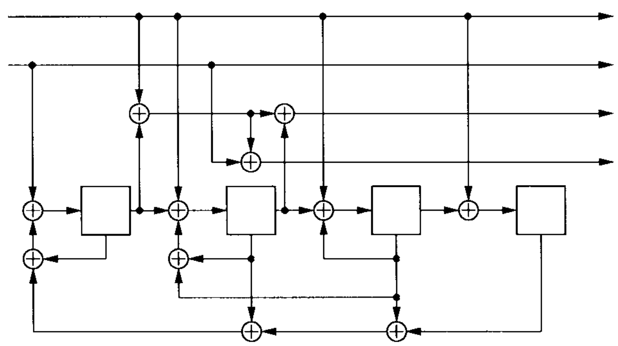

Fig. 3. A minimal realization of the systematic rate $R = 2/4$ binary encoder.

convolutional encoder has $d_{\mathrm{free}} = 8$ which meets the Griesmer bound.

Furthermore, in [12] Benedetto *et al.* reported a 16-state $R = 1/2$ convolutional code over $\mathbb{Z}_4$ with generator

$$G(D) = (1 + D + D^2 \quad 2 + 3D + 2D^2) \tag{8}$$

that achieves $d^2_{\mathrm{E,\,free}} = 16$. This code has $n(12) = 289$ and, hence, is slightly inferior to (3).

Recently, Calderbank *et al.* [13] used "unwrapping" of their tail-biting representation of the $(24, 12, 8)$ extended Golay code to construct a most interesting 16-state convolutional code with $d_{\mathrm{free}} = 8$. Their GCC (Golay convolutional code) can be encoded by a rate $R = 4/8$ time-invariant convolutional encoder or with a rate $R = 1/2$ time-varying, period 4 convolutional encoder; see also [14].

ACKNOWLEDGMENT

We are grateful to R. Garello and his colleagues for drawing our attention to [10]–[12].

REFERENCES

[1] J. A. Heller, "Short constraint length convolutional codes," Jet Propulsion Lab., California Inst. Technol., Pasadena, *Space Programs Summary 37-54*, vol. 3, pp. 171–177, 1968.
[2] J. H. Griesmer, "A bound for error-correcting codes," *IBM J. Res. Develop.*, vol. 4, pp. 532–542, 1960.
[3] G. D. Forney, Jr., N. J. A. Sloane, and M. D. Trott, "The Nordstrom–Robinson Code is the binary image of the octacode," *DIMACS, Ser. Discrete Math. Theor. Comput. Sci.*, vol. 14, pp. 19–26, 1993.
[4] A. R. Hammons, Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, "The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 301–319, Jan. 1994.
[5] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.
[6] R. Johannesson, Z.-X. Wan, and E. Wittenmark, "Some structural properties of convolutional codes over rings," *IEEE Trans. Inform. Theory*, vol. 44, pp. 839–845, Mar. 1998.
[7] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.
[8] L.-N. Lee, "Short unit-memory byte-oriented binary convolutional codes having maximal free distance," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 349–352, May 1976.
[9] R. Garello, private communication, Dec. 1, 1997.
[10] S. Benedetto, R. Garello, M. Mondin, and G. Montorsi, "Geometrically uniform partitions of $L \times$ MPSK constellations and related binary codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1773–1798, Nov. 1993.
[11] S. Benedetto, R. Garello, and M. Mondin, "Group trellis codes over multidimensional unbalanced PSK and their optimality," *IEEE Trans. Commun.*, vol. 43, pp. 2922–2932, Dec. 1995.
[12] S. Benedetto, R. Garello, M. Mondin, and G. Montorsi, "Geometrically uniform TCM codes over groups based on $L \times$ MPSK constellations," *IEEE Trans. Inform. Theory*, vol. 40, pp. 137–152, Jan. 1994.
[13] A. R. Calderbank, G. D. Forney, Jr., and A. Vardy, "Minimal tail-biting trellises: The Golay code and more," *IEEE Trans. Inform. Theory*, submitted for publication.
[14] ——, "Classification of the 16-state tail-biting representations of the binary Golay code," in *Codes, Curves, and Signals: Common Threads in Communications*, A. Vardy, Ed. Boston, MA: Kluwer, 1998.

