. 7). Instead of defining the CC only in terms of its infinite generator or parity-check matrix, as is done in Ref. 7, here the infinite matrices are converted first into compact matrices in terms of the delay operator D. If

$$G(D) = G_0 + G_1 D + \ldots, + G_m D^m \qquad (48)$$

is a generator matrix of a CC of memory $M = m$, as defined in Eq. (3), then evidently

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \ldots & G_m & 0 & 0 & \ldots \\ 0 & G_0 & G_1 & G_2 & \ldots & G_m & 0 & \ldots \\ 0 & 0 & G_0 & G_1 & G_2 & \ldots & G_m & \ldots \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \end{bmatrix} \qquad (49)$$

is the infinite generator matrix associated with $G(D)$. Thus, a systematic code with generator matrix $G(D) = [I_k, P(D)]$ has

$$G = \begin{bmatrix} I_k & P_0 & 0 & P_1 & 0 & P_2 & \ldots & 0 & P_m & & & & \\ & & I_k & P_0 & 0 & P_1 & 0 & P_2 & \ldots & 0 & P_m & & \\ & & & & I_k & P_0 & 0 & P_1 & 0 & P_2 & \ldots & 0 & P_m \\ & & & & & & \cdot & \cdot & \cdot & & & & \\ & & & & & & & \cdot & \cdot & \cdot & & & \\ & & & & & & & & \cdot & \cdot & \cdot & & \end{bmatrix} \qquad (50)$$

as its companion infinite generator matrix, where

$$P(D) = P_0 + P_1 D + \ldots, P_m D^m \qquad (51)$$

where 0 is the $k \times k$ all-zero matrix and P_i are $k \times k\ (n-k)$ matrices. Since, by Eq. (35), $H(D) = [-P^T(D), I_{n-k}]$ is a parity-check matrix of $G(D)$, given above, the associated infinite parity-check matrix is

$$H = \begin{bmatrix} P_0^T & I & & & & & \\ P_1^T & 0 & P_0^T & I & & & \\ P_2^T & 0 & P_1^T & 0 & P_0^T & I & \\ \vdots & \vdots & P_2^T & 0 & P_1^T & 0 & \ddots \\ P_m^T & 0 & \vdots & \vdots & \vdots & \vdots & \\ & & P_m^t & 0 & \vdots & \vdots & \\ & & & & P_m^T & 0 & \\ & & & & & & \ddots \end{bmatrix} \qquad (52)$$

The results in Eqs. (50) and (52) are given in Ref. 6 in the same notation.

In terms of Eqs. (51) and (52), Blahut defines an $(n, k) = (2^m, 2^m - 1)$ Wyner-Ash code as follows: Let H^1 be the parity-check matrix of the binary $(2^m - 1, 2^m - 1 - m)$ Hamming one-error-correcting block code. Choose matrices $P_1^T, P_2^T, \ldots, P_m^T$ to be the m rows of the parity-check matrix H^1, i.e.,

$$H^1 = \begin{bmatrix} P_1^T \\ P_2^T \\ \cdot \\ \cdot \\ \cdot \\ P_m^T \end{bmatrix}$$

$$= [P_1, P_2, \ldots, P_m]^T \tag{53}$$

Finally, let P_0^T be a vector of $2^m - 1$ ones, i.e.,

$$P_0^T = \underbrace{[1, 1, 1, \ldots, 1]}_{2^m - 1} \tag{54}$$

Blahut shows (Ref. 7, Theorem 12.5.1) that the minimum distance of the Wyner-Ash code is 3 and, as a consequence, it will correct at least one error. To understand this code in more detail and to apply the decoding technique developed in the last section to it, consider now an example for $m = 2$.

