It is further noted that the bounds derived do not depend on the nature of the noise or on the channel impulse response for given L, and so will apply generally for equally likely transmission.

REFERENCES

[1] W. Feller, *An Introduction to Probability Theory and its Applications*, first ed. New York: Wiley, 1950, vol. I.

[2] D. L. Duttweiler, J. E. Mazo, and D. G. Messerschmitt, "An upper bound on the error probability in decision-feedback equalization," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 490–497, July 1974.

[3] G. Salomonsson, "An equalizer with feedback filter," *Ericsson Technics*, no. 2, pp. 57–101, 1972.

[4] M. E. Austin, "Decision feedback equalization for digital communication over dispersive channels," Electronics, Mass. Inst. Technol., Cambridge, Tech. Rep. 461, Aug. 11, 1967.

[5] D. A. George, R. R. Bowen, and J. R. Storey, "An adaptive decision feedback equalizer," *IEEE Trans. Commun. Tech.*, vol. COM-19, pp. 281–293, June 1971.

[6] Y. Anzai, "A note on reachability of discrete-time quantized control systems," *IEEE Trans. Automat. Contr.* (Short Papers), vol. AC-19, pp. 575–577, Oct. 1974.

[7] M. Tomlinson, "New automatic equalizer employing modulo arithmetic," *Electron. Lett.*, vol. 7, no. 5/6, pp. 138–139, Mar. 25, 1971.

[8] R. Price, "Nonlinearly feedback-equalized PAM vs. capacity for noisy filter channels," in *Conf. Rec., IEEE Int. Conf. Commun.*, Philadelphia, PA, June 1972, pp. 19–21.

[9] J. W. Mark and S. S. Haykin, "Adaptive equalization for digital communication," *Proc. Inst. Elec. Eng.* (London), vol. 118, pp. 1711–1720, Dec. 1971.

[10] P. Vidal, *Nonlinear Sampled Data Systems.* New York: Gordon and Breach, 1971.

[11] P. Monsen, "Feedback equalization for fading dispersive channels," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 56–64, Jan. 1971.

[12] H. Kobayashi and D. T. Tang, "A decision feedback receiver for channels with strong intersymbol interference," *IBM J. Res. Develop.*, vol. 17, pp. 413–419, Sept. 1973.

[13] P. Monsen, "Adaptive equalization of the slow fading channel," *IEEE Trans. Commun.*, vol. COM-22, pp. 1064–1075, Aug. 1974.

[14] A. Cantoni and P. Butler, "Behaviour of decision feedback inverses with deterministic and random inputs," Univ. Newcastle, Newcastle, Australia, Tech. Rep. EE7505, Mar. 1975.

★

Antonio Cantoni (M'74), for a photograph and biography please see p. 809 of the August 1976 issue of this TRANSACTIONS.

★

Paul Butler (S'72–M'75), for a photograph and biography please see p. 809 of the August 1976 issue of this TRANSACTIONS.

Syndrome Decoding of Binary Rate-½ Convolutional Codes

J. PIETER M. SCHALKWIJK, MEMBER, IEEE, AND A. J. VINCK

Abstract—**The classical Viterbi decoder recursively finds the trellis path (code word) closest to the received data. Given the received data, the syndrome decoder first forms a syndrome instead. Having found the syndrome, that only depends on the channel noise, a recursive algorithm like Viterbi's determines the noise sequence of minimum Hamming weight that can be a possible cause of this syndrome. Given the estimate of the noise sequence, one derives an estimate of the original data sequence. The bit error probability of the syndrome decoder is no different from that of the classical Viterbi decoder. However, for short constraint length codes the syndrome decoder can be implemented using a read-only memory (ROM), thus obtaining a considerable saving in hardware. The syndrome decoder has at most $\frac{3}{4}$ as many path registers as does the Viterbi decoder. There exist convolutional codes for which the number of path registers can be even further reduced.**

Paper approved by the Editor for European Contributions of the IEEE Communications Society for publication after presentation at the Information Theory Colloquium, Ketzhely, Hungary. Manuscript received September 29, 1975; revised April 15, 1976. This research was supported in part by the U.S. Navy under Contract N00123-75-C-0557.

The authors are with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.

I. INTRODUCTION

THIS paper extends and generalizes some earlier results [1] on syndrome decoding of rate-$\frac{1}{2}$ convolutional codes. The binary code generated by the encoder of Fig. 1 will again be used as an example throughout the paper. The additions in Fig. 1 are modulo-2 and all binary sequences $\cdots, b_{-1}, b_0, b_1, \cdots$ are represented as power series $b(\alpha) = \cdots + b_{-1}\alpha^{-1} + b_0 + b_1\alpha + \cdots$. The encoder has connection polynomials $C_1(\alpha) = 1 + \alpha^2$ and $C_2(\alpha) = 1 + \alpha + \alpha^2$. In general, the encoder outputs are $C_1(\alpha)x(\alpha)$ and $C_2(\alpha)x(\alpha)$. The syndrome $z(\alpha)$ only depends on $n_1(\alpha)$ and $n_2(\alpha)$, i.e., not on the data sequence $x(\alpha)$, for

$$z(\alpha) = C_2(\alpha)[C_1(\alpha)x(\alpha) + n_1(\alpha)] + C_1(\alpha)[C_2(\alpha)x(\alpha) + n_2(\alpha)]$$
$$= C_2(\alpha)n_1(\alpha) + C_1(\alpha)n_2(\alpha). \quad (1)$$

Having formed the syndrome $z(\alpha)$, Section III describes a recursive algorithm like Viterbi's [2] to determine from the

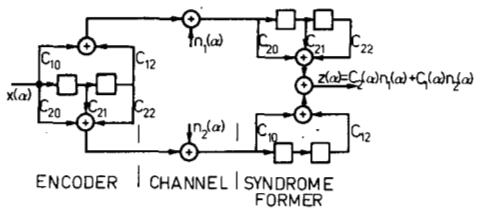

Fig. 1. Encoding and syndrome forming for a $R = \frac{1}{2}$ code.

syndrome $z(\alpha)$ the noise sequence pair $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ of minimum Hamming weight that can be a possible cause of this syndrome.

Given the estimate $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ of the noise sequence pair, one derives an estimate $\hat{x}(\alpha)$ of the original data sequence $x(\alpha)$ as follows. For a noncatastrophic code, $C_1(\alpha)$ and $C_2(\alpha)$ are relatively prime. Hence, by Euclid's algorithm [3] there exist polynomials $D_1(\alpha)$ and $D_2(\alpha)$ such that $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$. For the example of Fig. 1, we have $D_1(\alpha) = 1 + \alpha$, $D_2(\alpha) = \alpha$. We receive the sequence pair

$$y_i(\alpha) = C_i(\alpha)x(\alpha) + n_i(\alpha), \qquad i = 1, 2, \tag{2}$$

and form the estimate

$$\hat{x}(\alpha) = D_1(\alpha)[y_1(\alpha) + \hat{n}_1(\alpha)] + D_2(\alpha)[y_2(\alpha) + \hat{n}_2(\alpha)]. \tag{3}$$