Convolutional Encoder State Estimation

A. J. Han Vinck, *Senior Member, IEEE*,
Petr Dolezal, and Young-Gil Kim

***Abstract*—To estimate the convolutional encoder state from received data, one may use the inverse to the encoder G. However, channel errors make this method unreliable. We propose a method that uses the received data in the following way. We calculate the syndrome, and after a specific number of received syndrome values equal to zero, we expect that the corresponding received data is also error-free. The received data is then used to build the inverse and give an estimate for the encoder state. The method can be used in situations where knowledge of the encoder state helps the decoding process or for synchronization purposes. We analyze the performance of the described method with respect to state estimation error probability and the average time it takes before we can estimate the encoder state with a certain desired reliability.**

***Index Terms*—Convolutional codes, state recovery.**

I. Introduction

A general convolutional encoder is specified by its $k \times n$ generator matrix G. For minimal encoders, we can derive the delay-free right inverse G^{-1} and the syndrome former H^T, see [1]. We concentrate on rate $R = 1/2$, or $k = 1$ and $n = 2$ convolutional encoders, with a standard constraint length 6 encoder as a working example. Nonsystematic, noncatastrophic encoders are described by the pair of binary polynomials (g_1, g_2). Generally, one assumes that both polynomials have zero delay and maximum degree or constraint length m. The inverse consists of two polynomials, d_1 and d_2 such that $g_1 d_1 + g_2 d_2 = 1$, where all operations are modulo 2. The realization of the inverse is called invertor. The syndrome former is the pair $(g_2, g_1)^T$. Using the delay operator D notation, the encoder input sequence, the encoder output sequence pair and the received channel output sequence pair are given by $X(D)$, $(C_1(D), C_2(D)) = X(D) * G$, and $(R_1(D), R_2(D))$, respectively. The received sequence

$$(R_1(D), R_2(D)) = (C_1(D) + n_1(D), C_2(D) + n_2(D))$$

where $n_i(D)$, $i = 1, 2$ is the channel error sequence that results from a hard-decision detection and a $+$ denotes modulo two addition. Note

Manuscript received February 22, 1997; revised December 2, 1997. This work was sponsored by the VW foundation.
A. J. Han Vinck and P. Dolezal are with the Institute for Experimental Mathematics, University of Essen, 45326, Essen, Germany.
Y.-G. Kim is with KAIST, Taejon, South Korea.
Publisher Item Identifier S 0018-9448(98)03477-4.

that by linearity

$$\begin{aligned}(R_1(D),\, R_2(D)) * H^T &= (C_1(D) + n_1(D),\, C_2(D) + n_2(D)) * H^T \\ &= (n_1(D),\, n_2(D)) * H^T \\ &= Z(D) \end{aligned} \tag{1}$$

and

$$\begin{aligned}(R_1(D),\, R_2(D)) * G^{-1} &= (C_1(D) + n_1(D),\, C_2(D) + n_2(D)) * G^{-1} \\ &= [C_1(D),\, C_2(D)] * G^{-1} + (n_1(D),\, n_2(D)) * G^{-1} \\ &= X(D) + (n_1(D),\, n_2(D)) * G^{-1} \\ &= X'(D). \end{aligned} \tag{2}$$

The encoder state at time t, S_t, is defined as the content of the encoder shift register realization and follows directly from $X(D)$. For a constraint length m, $R = 1/2$ encoder in the obvious realization, the state

$$S_t = (X_{t-1},\, X_{t-2},\, \cdots,\, X_{t-m}).$$

The input X_t and S_t together determine the state S_{t+1}. Hence, if the channel is noiseless for a certain period, one can use the invertor output to reconstruct the encoder state. Forney [1] indicated already this possibility in his fundamental paper on algebraic structure of convolutional codes. However, we consider the problem how to construct an implementation of a state estimator that uses the inverse with high reliability when the channel is noisy. We give a system description in Section II. We introduce a certain delay, or so-called time to decision (TTD) before we give an estimate output. The delay TTD will be shown to be the key to a reliable estimate. In Section III we consider the encoder estimation error probability depending on the TTD. Of course, there is a tradeoff between reliability and delay. Section IV gives simulation results that confirm the analysis.