Example: The $m = 2$, the parity-check matrix of the Hamming code, is

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

so that, by Eqs. (53) and (54), $P_0^T = [1 \;\; 1 \;\; 1]$, $P_1^T = [1 \;\; 1 \;\; 0]$, and $P_2^T = [1 \;\; 0 \;\; 1]$. Thus, by Eqs. (51),

$$P(D) = \begin{bmatrix} 1 + D + D^2 \\ 1 + D \\ 1 \quad\;\; + D^2 \end{bmatrix}$$

and, by Eqs. (34) and (35),

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

and

$$H(D) = [1 + D + D^2, 1 + D, 1 + D^2, 1] \tag{55}$$

are the generator and parity-check matrices of the (4, 3) Wyner-Ash CC, respectively. Also, by Eqs. (37) and (38)

$$B = B^{-1}$$

$$= \begin{bmatrix} I_k, P(D) \\ 0, I_{n-k} \end{bmatrix}$$

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

so that, by Eqs. (39) and (40), $B_2 = [0 \;\; 0 \;\; 0 \;\; 1]$ and $\vec{B}_2 = H^T$ and, finally, by Eq. (42),

$$R = \vec{B}_2 \, B_2 \equiv \begin{bmatrix} 1 + D + D^2 \\ 1 + D \\ 1 \quad\;\; + D^2 \\ 1 \end{bmatrix} \quad [0 \;\; 0 \;\; 0 \;\; 1]$$

$$= \begin{bmatrix} 0 \; 0 \; 0, & 1 + D + D^2 \\ 0 \; 0 \; 0, & 1 + D \\ 0 \; 0 \; 0, & 1 \quad\;\; + D^2 \\ 0 \; 0 \; 0, & 1 \end{bmatrix} \tag{56}$$

The above results for this 3/4 rate CC can now be used to explicitly obtain the general solution $e(D)$ in Eq. (43) of the syndrome equation. This is accomplished by substituting Eqs. (55) and (56) into Eq. (25) or directly from Eq. (43). The result is

$$\begin{aligned} e(D) \equiv e &= [e_1, e_2, e_3, e_4] \\ &= [t, (t_1 + z_1)(1 + D + D^2) \\ &\quad + (t_2 + z_2)(1 + D) \\ &\quad + (t_3 + z_3)(1 + D^2) + z_4] \end{aligned} \tag{57}$$

where

$$t(D) \equiv t = [t_1, t_2, t_3] \tag{58}$$

By Eqs. (41) and (44), e in Eq. (57) can also be expressed more compactly as

$$e = [t, r + s] \tag{59}$$

where s is the syndrome,

$$\begin{aligned} s(D) \equiv s = {} & z_1(1 + D + D^2) + z_2(1 + D) \\ & + z_3(1 + D^2) + z_4 \end{aligned} \tag{60}$$

and

$$\begin{aligned} r(D) \equiv r = {} & t_1(1 + D + D^2) + t_2(1 + D) \\ & + t_3(1 + D^2) \end{aligned} \tag{61}$$

Note that the term $r(D) \equiv r$ in Eqs. (59) and (61), in order to minimize the Hamming weight of $e(D)$, must be chosen to "cancel $s(D)$" in Eq. (59). For this reason, one might call $r(D)$ the *regulator* needed to cancel the syndrome $s(D)$.

Now, the formal power series for $e(D)$ in the delay operator is explicitly

$$\begin{aligned} e(D) &= [e_1(D), e_2(D), \ldots, e_n(D)] \\ &= \sum_{j=0}^{\infty} [e_{1j}, e_{2j}, \ldots, e_{nj}]\, D^j \end{aligned} \tag{62}$$

Define the truncation of $e(D)$ at stage or frame time N in terms of Eq. (62) as

$$[e(D)]_N = \sum_{j=0}^{N} [e_{1j}, e_{2j}, \ldots, e_{nj}]\, D^j \tag{63}$$

Thus, the Hamming weight of the sequence of possible errors in N frames is

$$\begin{aligned} \| [e(D)]_N \| &= \sum_{j=0}^{N} \| [e_{1j}, e_{2j}, \ldots, e_{nj}] \| \\ &= \sum_{j=0}^{N} \| \operatorname*{coef}_{D^j} [e(D)] \| \end{aligned} \tag{64}$$

where the latter expression under the summation is the Hamming weight of the coefficient of the jth power of D.