Note that if the noise sequence estimate $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ is correct we have

$$\begin{aligned} y_i(\alpha) + \hat{n}_i(\alpha) &= C_i(\alpha)x(\alpha) + n_i(\alpha) + \hat{n}_i(\alpha) \\ &= C_i(\alpha)x(\alpha), \qquad i = 1, 2, \end{aligned}$$

and, hence

$$\hat{x}(\alpha) = D_1(\alpha)C_1(\alpha)x(\alpha) + D_2(\alpha)C_2(\alpha)x(\alpha) = x(\alpha).$$

Note that (3) for the estimate $\hat{x}(\alpha)$ of the data sequence $x(\alpha)$ can be rewritten as

$$\hat{x}(\alpha) = [D_1(\alpha)y_1(\alpha) + D_2(\alpha)y_2(\alpha)] + \hat{\omega}(\alpha), \tag{4}$$

where

$$\omega(\alpha) = D_1(\alpha)n_1(\alpha) + D_2(\alpha)n_2(\alpha), \tag{5}$$

and $\hat{\omega}(\alpha)$ equals the right-hand side (RHS) of (5) with $\hat{n}_i(\alpha)$ substituted for $n_i(\alpha)$, $i = 1, 2$. The term in square brackets in (4) can be computed directly from the received data using very simple circuitry. As there is no need to distinguish between pairs $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ and $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]'$ that lead to the same value for $\hat{\omega}(\alpha)$ in (4), the algorithm to be discussed in Section III computes $\hat{\omega}(\alpha)$ directly.

II. STATE DIAGRAM

In Fig. 2 we have redrawn the syndrome former of our example. As, according to (1), the syndrome $z(\alpha)$ only depends on the noise pair $[n_1(\alpha),n_2(\alpha)]$, all other binary

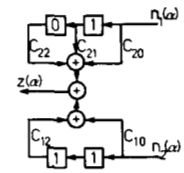

Fig. 2. Syndrome former in its base state.

sequences have been omitted from Fig. 2. For minimum distance decoding we are now presented with the following problem. Given the syndrome $z(\alpha)$, determine the noise pair $[\hat{n}_1(\alpha), \hat{n}_2(\alpha)]$ of minimum Hamming weight that can be a cause of this syndrome. Before tackling this problem in Section III, it will be necessary to first derive some general properties of the state diagram of a syndrome former for a binary rate $-\frac{1}{2}$ convolutional code.

Let ν be the number of memory stages of the encoder, i.e., $\nu = 2$ for the encoder of Fig. 1. The corresponding syndrome former of Fig. 2 has $2^{2\nu} = 16$ "physical states," where a physical state is defined as the contents $S = [s_1(\alpha),s_2(\alpha)]$, where

$$s_i(\alpha) = s_{i,-\nu}\alpha^{-\nu} + s_{i,-\nu+1}\alpha^{-\nu+1} + \cdots + s_{i,-1}\alpha^{-1}, \qquad i = 1, 2,$$

of the $2\nu = 4$ memory cells of the syndrome former. Thus, at first sight, the state diagram of the syndrome former appears more complicated than the state diagram used to implement the classical Viterbi decoder [2], that has only $2^\nu = 4$ states. However, on closer inspection, it turns out that the $2^{2\nu} = 16$ physical states of the syndrome former can be divided into $2^\nu = 4$ equivalence classes or "abstract states," where any two physical states in the same equivalence class give the same output $z(\alpha)$ irrespective of the input pair $[n_1(\alpha),n_2(\alpha)]$. In general, to prove the existence of the 2^ν, $\nu = 1, 2, \cdots$, abstract states defined above, we need the following definitions. As the syndrome former is a time invariant circuit we assume without loss of generality that the state S is present at time $t = 0$.

Definition 1: A "zero-equivalent" state is a physical state with the property that if the syndrome former is in such a state, an all-zero input $[n_1(\alpha),n_2(\alpha)]_0^\infty$ gives rise to an all-zero output $[z(\alpha)]_0^\infty$, where $[b(\alpha)]_{k_1}^{k_2}$ indicates that part of the power series $b(\alpha)$ for which $k_1 \leqslant \exp \alpha \leqslant k_2$.

Definition 2: A "base" state $S^b = [\alpha^{-1},s_2{}^b(\alpha)]$ is a zero-equivalent state with a single "1" in the rightmost position of the top register of the syndrome former, see Fig. 2.

A base state can be constructed as follows. Start with the top and bottom registers of the syndrome former, Fig. 2, all zero. Put a 1 in the leftmost position of the top register. Assuming that $C_1(\alpha)$ and $C_2(\alpha)$ both have a nonzero term of degree ν, we now have to put a 1 in the leftmost position of the bottom register, as otherwise the corresponding digit of the syndrome $z(\alpha)$ would differ from zero. Subsequently, shift the contents of the top and the bottom register one place to the right, feeding 0's into both leftmost positions, respectively. If the corresponding digit of the syndrome $z(\alpha)$ differs from zero set the leftmost position of the bottom register equal to 1, thus complementing the corresponding syndrome digit, etc. This process continues until the single 1 in the top register is in the rightmost position. The bottom register now contains

$s_2{}^b(\alpha)$. It is clear that the above construction leads to a unique result. This base state S^b is indicated in the example of Fig. 2. However, there might conceivably be another construction leading to a different base state. The following theorem shows that this is not the case.

Theorem 1: The base state $S^b = [\alpha^{-1}, s_2{}^b(\alpha)]$ is unique.

Proof: Suppose there are two base states $S_1{}^b$ and $S_2{}^b$. These base states are zero-equivalent states, hence, so is their sum $S = S_1{}^b + S_2{}^b$. But as the sum of two base states the physical state S has all zeros in the top register of the syndrome former

As $C_1(\alpha)$ has a nonzero term of degree ν, the only zero-equivalent state with the top register contents all zero is the all-zero state. Hence, S is the all-zero state and the physical states $S_1{}^b$ and $S_2{}^b$ are equal. Q.E.D.

We will now show that there are 2^ν equivalence classes of physical states and that each class has a unique representative physical state for which the contents of the top register of the syndrome former, Fig. 2, is all zero. It is these representative states $S = [0, s_2(\alpha)]$, to be referred to as "the states," that will be used in the remainder of the paper.

Theorem 2: The $2^{2\nu}$, $\nu = 1, 2, \cdots$, physical states of the syndrome former, corresponding to a binary rate-$\frac{1}{2}$ encoder with ν memory cells, can be divided into 2^ν equivalence classes or abstract states. Each equivalence class has a unique representative physical state $S = [0, s_2(\alpha)]$ for which the top register, see Fig. 2, of the syndrome former is all zero.