II. System Description

We describe a "state-tracking" method for $R = 1/2$ convolutional codes. The principle can be extended to general $R = k/n$ codes. The state estimator to be used is given in Fig. 1. The received binary sequence pair $(R_1(D),\, R_2(D))$ is multiplied by H^T as in (1). The resulting syndrome sequence enters a counter. After a number of zero syndrome values, the counter gives a ready-to-read S' signal. The incoming sequence pair is delayed by a time d before entering the inverse forming circuit G^{-1} as given in (2). The delay d plays a key role in the determination of S'. The register following the invertor has length m and contains m symbols of the estimate S'. Hence, the content of the register is used as a "state-tracker." Of course, we have to specify the basic operations of the decision mechanism to be used as a state tracker.

After receiving a syndrome value equal to one, the counter is reset to zero. After receiving $d+m+m'+i$, $i \geq 0$, syndrome values equal to zero, the signal ready-to-read S' is given. We thus can formulate the following decision rule.

Decision Rule: The content of the "state-tracker" at time $t + d + m + m' + 1 + i$ is given as an estimate for the encoder state at time $t + m + m' + 1 + i$ if $d + m + m' + i$ subsequent values of the syndrome are equal to zero (ready-to-read S' signal).

In the following section we analyze the performance of the decision rule.

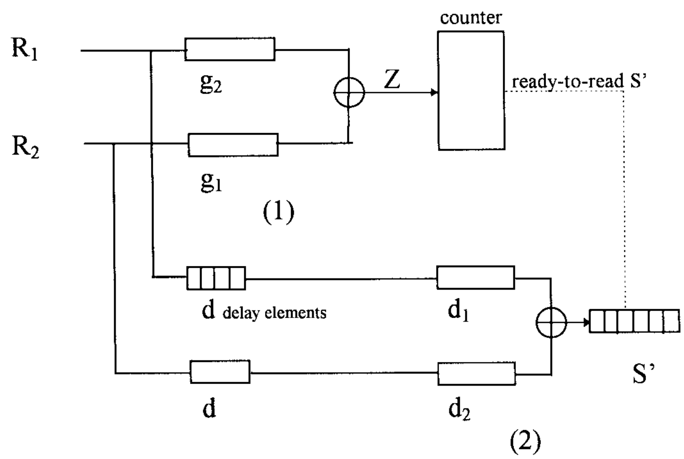

Fig. 1. (1) Syndrome former and (2) inverse forming circuit with "state-tracker" S'.

III. Performance Analysis

In this section we determine the estimation error probability. The estimation error probability will be shown to depend on the time-to-decision (TTD) and the signal-to-noise ratio on the channel. The time-to-decision interval equals a certain specified number of subsequent zeros in the syndrome former output sequence after a syndrome value equal to one appeared. As indicated above, we set the TTD equal to $d + m + m' + i$. The reason for taking this length will be clear from the derivation of the estimation error probability. As a channel we assume the hard decision additive white Gaussian noise (AWGN) channel, or binary symmetric channel with transition probability p. For high signal-to-noise ratio (SNR), SNR $= 10\,\log(E_s/N_0)$, the transition probability p is proportional to $\exp(-E_s/N_0)$. In the coded situation with rate $R = k/n$, $E_s = RE_b$, where E_b is the available energy per information bit.

Definition: An estimation error occurs whenever the "state-tracker" output or encoder state estimate S' at time $t+m+m'+d+1+i$ is not equal to the actual encoder state at time $t+m+m'+1+i$.

We first concentrate on the estimation error probability. Since all operations in the receiver are linear, we may assume transmission of the all-zero codeword.

Observation: From Figs. 1 and 2, we observe that the content of the state-tracker at time $t + d + m + m' + 1 + i$ is determined by the noise input pairs at time

$$t + 1 + i, t + 2 + i, \cdots, t + m + m' + i.$$

Since we consider transmission of the all-zero codeword, the description only has to consider the channel noise digits. Let H_t denote the abstract state of the syndrome former at time t, and n_t the corresponding input pair of noise digits. The syndrome former output, in the adjoint obvious realization, at time t is denoted as Z_t. Suppose that $Z_t = 1$. In addition, the following $d+m+m'+i$ values of Z_{t+1}, Z_{t+2}, $\cdots$, $Z_{t+d+m+m'+i}$ are assumed to be zero. According to the decision rule, the state-tracker then gives an estimate for the encoder state at time $t + m + m' + 1 + i$. For any syndrome former state H_{t+1} we can construct input pairs n_{t+1}, n_{t+2}, $\cdots$, $n_{t+m+m'+i+d}$ such that the corresponding syndrome outputs are equal to zero. Since every abstract syndrome former state corresponds uniquely with an encoder state, the above inputs must be a code sequence segment that follows a path through the encoder state diagram starting at the corresponding encoder state. We need the following definition.