By Eqs. (57) and (64) for this particular example of convolutional code,

$$\operatorname*{coef}_{D^j} [e(D)] = [t_{1j}, t_{2j}, t_{3j}, r_j + s_j] \tag{65}$$

where

$$\begin{aligned} r_j = {} & t_{1,j} + t_{1,j-1} + t_{1,j-2} \\ & + t_{2,j} + t_{2,j-1} \\ & + t_{3,j} \qquad\quad + t_{3,j-2} \end{aligned} \tag{66}$$

is the regulator function at frame j, and

$$\begin{aligned} s_j = {} & z_{1j} + z_{1,j-1} + z_{1,j-2} \\ & + z_{2j} + z_{2,j-1} \\ & + z_{3j} \qquad\quad + z_{3,j-2} \\ & + z_{4j} \end{aligned} \tag{67}$$

is the syndrome function at frame j in terms of binary variables t_{ij} and z_{ij}, defined by,

$$\begin{aligned} r(D) &= \sum_{j=0}^{\infty} [t_{1j}, t_{2j}, \ldots, t_{kj}] D^j \\ &= \sum_{j=0}^{\infty} \underline{t}_j D^j \end{aligned}$$

$$z(D) = \sum_{j=0}^{\infty} [z_{1j}, z_{2j}, \ldots, z_{nj}] D^j$$

Note in Eq. (66) that r_j is a function of $\underline{t}_j = [t_{1j}, t_{2j}, t_{3j}]$, $\underline{t}_{j-1} = [t_{1,j-1}, t_{2,j-1}, t_{3,j-1}]$ and $\underline{t}_{j-2} = [t_{1,j-2}, t_{2,j-2}, t_{3,j-2}]$. That is, at frame j, r_j is a function of $\underline{t}_j$ at frame j; a function of $\underline{t}_{j-1}$ at frame $j-1$; and a function of t_{j-2} at frame $j-2$; i.e.,

$$r_j = r(\underline{t}_j, \underline{t}_{j-1}, \underline{t}_{j-2}) \tag{68}$$

If the values of the regulator function r_j at frame j are imagined to be generated by a sequential circuit, then the pair

$$\sigma_j = (\underline{t}_{j-1}, \underline{t}_{j-2}) \tag{69}$$

constitutes the values of the internal state of the circuit and vector $\underline{t}_j$ is the jth input to the circuit.

Let the sequential circuit with output

$$u_j = [\underline{t}_j, r(\underline{t}_j, \sigma_j)] \tag{70}$$

be the regulator circuit of the decoder, where σ_j is the internal state defined by Eq. (69). Also, call the set of all allowable paths generated by Eq. (70) the regulator tree or trellis. Finally, by Eq. (59), the error trellis of the code is, for all paths generated,

$$v_j = [\underline{t}_j, s_j + r(\underline{t}_j, \sigma_j)] \tag{71}$$

To illustrate the above concepts, let the input to the present example of the (4, 3) CC be

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

i.e., $x_1 = [1\ 0\ 1\ 0\ 1] = x_2 = x_3$. By the generating matrix given in Eq. (55), the output $y = [y_1, y_2, y_3, y_4]$ is obtained in what follows: $y_1 = y_2 = y_3 = x_1 = [1\ 0\ 1\ 0\ 1]$, and $y_4 = (1 + D + D^2)\, x_1 + (1 + D)\, x_2 + (1 + D^2)\, x_3$. Explicitly, y_4 is computed from this relation as follows:

$$\begin{array}{r l ccccccc}
x_1 & : & 1 & 0 & 1 & 0 & 1 & & \\
Dx_1 & : & & 1 & 0 & 1 & 0 & 1 & \\
D^2 x_1 & : & & & 1 & 0 & 1 & 0 & 1 \\
x_2 & : & 1 & 0 & 1 & 0 & 1 & & \\
Dx_2 & : & & 1 & 0 & 1 & 0 & 1 & \\
x_3 & : & 1 & 0 & 1 & 0 & 1 & & \\
D^2 x_3 & : & & & 1 & 0 & 1 & 0 & 1
\end{array}$$

$$y_4 = [1\ 0\ 1\ 0\ 1\ 0\ 0]$$

Thus, the output of the encoder is

$$\begin{aligned} y = [&1\ 1\ 1\ 1, 0\ 0\ 0\ 0, 1\ 1\ 1\ 1, \\ &0\ 0\ 0\ 0, 1\ 1\ 1\ 1] \end{aligned} \tag{72}$$