Proof: Two physical states were related if they resulted in the same output $z(\alpha)$ irrespective of the input pair $[n_1(\alpha), n_2(\alpha)]$. To prove that this relation is an equivalence relation, we must show that it is reflexive, symmetric, and transitive. Reflexivity and symmetry are obvious. To show transitivity let S_1 be related to S_2 and S_2 be related to S_3. Since S_1 and S_2 produce the same output $z(\alpha)$, their sum $S_1 + S_2$ is a zero-equivalent state, as is $S_2 + S_3$. But $S_1 + S_3 = (S_1 + S_2) + (S_2 + S_3)$. Hence, $S_1 + S_3$ is the sum of two zero-equivalent states and thus also zero equivalent. In other words, the physical states S_1 and $S_1 + (S_1 + S_3) = S_3$ produce the same output $z(\alpha)$ and thus the relation defined above is transitive. This completes the first part of the proof. The relation defined above is an equivalence relation and, hence, divides the set of physical states into equivalence classes or abstract states. As the sum of zero-equivalent states is again zero equivalent, left shifts of the base state S^b can be added to the all zero state to obtain a zero-equivalent state for which the top register has any desired contents. Two zero-equivalent states S_1 and S_2 that have the same top register contents are identical, for their sum $S = S_1 + S_2$ is a zero-equivalent state with top register contents all zero. Hence, as shown in the proof of Theorem 1, S is the all-zero state and thus $S_1 = S_2$. In other words, there are 2^ν zero-equivalent states and, in fact, all equivalence classes have 2^ν members, giving $2^{2\nu}/2^\nu = 2^\nu$ abstract states. As we can add left shifts of the base state S^b to any particular physical state without leaving its equivalence class, each equivalence class has a representative member, called the state $S = [0, s_2(\alpha)]$, that has the contents of the top register, Fig. 2, of the syndrome former all zero. To prove uniqueness of the representative state within an equivalence class, assume two representative states S_1 and S_2. The sum $S_1 + S_2$ of two representative states S_1 and S_2 within the same equivalence class is a zero-equivalent state with top register contents all zero. This sum state again must be the all-zero state. Hence, $S_1 + S_2 = 0$ or $S_1 = S_2$, proving that the representative state of an equivalence class is unique. Q.E.D.

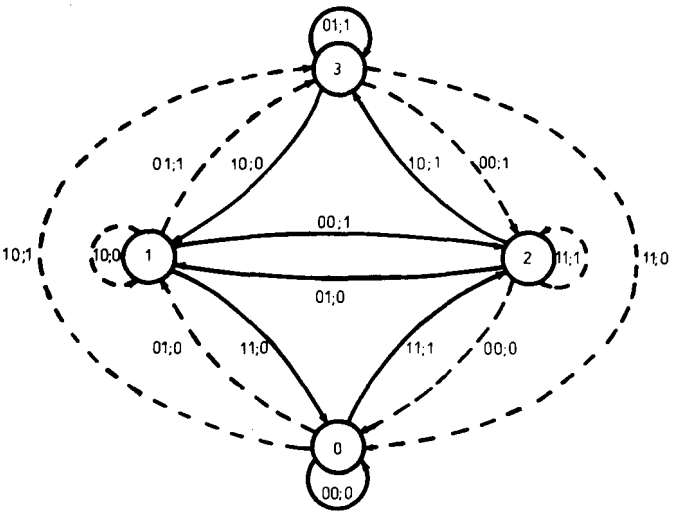

Fig. 3. State diagram of syndrome former.

We are now ready, as an example, to construct the state diagram, see Fig. 3, of the syndrome former of Fig. 2. The states $S_0 = [0,0]$, $S_1 = [0, \alpha^{-1}]$, $S_2 = [0, \alpha^{-2}]$, and $S_3 = [0, \alpha^{-2} + \alpha^{-1}]$ are representative states with the contents of the bottom register, Fig. 2, of the syndrome former equal to 00, 01, 10, and 11, respectively. An input $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [0,1]$ brings us from state S_0 to state S_1. An input $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [1,0]$ brings us from state S_0 to state $S' = [\alpha^{-1}, 0]$, which is not a representative physical state. The representative state in the equivalence class of S' can be found through addition of the base state $S^b = [\alpha^{-1}, s_2{}^b(\alpha)]$, where $s_2{}^b(\alpha) = \alpha^{-2} + \alpha^{-1}$ from Fig. 2. Hence, $S' + S^b = [0, \alpha^{-2} + \alpha^{-1}]$, i.e., an input $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [1,0]$ brings us from state S_0 to state $S_3 = S' + S^b$. In the same fashion $[n_1(\alpha), n_2(\alpha)]_0{}^0 = [1,1]$ brings us from state S_0 to state S_2. All states in the state diagram of Fig. 3 have now been identified and we leave it to the reader to fill in the remaining edges. A solid edge in Fig. 3 indicates that the syndrome digit corresponding to the particular transition is 0, a dashed edge corresponds to a syndrome digit 1. The numbers next to the edges are the values n_1, n_2; ω. As has been explained in Section I, it is the coefficients $\cdots$, ω_{-1}, ω_0, ω_1, $\cdots$ of the power series $\omega(\alpha)$ of (5) that one is really interested in. It requires some explanation that the generic value ω of these coefficients can also be indicated next to the edges in Fig. 3.

Theorem 3: The value of the coefficient ω_k of the power series $\omega(\alpha) = \cdots + \omega_{-1}\alpha^{-1} + \omega_0 + \omega_1\alpha + \cdots$ defined by (5) is uniquely determined by the state $S(k)$ of the syndrome former at time $t = k$ and by the value of its input $[n_{1k}, n_{2k}]$, $k = \cdots, -1, 0, +1, \cdots$.

Proof: As the syndrome former is a time-invariant circuit, all one must prove is that ω_0 is uniquely determined by $S(0)$ and $[n_{10}, n_{20}]$. Equation (5) can be rewritten as

$$\omega_0 = [D_1'(\alpha)n_1(\alpha) + D_2'(\alpha)n_2(\alpha)]_0{}^0 + (D_{10}n_{10} + D_{20}n_{20}), \tag{6}$$

where the prime on $D_i'(\alpha)$, $i = 1, 2$, indicates that a possible constant term D_{i0} has been omitted. The second term $(D_{10}n_{10} + D_{20}n_{20})$ on the RHS of (6) is completely determined by the input $[n_{10},n_{20}]$ to the syndrome former at time $t = 0$. Now if the state $S(0)$ of the syndrome former at time $t = 0$ determines $[n_1(\alpha),n_2(\alpha)]_{-\nu}{}^{-1}$ and if deg $D_1'(\alpha) \leqslant \nu$ and deg $D_2'(\alpha) \leqslant \nu$, then $S(0)$ determines the value of the first term $[D_1'(\alpha)n_1(\alpha) + D_2'(\alpha)n_2(\alpha)]_0{}^0$ on the RHS of (6) and we are done. Now from Berlekamp [3, p. 27] we know that there exist polynomials $D_1(\alpha)$ and $D_2(\alpha)$ satisfying $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$ and thus that deg $D_1(\alpha) <$ deg $C_2(\alpha)$ and deg $D_2(\alpha) <$ deg $C_1(\alpha)$. Hence, the degrees of both $D_1'(\alpha)$ and $D_2'(\alpha)$ are less than the common degree ν of $C_1(\alpha)$ and $C_2(\alpha)$. However, the state $S(0)$ determines $[n_1(\alpha), n_2(\alpha)]_{-\nu}{}^{-1}$ only to within an equivalence class. It thus remains to be shown that addition of a zero-equivalent state $S = [s_1(\alpha),s_2(\alpha)]$ does not affect the value of $[D_1'(\alpha)n_1(\alpha) + D_2'(\alpha)n_2(\alpha)]_0{}^0$. For a zero-equivalent state we have by definition

$$[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)]_0{}^\infty = 0. \tag{7}$$