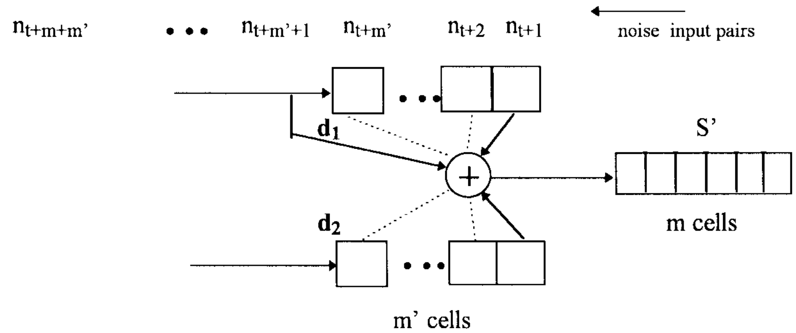

Fig. 2. Invertor with state-tracker and corresponding inputs for $i = 0$.

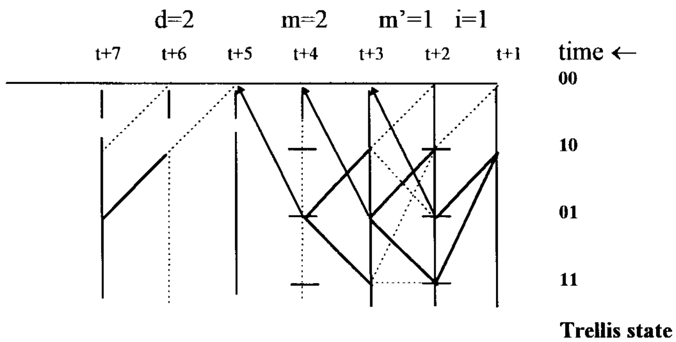

Fig. 3. Illustration of a set of events that do not cause estimation errors.

Definition: A code sequence segment generated by an encoder input that leads to the all-zero encoder state and stays in the all-zero state is said to end in the all-zero state.

Now, consider only those code sequence segments as input to the syndrome former that reach the all-zero state at or before time $t+m+m'+1+i$ and do not leave the all-zero state in the time interval $t+m'+1+i$ to $t+m+m'+1+i$. As a consequence, for these inputs, the state-tracker content at time $t+m+m'+1+i+d$ must contain the all-zero state. This observation leads to the following theorem.

Theorem: Noise inputs that correspond to code sequence segments ending in the zero state in the time interval $(t + m' + 1 + i$ to $t + m + m' + 1 + i)$ do not cause estimation errors at time $t + d + m + m' + 1 + i$.

Proof: Any code sequence segment ending in the all-zero state in the defined time interval must have at least m encoder inputs equal to zero at time $t + m + m' + 1 + i$.

All other syndrome former inputs that lead to the zero syndrome outputs, but are not equivalent to a code sequence segment that ends in the zero encoder state in the time interval from $t + m' + 1 + i$ to $t + m + m' + 1 + i$ may cause a nonzero state-tracker content at time $t + m + m' + d + 1 + i$ and thus causes an estimation error. The syndrome former inputs that may cause errors in the state-tracker thus contain the following remaining classes of error events.

$-\mathbf{E_1}$*:* Error events equal to code sequences that start from the all-zero state at time $t+1,\ t+2,\ \cdots,\ t+m+m'+i$, that do not end in the zero state in the interval $(t+m'+1+i$ to $t+m+m'+1+i)$. Note that these error events must have a length of at least $d+1$ pairs, when $d \le m$. For $d > m$, the minimum length of the error event is $m + 1$. Here, the variable d is the key parameter.

$-\mathbf{E_2}$*:* Error events equal to code sequence segments that start from a particular encoder state at time $t + 1$, but do not end in the all-zero state in the interval $(t + m' + 1 + i$ to $t + m + m' + 1 + i)$. These error events have a minimum length of $m + m' + 1 + i$.