Assume y, given in Eq. (72), is transmitted over a binary symmetric channel with probability of error somewhat less than $1/12 = 0.0833 \ldots$. Then, suppose that the received coded sequence is

$$\begin{aligned} z = [&1\ 1\ 0\ 1, 0\ 0\ 0\ 0, 1\ 1\ 1\ 1, \\ &0\ 0\ 0\ 0, 0\ 1\ 1\ 1] \end{aligned} \tag{73}$$

i.e., $z_1 = [1\ 0\ 1\ 0\ 0]$, $z_2 = [1\ 0\ 1\ 0\ 1]$, $z_3 = [0\ 0\ 1\ 0\ 1]$ and $z_4 = [1\ 0\ 1\ 0\ 1]$. By Eq. (60), the syndrome sequence for this value of received sequence is computed to be

$$s = [1\ 0\ 1\ 0\ 1\ 1\ 1] \tag{74}$$

by the same method used above to obtain y_4.

It is shown in Ref. 7 (p. 366) that the rate 3/4 code of this example can correct one error in every 3 frame times or code length of 12. As a consequence, one needs only to correct one error every 3 frames. This limits the number of values of $t = [t_1, t_2, t_3]$ to 4, namely the values

$$\begin{aligned} &[0\ 0\ 0] \equiv 0, \quad [1\ 0\ 0] \equiv 1 \\ &[0\ 1\ 0] \equiv 2, \quad [0\ 0\ 1] \equiv 3 \end{aligned} \tag{75}$$

Note that the four values of t in Eq. (75) allow for, at most, one error, and that these four values are conveniently labeled by the integers $t = 0, 1, 2$, or 3.

Figure 1 shows a constrained regulator trellis with outputs $[t, r]$. In Fig. 1, note that, because of the limited error-correction capability of the code, the number of internal states $\sigma = (Dt, D^2 t)$ of the regulator circuit can be limited to 7 out of a possible 64. Moreover, the number of state transitions can be limited to those shown in Fig. 1 for the regulator trellis. The branches of the regulator trellis are labeled with the value $[t, r]$. For example, the branch from state $\sigma = [0\ 0]$ to $\sigma = [3\ 0]$ is labeled by $[t, r] = [3, 1] \equiv [0, 0, 1, 1]$, which means $t_1 = 0$, $t_2 = 0$, $t_3 = 1$, and $r = 1$.

To decode the message in Eq. (73), by Eq. (70) an error trellis is created by adding the vector $[0, s]$ to all labels in the regulator trellis where s is the syndrome value. Thus, in Fig. 2, the values of $[0, s]$, where s is the syndrome value in Eq. (74), appear on all possible transitions $\sigma = [0\ 0]$ to $\sigma = [0\ 0]$ on the top line of the error trellis. At each node, the cumulative Hamming weight of the path, passing through that node, is written.

The Hamming weight at each node, plus the weight of a possible branching from that to the next node, is used to eliminate branches. The technique is similar to the method in Viterbi decoding for eliminating branches. To illustrate, in Fig. 2 there are four branches at Frame 2 which could go to state or node $\sigma = [0\ 0]$. The transition is chosen as the branch from $\sigma = [0\ 3]$ to $\sigma = [0\ 0]$ since the node weight 2 plus branch weight 0 is 2, the *minimum* of the 4 possible transitions.

The minimum overall path weight of the error trellis in Fig. 2 is

$$[0\ 0, 3\ 0, 0\ 3, 0\ 0, 0\ 0, 1\ 0, 0\ 1, 0\ 0, 0\ 0]$$

in terms of state values $\sigma = Dt,\ D^2t$. Hence, based on the criterion of Eq. (27), the best estimate of t is

$$\begin{aligned}\hat{t} &= [3, 0, 0, 0, 1, 0, 0, 0]\\ &= [0\ 0\ 1, 0\ 0\ 0, 0\ 0\ 0, 0\ 0\ 0,\\ &\quad 1\ 0\ 0, 0\ 0\ 0]\end{aligned}$$

If this vector is added component-wise to z in Eq. (73), the message is corrected to yield $\hat{x} = x$, the original message.