From $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$ it follows that $D_1'(\alpha)C_1(\alpha) + D_2'(\alpha)C_2(\alpha) = 0$. Thus we can define the polynomial $P(\alpha)$ by

$$P(\alpha) = D_1'(\alpha)C_1(\alpha) = D_2'(\alpha)C_2(\alpha). \tag{8}$$

It now follows that

$$P(\alpha)[D_1'(\alpha)s_1(\alpha) + D_2'(\alpha)s_2(\alpha)] = D_1'(\alpha)D_2'(\alpha)[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)]. \tag{9}$$

As deg $C_1(\alpha) >$ deg $D_2'(\alpha)$, we have deg $P(\alpha) >$ deg $D_1'(\alpha)D_2'(\alpha)$. Thus, if $[D_1'(\alpha)s_1(\alpha) + D_2'(\alpha)s_2(\alpha)]$ had a nonzero term of degree 0, then $[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)]$ would have a nonzero term of degree larger than 0. As, according to (7), this cannot be the case it follows that $[D_1'(\alpha)s_1(\alpha) + D_2'(\alpha)s_2(\alpha)]_0{}^0 = 0$. Q.E.D.

In fact, we even have the stronger result that all edges leading to the same state $S(k + 1)$, $k = \cdots, -1, 0, +1, \cdots$, have the same value of ω_k associated with them.

Corollary: The value of ω_k, $k = \cdots, -1, 0, +1, \cdots$, is uniquely determined by the value of $S(k + 1)$.

Proof: As the syndrome former is a time-invariant circuit, all one must prove is that ω_0 is uniquely determined by the state $S(1)$ of the syndrome former at time $t = 1$. According to (6), the value of ω_0 is uniquely determined by $[n_1(\alpha), n_2(\alpha)]_{-\nu+1}{}^0$ as the degree of both $D_1(\alpha)$ and $D_2(\alpha)$ is less than the common degree ν of $C_1(\alpha)$ and $C_2(\alpha)$. In fact, we only need to know $[n_1(\alpha),n_2(\alpha)]_{-\nu+1}{}^0$ to within an equivalence class. The proof is identical to the proof of Theorem 3, except that we redefine $P(\alpha)$ of (8), as

$$Q(\alpha) = D_1(\alpha)C_1'(\alpha) + D_2(\alpha)C_2'(\alpha). \tag{10}$$

As $S(1)$, in fact, defines $[n_1(\alpha),n_2(\alpha)]_{-\nu+1}{}^0$ to within such an equivalence class, $S(1)$ uniquely determines ω_0. Q.E.D.

III. ALGORITHM

As the recursive algorithm to be described in this section is similar to Viterbi's [2], we can be very brief. For reasons of clarity the decoding algorithm will be explained using the code generated by the encoder of Fig. 1 as an example. Fig. 4 represents the kth section, $k = \cdots, -1, 0, +1, \cdots$, of the trellis diagram corresponding to the state diagram of Fig. 3. The decoding algorithm is to find the coefficients ω_k of the power series $\omega(\alpha) = \cdots + \omega_{-1}\alpha^{-1} + \omega_0 + \omega_1\alpha + \cdots$ associated with the path of minimum weight through the trellis diagram. The pertinent weight is the Hamming weight of the pair $[\hat{n}_1(\alpha), \hat{n}_2(\alpha)]$ associated with the particular path. As in the Viterbi algorithm [2], to find the minimum weight path we associate a metric with each possible state. The metrics at time $t = k$ can be computed recursively given the metrics at time $t = k - 1$. For the trellis diagram of Fig. 4, the recursion is given by

$$M_0(k + 1) = \bar{z}_k \min [M_0(k), M_1(k) + 2] + z_k \min [M_2(k), M_3(k) + 2] \tag{11a}$$

$$M_1(k + 1) = \bar{z}_k \min [M_2(k) + 1, M_3(k) + 1] + z_k \min [M_0(k) + 1, M_1(k) + 1] \tag{11b}$$

$$M_2(k + 1) = \bar{z}_k \min [M_0(k) + 2, M_1(k)] + z_k \min [M_2(k) + 2, M_3(k)] \tag{11c}$$

$$M_3(k + 1) = \bar{z}_k \min [M_2(k) + 1, M_3(k) + 1] + z_k \min [M_0(k) + 1, M_1(k) + 1], \tag{11d}$$

where $\bar{z}_k$ is the modulo-2 complement of z_k, $k = \cdots, -1, 0, +1, \cdots$. Note that for each value $z_k = 0$ or $z_k = 1$ two arrows impinge on each $(k + 1)$-state. The arrow associated with the minimum within the relevant pair of square brackets in (11) is called the "survivor." If both arrows have this property, flip a coin to determine the survivor. In the classical Viterbi [2] implementation of the algorithm each state S_j, $j = 0, 1, 2, 3$, has a metric register MR_j and a path register PR_j associated with it. The metric register is used to store the current metric value $M_j(k + 1)$ at time $t = k$, $k = \cdots, -1, 0, +1, \cdots$, associated with state S_j, $j = 0, 1, 2, 3$. The path register $\mathrm{PR}_j[0 : D - 1]$ stores the ω-values associated with the current sequence of the D most recent survivors leading up to state S_j at time $t = k$. The pertinent output is

$$\hat{\omega}_{k-D} = \text{CONTENTS } \mathrm{PR}_{j(k)}[D - 1 : D - 1], \tag{12a}$$

where $j(k)$ minimizes $M_j(k + 1)$, i.e.,

$$M_{j(k)}(k + 1) = \min_j M_j(k + 1). \tag{12b}$$

If more than one j satisfies (12b), select $j(k)$ arbitrarily among the candidates. As the algorithm returns $\hat{\omega}_{k-D}$ at time $t = k$,

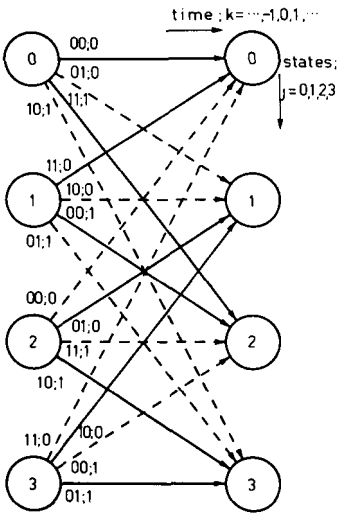

Fig. 4. kth section of the trellis diagram, $k = \cdots, -1, 0, +1, \cdots$.

$k = \cdots, -1, 0, +1, \cdots$, the path register length D is also referred to as the coding delay. The resulting bit error probability P_b decreases as the coding delay D increases. Increasing D beyond $5(\nu + 1)$ does not lead to appreciable further decrease in the value of P_b. This relation between the bit error probability P_b and the coding delay D will be elaborated on further in Section VI. The next section is concerned with a practical implementation of the syndrome decoder.

IV. IMPLEMENTATION

So far, the syndrome decoder has only been of theoretical interest as a possible alternative for the classical Viterbi decoder [2]. We will now study a practical implementation and in the next section make some comparisons as to the relative hardware complexity of these competing decoders.