In Figs. 3 and 4, we use a trellis representation of code sequences as paths through the trellis for an $m = 2$ encoder. Dashed lines correspond to an encoder input 1 and solid lines to an encoder input 0. On the horizontal axis we have the time and on the vertical axis we have the possible encoder states in the trellis representation. As an example, we take the simple encoder with $m = 2$, $m' = 1$, $d = 2$, and $i = 1$. Fig. 3 shows examples of syndrome-former inputs that do not cause estimation errors. For instance, there is an event from state (10) at time $t+1$ passing state (11) at time $t+2$ and ending in state (00) at time $t+4$. In Fig. 4 we illustrate both classes of error events in the encoder trellis diagram.

The first class of error events immediately gives the importance of the delay d. If we take a small value of d, some light-weighted error events may cause errors in the state-tracker. Values of d larger than m are not expected to improve the estimation error probability considerably. This will be illustrated in the simulation results. From the two classes of error events, we may upperbound the estimation error probability by

$$P_e \le P(E_1) + P(E_2). \tag{3}$$

For the class E_1, the Hamming weight of the error segments of length $d+1,\ d+2,\ \cdots$, can be found from the encoder state diagram. The probability that a particular code sequence segment of length $d+x$, $x > 0$ with Hamming weight y occurs is given by

$$P(d+x) = p^y (1-p)^{2(d+x)-y}. \tag{4}$$

The maximum value of (4) occurs when $x = 1$ and y assumes the lowest value for a particular value of d [3]. We also conclude that $d = m$ is a reasonable choice, since for values of $d > m$, the minimum distance of the code is the dominant factor. The minimum value for y can be used to approximate (4) as

$$\log P(d+1) \cong D_{d+1} \log p \tag{5}$$

where D_{d+1} is the minimum weight of an error segment of length $d+1$ for the class E_1. From (3) and (4), it can be concluded that for a given d, a rapid growth in the Hamming weight for code segments of length $d+1,\ d+2,\ \cdots$, is important for events in the class E_1. Hence, here the distance profile [2], [3] is shown to be again an important measure in the performance of convolutional decoding.

For the class E_2, the minimum-weight error segment of length $m+m'+1+i$ can be found by searching the encoder state diagram in backward direction. From inspection of the code properties of the particular code in use, one finds that this event is of much less importance than the event E_1. However, in general, we have to confirm this by code inspection. The influence of the class can be controlled by the variable i.

We also analyze the mean time to decision (MTTD), defined as the average time it takes before we observe a specified number of zero

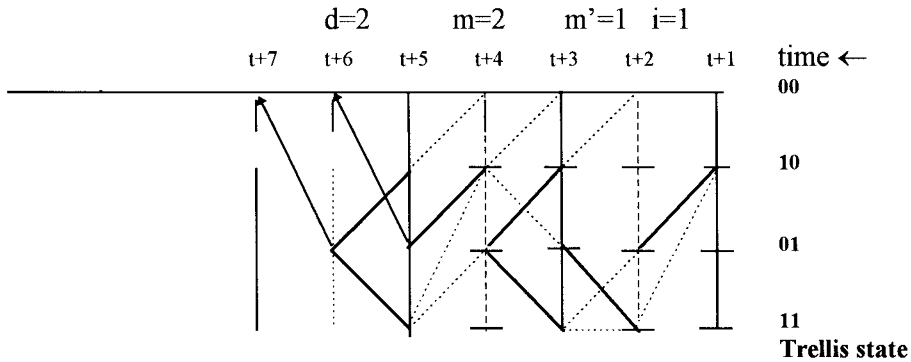

Fig. 4. Illustration of a set of error events that cause estimation errors.

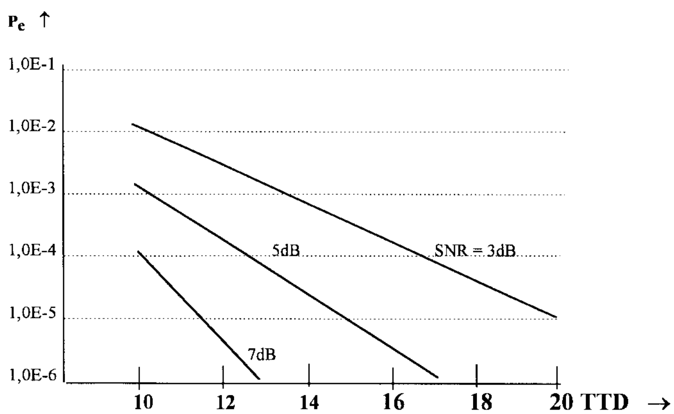

Fig. 5. Estimation error probability as a function of TTD $= 10 + d$.