References

1. Reed, I. S. and Truong, T. K., "New Syndrome Decoding for $(n, 1)$ Convolutional Codes," *Electronic Letters*, Vol. 19, No. 9, April 1983, pp. 344-346.
2. Reed, I. S. and Truong, T. K., "New Syndrome Decoding Techniques for Convolutional Codes Over *GF*(q)," to be published in *Proceedings IEE.*
3. Massey, J. L. and Sain, M. K., "Inverses of Linear Sequential Circuits," *IEEE Trans. Comput. C-17*, pp. 330-337, April 1968.
4. Forney, C. D., Jr., "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Info. Theor. IT-9*, 1963, pp. 64-74.
5. Vinck, A. J., de Paepe, A. J. P., and Schalkwijk, J. P. M., "A Class of Binary Rate One-Half Convolutional Codes that Allows an Improved Stack Decoder," *IEEE Trans. Info. Theor. IT-26*, No. 4, 1980, pp. 389-392.
6. Wyner, A. D. and Ash, R. B., "Analysis of Recurrent Codes," *IEEE Trans. Info. Theor. IT-9*, 1963, pp. 143-156.
7. Blahut, R. E., *Theory and Practice of Error Control Codes*, Addison-Wesley, London, 1983.

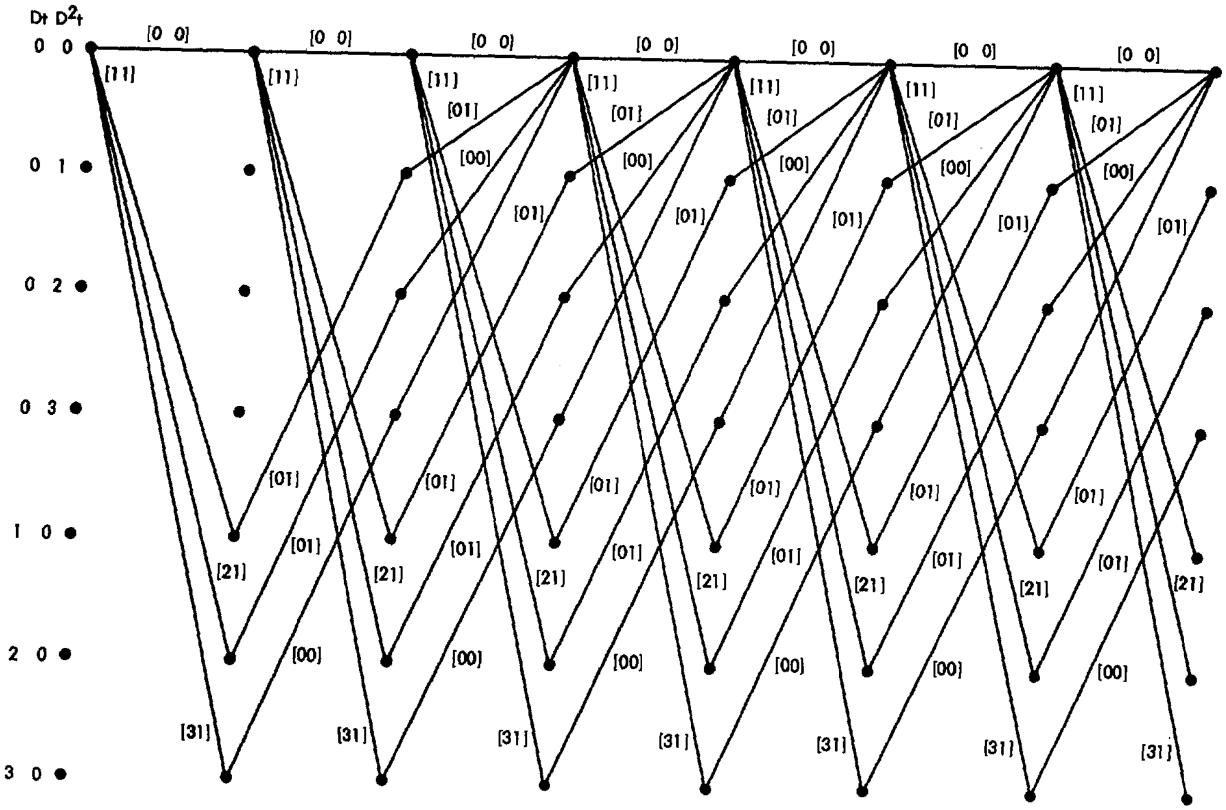

Fig. 1. Constrained regulator trellis with outputs $[t, r]$, where $r = t_1(1 + D + D^2) + t_2(1 + D) + t_3(1 + D^2) \equiv r(t, \sigma)$ and where $t = [t_1, t_2, t_3] = [0\ 0\ 0] \equiv 0$, $t = [1\ 0\ 0] \equiv 1$, $t = [0\ 1\ 0] \equiv 2$, and $t = [0\ 0\ 1] \equiv 3$

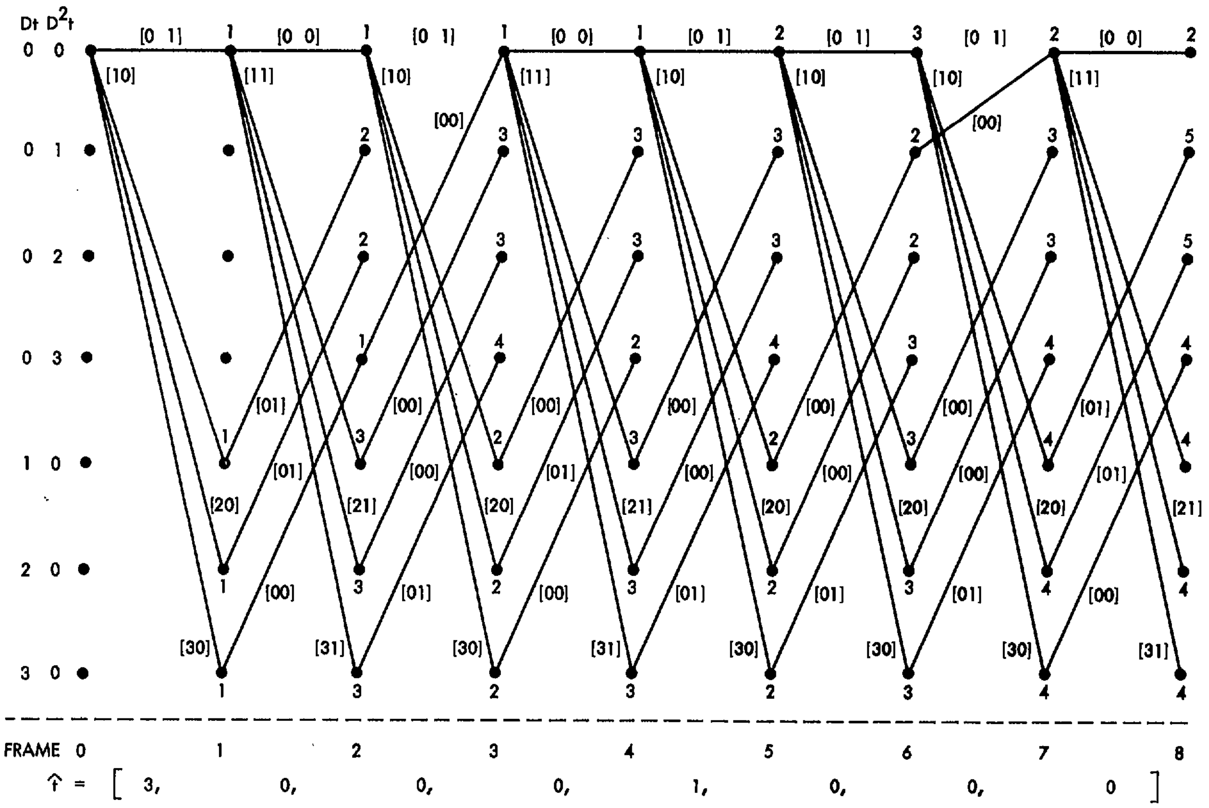

Fig. 2. Error trellis with input and state-transition constraints for one-error-correcting Wyner-Ash convolutional code