Using (11) we construct Table I. The first column just numbers the rows of the table. The second column lists all possible metric combinations $M_0(k), M_1(k), M_2(k), M_3(k)$ at time $k-1$. As only the differences between the metrics of a quadruple matter, we substract from each member of a quadruple of metrics the minimum value of the quadruple, i.e., all quadruples of metrics in Table I have one or more zeros. Columns 3 and 4 apply to the case that $z_k = 0$ and columns 5 and 6 to the case that $z_k = 1$. Columns 3 and 5 list the survivors, i.e., the indices of the associated $(k-1)$ states, and columns 4 and 6 the new metrics $M_0(k+1)$, $M_1(k+1)$, $M_2(k+1)$, $M_3(k+1)$ as given by (11). If there is a choice of survivors, the candidates are placed within parentheses in the survivor columns.

Table I contains more information than is necessary for the actual implementation of the syndrome decoder. As explained in Section III, knowledge of the survivor leading to each state, together with the index j_m of the minimum within each new quadruple of metrics, suffices to determine the key sequence $\omega(\alpha)$ of (5). Hence, we omit the quadruples of metrics from Table I and store the resulting Table II in a read-only memory (ROM). The operation of the core part of the syndrome decoder can now be explained using the block diagram of

TABLE I
METRIC TRANSITIONS

Row Number	Old Metrics	$z_k = 0$ Survivors	$z_k = 0$ New Metrics	$z_k = 1$ Survivors	$z_k = 1$ New Metrics
0	0000	0(2,3) 1 (2,3)	0101	2 (0,1) 3 (0,1)	0101
1	0101	0 2 1 2	0111	2 0 3 0	0111
2	0111	0(2,3) 1 (2,3)	0212	2 0 3 0	0000
3	0212	0 2 (0,1) 2	0222	2 0 3 0	0010
4	0222	0(2,3) (0,1) (2,3)	0323	2 0 3 0	1010
5	0010	0 3 1 3	0101	2 (0,1) 3 (0,1)	1101
6	0323	0 2 0 2	0323	2 0 3 0	1020
7	1010	0 3 1 3	1101	2 1 3 1	1101
8	1101	0 2 1 2	0000	2 (0,1) 3 (0,1)	0212
9	1020	0 3 1 3	1101	(2,3) 1 3 1	2101
10	2101	0 2 1 2	1000	2 1 3 1	0212
11	1000	0(2,3) 1 (2,3)	1101	2 1 3 1	0101

Fig. 5. Assume that at time k the ROM address register AR contains (AR) = 7 and the ROM data register DR contains (DR) = (ROM,7). Note, see Fig. 4 and also the corollary to Theorem 3, that the ω-values to be shifted into $PR_0[0:0]$, $PR_1[0:0]$, $PR_2[0:0]$, $PR_3[0:0]$ are 0011, respectively. Let $z_k = 1$. Then according to row 7 and column 5 of Table II, or according to the contents (DR) of the DR, replace

$$PR_0[1:D-1] \leftarrow \text{CONTENTS } PR_2[1:D-1]$$

$$PR_1[1:D-1] \leftarrow \text{CONTENTS } PR_1[1:D-1]$$

$$PR_2[1:D-1] \leftarrow \text{CONTENTS } PR_3[1:D-1]$$

$$PR_3[1:D-1] \leftarrow \text{CONTENTS } PR_1[1:D-1].$$

The rightmost digit, $PR_0[D-1:D-1]$, $PR_1[D-1:D-1]$, $PR_2[D-1:D-1]$, $PR_3[D-1:D-1]$, of all four path registers is fed to the selector, see Fig. 5, that determines $\hat{\omega}_{k-D}$ according to (12a) using as $j(k)$ the entry in row 7 and column 7, i.e., $j_m = 2$, of Table II which can also be found in the DR. To complete the kth cycle of the syndrome decoder, set (AR) = 8 and read DR $\leftarrow$ (ROM,8). The ROM decoder for the code of Fig. 1 has been realized in hardware using path registers of length $D = 11$. The experimental results will be discussed in Section VI.

V. PATH REGISTER SAVINGS

The ROM-implementation of the syndrome decoder as described in Section IV has been realized for the codes listed in column 2 of Table III. We will discuss some interesting aspects of this table. The first row lists several properties of code 1 that was used as an example throughout the earlier part of the paper. Column 3 lists the number of metric combinations of the various codes. The classical Viterbi decoder [2] can also be realized using a ROM in the manner described in Section IV. However, the Viterbi decoder for code 1 has 31 metric combinations, whereas the syndrome decoder has only 12 metric combinations. For the $\nu = 4$ codes this difference is even more pronounced. For codes 2 and 3 the syndrome decoder has, respectively, 1686 and 1817 metric combinations, whereas the classical Viterbi decoder for either of these codes has more than 15 000 metric combinations. Note that in

TABLE II
CONTENTS OF THE ROM

Old ROM Address	$z_k = 0$ Survivors	New ROM Address	Index j_m	$z_k = 1$ Survivors	New ROM Address	Index j_m
0	0(2,3) 1 (2,3)	1	(0,2)	2 (0,1)3(0,1)	1	(0,2)
1	0 2 1 2	2	0	2 0 3 0	2	0
2	0(2,3) 1 (2,3)	3	0	2 0 3 0	0	(0,1,2,3)
3	0 2 (0,1) 2	4	0	2 0 3 0	5	(0,1,3)
4	0(2,3)(0,1)(2,3)	6	0	2 0 3 0	7	(1,3)
5	0 3 1 3	1	(0,2)	2 (0,1)3(0,1)	8	2
6	0 2 0 2	6	0	2 0 3 0	9	(1,3)
7	0 3 1 3	8	2	2 1 3 1	8	2
8	0 2 1 2	0	(0,1,2,3)	2 (0,1)3(0,1)	3	0
9	0 3 1 3	8	2	(2,3) 1 3 1	10	2
10	0 2 1 2	11	(1,2,3)	2 1 3 1	3	0
11	0(2,3) 1 (2,3)	8	2	2 1 3 1	1	(0,2)

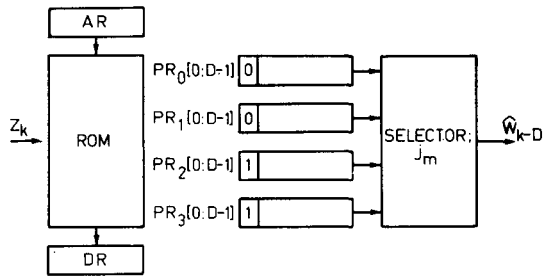

Fig. 5. Block diagram of the core of the syndrome decoder.

TABLE III
CODES REALIZED

code	connection polynomials; 0-th order right	number of metric combinations	minimum number of path registers	distance	number of paths at given distance	total number of associated bit errors
1	101	12	3	5	1	1
	111			6	2	4
2	10011	1,686	12	7	2	4
	11011			8	4	12
3	10011	1,817	9	7	3	7
	10111			8	3	10
4	10011	11,304	12	7	2	4
	11101			8	4	12

columns 5, 6, and 7 both error events at the free distance and error events at the free distance plus one are considered. In a binary comparison with the no-error sequence, an error event at distance $2k$ has the same probability of occurrence [4] as an error event at distance $2k - 1$, $k = 1, 2, \cdots$. Thus, in the case that the free distance is odd, error events at the free distance plus one should also be considered when comparing codes as to the bit error probability P_b. Studying columns 5, 6, and 7 of Table III, we observe that as far as the bit error probability is concerned code 4 is indistinguishable from codes 2 and 3. However, the syndrome decoder for code 4 requires 11 304 ROM-locations and code 4 is thus, from a complexity point of view, inferior to codes 2 and 3.