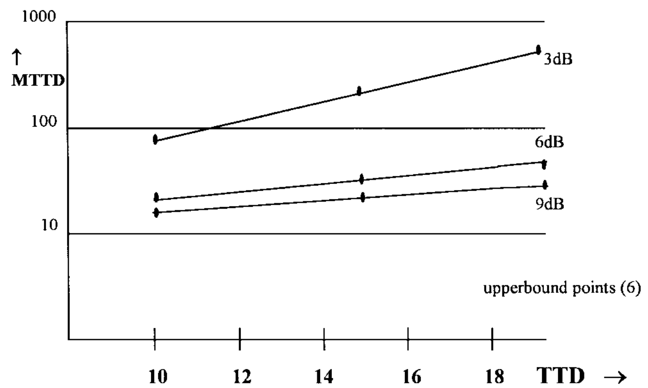

Fig. 6. The average number of received input pairs needed to reach the TTD.

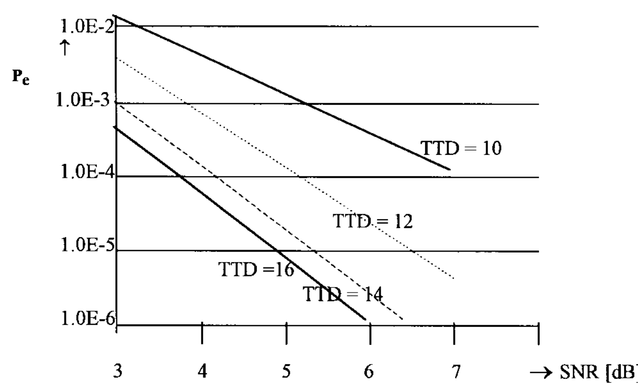

Fig. 7. Estimation error probability as a function of SNR, TDD $= d + 10$.

syndrome-former outputs. After receiving a syndrome value equal to one, we wait for $d + m + m' + i$ zero outputs before we make a decision. The question is how long do we have to wait before we have $d+m+m'+i$ zero outputs. We give an upper bound for MTTD. The average time it takes for $j - m$ consecutive syndrome outputs equal to zero to occur is less than or equal to the average time it takes to receive j noise digit input pairs equal to zero, since more (including nonzero) error events may cause a sequence of syndrome values equal to zero. Let $E(j)$ denote the average time it takes before j noise pairs equal to zero occur. We can thus upperbound the MTTD as

$$\text{MTTD} \le E(d + m + m' + i + m)$$

where

$$E(j) = \{(1-p)^{-2j} - 1\}/\{1 - (1-p)^2\}, \qquad j \ge 1. \tag{6}$$

For small values of p

$$E(j) \approx j + j^2 p. \tag{7}$$

In the next section we present simulation results for the MTTD and P_e.

IV. Measurements

In this section we present the simulation results for the state tracking method. As a reference code we use the standard nonsystematic constraint length 6 encoder with connection polynomials $(g_1, g_2) = (744, 554)$ and degree 4 inverse $(d_1, d_2) = (76, 12)$. After receiving TTD syndrome values equal to zero, we estimate the encoder state. After estimation, we start again with a randomly selected new encoder state. Simulations were carried out for SNR $= 10 \log(E_b/N_0)$ equal to 3, 5, and 7 dB, respectively. Fig. 5 gives the fraction of false state estimates as a function of TTD $= d + 10$. The logarithm of the estimation error probability $\log P_e$ has a slope depending on the signal-to-noise ratio, and decreases linearly with d. For $R = 1/2$, we know that the column distance function roughly grows linearly with the length of the considered codeword segment, and thus with increasing d. For the standard code, the column distance grows as $2, 3, 3, 4, 4, 4, 4, 5, 5, \cdots$. Given the hard decision channel error probabilities for SNR = 3, 5, and 7 dB, the derivatives in Fig. 5 are -0.3, -0.45, and -0.7, respectively. We conclude from Fig. 5 that $\log P_e \cong 0.3\,\text{TTD} * \log p$. Hence, Fig. 5 is in agreement with (5). We conclude that the described method can be used to improve the reliability of the state-tracking method for convolutional codes.