The number of metric combinations increases rapidly with the constraint length ν of the code. Hence, for larger values of ν the size of the ROM in the implementation according to Section IV soon becomes prohibitive. In the classical implementation [2] with a metric register and a path register for each state S_j, $j = 0, 1, \cdots, 2^\nu - 1$, the Viterbi decoder and the syndrome decoder require roughly the same amount of hardware per state. However, comparing (11b) and (11d) for code 1, one observes that S_1 and S_3 have the same metric value. Moreover, selecting the identical survivor in case of a tie, S_1 and S_3 also have the same path register contents. As far as metric and path register contents are concerned, the states S_1 and S_3 are not distinct. The metric register and the path register of either state S_1 or state S_3 can be eliminated. Thus, in the classical implementation with metric registers and path registers, the syndrome decoder for code 1 requires only $\frac{3}{4}$ of the amount of hardware that the Viterbi decoder requires. We will prove that, in general, one can eliminate the metric and path registers of half the odd numbered states, where the state number of a representative state $S = [0, s_2(\alpha)]$ is the value of the contents of the bottom register of the syndrome former interpreted as a binary number, i.e., odd states have $s_{2,-1} = 1$. Hence, the syndrome decoder is at most $\frac{3}{4}$ as complex as

is the Viterbi decoder. Looking again at Table III column 4 we see that code 1 can be realized with 3 instead of $2^\nu = 4$ path registers, and that codes 2 and 4 can be realized with 12 instead of $2^\nu = 16$ path registers. Code 3 requires even fewer, i.e., nine instead of $2^\nu = 16$ path registers. We now prove that the syndrome decoder is, in general, at most $\frac{3}{4}$ as complex as is the Viterbi decoder.

Theorem 4: The odd numbered (representative) states $S(1) = [0,s_2(\alpha)]$ and $S'(1) = [0,s_2'(\alpha)]$, where $s_2'(\alpha) = s_2(\alpha) + s_2{}^b(\alpha) + \alpha^{-1}$ have identical metric equations, compare (11b) and (11d), iff $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$.

Proof: An odd numbered state $S(1) = [0,s_2(\alpha)]$ has $s_{2,-1} = 1$. If both connection polynomials $C_1(\alpha)$ and $C_2(\alpha)$ have a nonzero term of degree ν, it follows from the construction of the base state S^b that $s_{2,-1}{}^b = 1$. Hence, $s_2'(\alpha) = s_2(\alpha) + s_2{}^b(\alpha) + \alpha^{-1}$ has $s_{2,-1}{}' = 1$, and the requirement that both $S(1)$ and $S'(1)$ are odd numbered states is consistent. Consider the following parent states:

$$S_a(0) = [0,\alpha s_2(\alpha)]_{-\nu}{}^{-1}$$

$$S_b(0) = [0,\alpha^{-\nu} + \alpha s_2(\alpha)]_{-\nu}{}^{-1}$$

$$S_c(0) = [0,\alpha s_2'(\alpha)]_{-\nu}{}^{-1}$$

$$S_d(0) = [0,\alpha^{-\nu} + \alpha s_2'(\alpha)]_{-\nu}{}^{-1}.$$

As both $C_1(\alpha)$ and $C_2(\alpha)$ have a nonzero term of degree ν, $S_a(0)$ and $S_b(0)$ give rise to complementary syndrome digits, and so do $S_c(0)$ and $S_d(0)$. For an input $[n_{10},n_{20}] = [0,1]$ the parent states $S_a(0)$ and $S_b(0)$ go into $S(1)$ and the parent states $S_c(0)$ and $S_d(0)$ go into $S'(1)$, and vice versa for an input $[n_{10},n_{20}] = [1,0]$. Assuming that $C_1(\alpha)$ and $C_2(\alpha)$ both have a nonzero constant term and that $[n_{10},n_{20}]$ is either $[0,1]$ or $[1,0]$, the syndrome value only depends on the parent state $S(0)$. Hence, $S(1)$ and $S'(1)$ have identical equations. Q.E.D.

Theorem 4 proves that the syndrome decoders for the $\nu = 4$ codes 2, 3, and 4 in Table III can be realized with no more than 12 instead of $2^\nu = 16$ path registers. Column 4 of Table III shows, however, that code 3 requires only 9 instead of $2^\nu = 16$ path registers. The following theorem shows how this further reduction in hardware can be accomplished.

Theorem 5: One 4-tuple of pairs of odd numbered parent states gives rise to two pairs of odd numbered states that have identical metric equations iff $C_{1,1} = C_{2,1}$ and $C_{1,\nu-1} = C_{2,\nu-1}$.

Proof: Assume that $S_a(0)$, $S_b(0)$, $S_c(0)$, and $S_d(0)$ in the proof of Theorem 4 are odd numbered states, i.e., $s_{2,-2} = s_{2,-2}{}' = 1$. This can be a consistent requirement if $s_{2,-2}{}^b = 0$, and from the construction of the base state S^b it is clear that $s_{2,-2}{}^b = 0$ iff $C_{1,\nu-1} = C_{2,\nu-1}$. If $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ are odd numbered states, then according to Theorem 4 there exists a corresponding 4-tuple of odd numbered states $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_d'(0)$ such that the corresponding components of these 4-tuples have identical metrics. According to Theorem 4, the first 4-tuple $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ gives rise to the states $S_p(1)$ and $S_q(1)$ that have identical metric equations. Similarly, according to Theorem 4 the second 4-tuple $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_d'(0)$ gives rise to the states $S_p'(1)$ and $S_q'(1)$ that also have identical metric equations. As the corresponding members of the parent 4-tuples $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ and $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_d'(0)$ have identical metrics, the four states $S_p(1)$, $S_q(1)$, $S_p'(1)$, and $S_q'(1)$ have identical metric equations iff corresponding states in the parent 4-tuples give rise to the same syndrome digit. For this to occur it is necessary that the difference $S = [0,s_2{}^b(\alpha) + \alpha^{-1}]$ between corresponding states is a zero-equivalent state. But the base state $S^b = [\alpha^{-1},s_2{}^b(\alpha)]$ is a zero-equivalent state, hence, $S = [\alpha^{-1},s_2{}^b(\alpha)] + [\alpha^{-1},\alpha^{-1}]$ is a zero-equivalent state iff $C_{1,1} = C_{2,1}$. Q.E.D.

It is easy to verify that the two 4-tuples $S_a(0)$, $S_b(0)$, $S_c(0)$, $S_d(0)$ and $S_a'(0)$, $S_b'(0)$, $S_c'(0)$, $S_d'(0)$ in Theorem 5 with $[n_{10},n_{20}]$ equal to either $[0,0]$ or $[1,1]$ lead to four even numbered states that have pairswise-identical metric equations.

Corollary: The 4-tuple of pairs of odd numbered parent states of Theorem 5 gives rise to 4 even numbered states that have pairswise identical metric equations.