In Fig. 6 we give the average number of received channel output pairs before we have TTD syndrome values equal to zero, with the SNR as a parameter.

From Fig. 6 we see that the $\log$ (MTTD) increases linearly with d and has a slope that depends on the signal-to-noise ratio, see also (7). The results for P_e as a function of the SNR and TTD as a parameter are given in Fig. 7. From the simulation results it follows that TTD ≥ 20 does not give significant improvements since a saturation effect occurs for TTD ≥ 20. We observe that introducing a delay $d = 6$ already gives a gain of more than 3 dB at an error probability of 10^{-4}. For larger values of d, the expected performance is determined by the free distance of the code, see (4).

Remarks: The described procedure gives a reliable state estimate if sufficient syndrome values equal to zero are received. Instead of estimating encoder states, we can also estimate encoder inputs. These estimates can then be used in decoding algorithms like that of Fano or the M-algorithm. We are presently investigating these applications.

V. Conclusions

We have shown how to use the inverse G^{-1} as a reliable encoder state-tracker. The estimation error probability strongly depends on the introduced decision delay d and the SNR. The method has a very low complexity and can be used to support decoding algorithms, like the Fano sequential decoding algorithm or the M-algorithm, where the presence or knowledge of the encoder state in the decoding process can be expected to improve performance of the decoding. We showed the relation between the estimation error probability and the column distance function. We also bounded the mean time to decision (MTTD) as an average for the time it takes before we observe the desired TTD $= d + m + m' + i$ zero syndrome outputs.

Acknowledgment

The authors wish to thank Adriaan van Wijngaarden for his support during the stay of the second author in Essen.

References

[1] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.
[2] R. Johannesson, "Robustly optimal rate one-half binary convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 464–468, July 1975.
[3] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

Critical Lengths of Error Events in Convolutional Codes

Jørn Justesen and Jakob Dahl Andersen

***Abstract*—If the calculation of the critical length is based on the expurgated exponent, the length becomes nonzero for low error probabilities. This result applies to typical long codes, but it may also be useful for modeling error events in specific codes.**

***Index Terms*—Burst lengths, convolutional codes, critical length, ensemble analysis.**

I. Introduction

In the analysis of concatenated codes and other systems combining convolutional codes with multiplexing or other stages of coding it is important to know the distribution of error events after decoding of the convolutional code. For a specific code this distribution can be simulated or calculated from the weight generating function of the code and the properties of the channel [1]. However, it is of interest to compare these results to general properties of error patterns by considering the performance of average codes of sufficiently large constraint length.

The analysis of error-correcting codes through the error exponents for randomly chosen codes is a classical technique in information theory [2]. Following this tradition, Forney [3] introduced the concept of the critical length for long convolutional codes as the length of the error events that dominate the lower bound on error probability. This analysis, which also appears in Viterbi and Omura's text [1], indicates that for rates below R_0 the critical length is zero. However, for typical binary codes, the critical length is determined by the length of the minimum-weight codewords, and consequently it is greater than zero. For higher rates, the error events are longer, and the bound on error probability is known to be tight.

In Section II, we discuss the derivation of error exponents for convolutional codes, and make some comments on the relationship between distances and error exponents. In Section III, the correction to the critical length is presented, and in Section IV the critical length is derived as a function of the signal-to-noise ratio. In Section V the exponential decrease in probability for long burst is analyzed and finally, in Section VI, the results of the asymptotic analysis are compared to simulated and calculated results for a specific code with moderate memory.

II. Error Exponents for Block Codes and Convolutional Codes

For simplicity we shall discuss only the performance on the binary-symmetric channel (BSC) and the additive white Gaussian noise channel (AWGN). For long codes, the error probability of block codes and convolutional codes is upper-bounded in terms of the error exponents. Usually the exponents are derived for general memoryless channels and later specialized to BSC and other simple cases. This approach may obscure the arguments. However, we shall not give a simplified derivation but rather give a few comments interpreting the

Manuscript received November 14, 1995; revised December 12, 1997. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, Trondheim, Norway, 1994.

The authors are with the Department of Telecommunication, Technical University of Denmark, DK-2800 Lyngby, Denmark.

Publisher Item Identifier S 0018-9448(98)03757-2.