Summarizing, we have the following results. According to Theorem 4, the odd numbered states have pairswise-identical metric equations for codes, such as codes 1, 2, and 4 of Table III, with $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$. Hence, for such codes $2^{\nu-2}$, $\nu = 2, 3, \cdots$, metric and path register combinations can be eliminated. This leads to a syndrome decoder of $\frac{3}{4}$ of the hardware complexity of the Viterbi decoder. According to Theorem 5, each 4-tuple of pairs of odd numbered states leads to two pairs of odd numbered states that have identical metric equations for codes, such as code 3 of Table III, with $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$ and $C_{1,1} = C_{2,1}{}'C_{1,\nu-1} = C_{2,\nu-1}$. Hence, for such codes an additional $2^{\nu-4}$, $\nu = 4, 5, \cdots$, metric and path register combinations can be eliminated. According to the corollary to Theorem 5, the 4-tuple of pairs of odd numbered states mentioned above also leads to four even numbered states that have pairswise-identical metric equations. This leads to an additional saving of $2.2^{\nu-4}$ metric and path register combinations. The total savings for codes with $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$ and $C_{1,1} = C_{2,1}{}'C_{1,\nu-1} = C_{2,\nu-1}$ is thus equal to $2^{\nu-2} + 3.2^{\nu-4}$, $\nu = 4, 5, \cdots$. The resulting syndrome decoder has 9/16 of the hardware complexity of the Viterbi decoder. Continuing this series of reductions the ultimate savings in metric and path register combinations is equal to $2^{\nu-2} + 3.2^{\nu-4} + 3^2.2^{\nu-6} + \cdots$ [9], leading to a syndrome decoder of hardware complexity equal to $(\frac{1}{2}\sqrt{3})^\nu$ times the hardware complexity of the Viterbi decoder. Note that in order to achieve this ultimate savings in hardware complexity one must put severe constraints on the encoder. Code 3 of Table III still achieves the maximum free distance for constraint length $\nu = 4$ codes. It is quite conceivable, however that in putting on further constraints on the encoder for larger values of ν it is no longer possible to achieve the maximum free distance. However, by only requiring $C_{1,0} = C_{2,0} = C_{1,\nu} = C_{2,\nu} = 1$, and $C_{1,1} = C_{2,1}{}'C_{1,\nu-1} = C_{2,\nu-1}$ one already achieves the reduction of 7/16, as shown above.

One final comment is in order. The hardware reduction with respect to the Viterbi decoder has been equated with the savings in metric and path register combinations. Note that the

path registers in the snydrome decoder are used to store the binary ω-values, compare (5), i.e., they are binary storage registers just as are the path registers of the Viterbi decoder. One might remark that the path registers of the syndrome decoder are more complex because each state has four possible parent states instead of two as in the Viterbi decoder. However, by filling the path registers serially this aspect hardly adds to the complexity.

VI. EXPERIMENTAL RESULTS

The solid lines in Fig. 6 give the measured bit error probability P_b of code 3 of Table III as a function of the transition probability p of the binary symmetric channel (BSC), for both a path register length $D = 11$ and a path register length $D = 16$. The dashed line in Fig. 6 is a refinement of Van De Meeberg's [4] of Viterbi's upper bound on the bit error probability. This dashed bound is valid for infinite path register length. We extended Van De Meeberg's upper bound to also apply to finite path register lengths. The derivation of this extended bound will be published shortly. The dashed curves in Fig. 6 give the upper bound on the bit error probability for both $D = 11$ and $D = 16$. It is clear from Fig. 6 that it does not pay to increase the path register length much beyond $D = 16$.

It appears, so far, that the syndrome decoder is an interesting (from the hardware point of view) substitute for the classical Viterbi decoder. In closing, we want to mention two important applications of the syndrome decoder where the classical Viterbi decoder cannot be used. These applications are in feedback communications [5], and in source coding (data reduction) [6]. In the next two paragraphs we describe these uses, both of which have been simulated on the computer, of the syndrome decoder, respectively.

Reference [5] describes a coding strategy for duplex channels that enables one to transfer the hardware or the program complexity from the passive (receiving) side to the active (transmitting) side of the duplex channel. As pointed out in reference [5], this coding strategy can be used to great advantage in a computer network with a star configuration. For the information flow from the central computer to the satellites one uses the duplex strategy thus only requiring one complex one-way decoder at the central facility. For the information flow from a satellite computer towards the central facility one uses one-way coding, again using the complex one-way decoder at the central computer. One thus saves a number of complex one-way decoders equal to the number of satellite computers in the multiple dialog system (MDS). The duplex strategy [5] requires at the active (transmitting) side of the duplex channel an estimate of the forward noise $n_1(\alpha)$. To form this estimate $\hat{n}_1(\alpha)$, the data received at the passive station are scrambled by a convolutional scrambler $C(\alpha)$ and sent back to the active station. At the active station one can now form the estimate $\hat{n}_1(\alpha)$ using the Viterbi decoder for the "systematic" convolutional code generated by an encoder with connection polynomials $C_1(\alpha) = 1$, $C_2(\alpha) = C(\alpha)$. It is well known that "nonsystematic" convolutional codes are more powerful than systematic convolutional codes. With our syndrome decoder we are now able to find $\hat{n}_1(\alpha)$ according to a nonsystematic code. To this end we modify the feedback-free scrambler $C(\alpha)$ [5] into a feedback scrambler $C_2(\alpha)/C_1(\alpha)$, see Fig. 7. Note that $z(\alpha)$ according to Fig. 7 is identical to (1). Hence, we can use our syndrome decoder to obtain $\hat{n}_1(\alpha)$. Fig. 8 gives the feedback scrambler for the convolutional code generated by the encoder of Fig. 1.

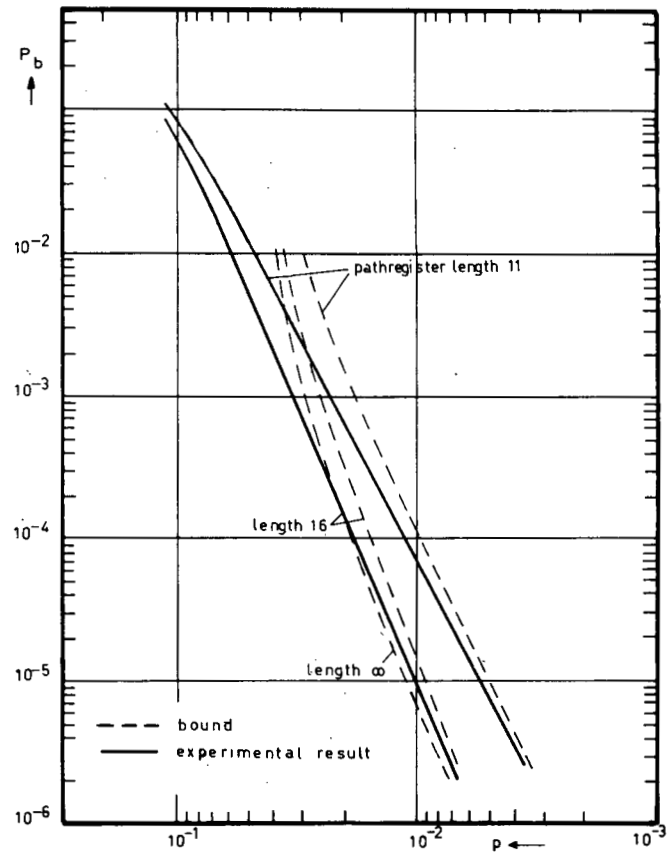

Fig. 6. Bit error rate P_b versus channel transition probability p.

Note that the syndrome former, Fig. 2, has two input sequences $n_1(\alpha)$, $n_2(\alpha)$ and one output sequence $z(\alpha) = C_2(\alpha)n_1(\alpha) + C_1(\alpha)n_2(\alpha)$. Thus, the syndrome former compresses two binary streams $n_1(\alpha)$, $n_2(\alpha)$ into one stream $z(\alpha)$ and, hence, achieves a data compression of a factor of 2. The estimator part of the syndrome decoder can with high probability of being correct recover the original sequences $n_1(\alpha)$, $n_2(\alpha)$ given the compressed data sequence $z(\alpha)$. The use of a syndrome decoder for data compression has also been studied by Massey [6]. In general, to obtain a data compression factor n, $n = 2, 3, \cdots$, one used the syndrome decoder of a rate $-(n-1)/n$ convolutional code.

VII. CONCLUSIONS

This paper considers the syndrome decoding of rate $-\frac{1}{2}$ convolutional codes. Table III shows that the number of metric combinations of the syndrome decoder is small compared to the number of metric combinations of the corresponding Viterbi decoder. For the constraint length $\nu = 4$ code of row 3 of Table III, for example, the number of metric combinations with syndrome decoding is 1817, whereas the Viterbi decoder for this same code has over 15 000 metric

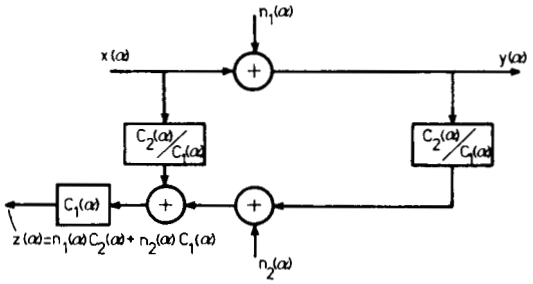

Fig. 7. Duplex channel with feedback scramblers.

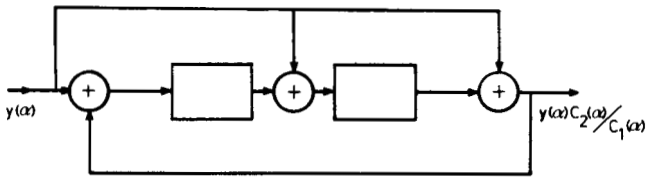

Fig. 8. Feedback scrambler with $C_1(\alpha) = 1 + \alpha^2$, $C_2(\alpha) = 1 + \alpha + \alpha^2$.

combinations. This relatively small number of metric combinations for small constraint length codes enables the ROM-implementation of Section IV, that eliminates the need for metric registers. For larger constraint lengths, the storage requirements of the ROM would become excessive. However, by putting mild constraints on the encoder it is possible to eliminate more than half of the metric and path register combinations. The syndrome decoder of code 3 of Table III, for example, only requires nine path registers, whereas the corresponding Viterbi decoder has $2^\nu = 16$ path registers.

The idea of syndrome decoding can be extended to rate $-k/n$ convolutional codes. Forney [7], [8] describes the mathematical tools necessary to find the general syndrome former equations and the equations of the inverse encoder.

Note added in proof: A. W. J. Kolen has pointed out a mistake in the proof of Theorem 3, i.e., $D_1'(\alpha)C_1(\alpha) + D_2'(\alpha)C_2(\alpha) = 0$ is incorrect. The proof can be corrected by observing that deg $[C_2(\alpha)s_1(\alpha) + C_1(\alpha)s_2(\alpha)] >$ deg $[D_1(\alpha)s_1(\alpha) + D_2(\alpha)s_2(\alpha)]$.

ACKNOWLEDGMENT

The authors want to thank L. J. A. E. Rust for his encouragement and help with the computer simulations.

REFERENCES

[1] J. P. M. Schalkwijk and A. J. Vinck, "Syndrome decoding of convolutional codes," *IEEE Trans. Commun.* (Corresp.), vol. COM-23, pp. 789–792, July 1975.

[2] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol. (Special Issue on Error Correcting Codes–Part II)*, vol. COM-19, pp. 751–772, Oct. 1971.

[3] E. R. Berlekamp, *Algebraic Coding Theory.* New York: McGraw-Hill, 1968.

[4] L. Van De Meeberg, "A tightened upper bound on the error probability of binary convolutional codes with Viterbi decoding," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 389–391, May 1974.

[5] J. P. M. Schalkwijk, "A coding scheme for duplex channels," *IEEE Trans. Commun. (Special Issue on Communications in Europe),* vol. COM-22, pp. 1369–1374, Sept. 1974.

[6] J. L. Massey, "The codeword and syndrome methods for data compression with error-correcting codes," in *Proc. NATO Advanced Study Institute on New Directions in Signal Processing in Communications and Control,* Darlington, England, Aug. 5–17, 1974.

[7] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory,* vol. IT-16, pp. 720–738, November 1970; also, correction appears in vol. IT-17, p. 360, May 1971.

[8] —, "Structural analysis of convolutional codes via dual codes," *IEEE Trans. Inform. Theory,* vol. IT-19, pp. 512–518, July 1973.

[9] J. P. M. Schalkwijk, "Symmetries of the state diagram of the syndrome former of a binary rate-½ convolutional code," Lecture Notes, CISM Udine Summer School on Coding, Udine, Italy, Sept. 2–12, 1975.

★

J. Pieter M. Schalkwijk (M'66) was born in Rijswijk, The Netherlands, on November 1, 1936. He received the M.S. degree in electrical engineering from the Technological University, Delft, The Netherlands, in 1959, and the Ph.D. degree, also in electrical engineering, from Stanford University, Stanford, CA, in 1965.

From 1959 to 1961, while in military service in The Netherlands, he was assigned to work at Philips, Hengelo, The Netherlands, on the control problems that arise when using a digital computer as part of a radar fire-control system. In 1961 he joined the National Defense Research Laboratory of The Netherlands, where he carried out comparative tests of HF digital-data terminals. In 1963 he worked on windtunnel measurements at the National Aeronautics and Space Research Laboratories in Amsterdam, The Netherlands. In 1963 he came to the U.S. to continue his advanced study. In 1965 he joined the Applied Research Laboratory of Sylvania Electronic Systems, Waltham, MA, where he worked on problems of signal design and reception for digital-data transmission. From 1968 to 1972 he was an Assistant Professor of Information and Computer Science at the University of California at San Diego. Since 1972 he has been a Professor of Communication Theory at the Technological University, Eindhoven, The Netherlands.

Dr. Schalkwijk is a member of The Netherlands Electronics and Radio Society, The Royal Netherlands Institute of Engineers, and the Association for Computing Machinery.

★

A. J. Vinck was born in Breda, The Netherlands, on May 15, 1949. He received the M.S. degree in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 1974.

He is currently working at the Department of Electrical Engineering, Eindhoven University of Technology, where he is involved in research on convolutional codes.