Systematic feed-forward convolutional encoders are better than other encoders with an M-algorithm decoder

Osthoff, Harro; Anderson, John B; Johannesson, Rolf; Lin, Chin-foo

Published in:
IEEE Transactions on Information Theory

DOI:
10.1109/18.661531

1998

[13] A. B. Sesay and M. R. Gibbard, "Asymmetric signal processing for indoor wireless LANs," in *6th IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications* (Toronto, Ont., Canada, Sept. 1995), pp. 6–10.

[14] R. D. Wesel, "Trellis code design for correlated fading and achievable rates for tomlinson harashima precoding," Ph.D. dissertation, Stanford University, Stanford, CA, Aug. 1996.

[15] J. M. Cioffi, G. P. Dudevoir, M. V. Eyuboğlu, and G. D. Forney, Jr., "MMSE decision-feedback equalizers and coding—Parts I and II," *IEEE Trans. Commun.*, vol. 43, pp. 2582–2604, Oct. 1995.

[16] J. M. Cioffi, "Signals and detection," in *Stanford Univ. Lecture Notes*, 1994.

[17] A. K. Aman, R. L. Cupo, and N. A. Zervos, "Combined trellis coding and DFE through tomlinson precoding," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 876–884, Aug. 1991.

[18] G. J. Pottie and M. V. Eyuboğlu, "Combined coding and precoding for PAM and QAM HDSL systems," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 861–870, Aug. 1991.

[19] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, "Capacity penalty due to ideal zero-forcing decision-feedback equalization," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1062–1071, July 1996.

[20] S. Shamai (Shitz) and R. Laroia, "The intersymbol interference channel: Lower bounds on capacity and precoding loss," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1388–1404, Sept. 1996.

Systematic Feed-Forward Convolutional Encoders Are Better than Other Encoders with an M-Algorithm Decoder

Harro Osthoff, John B. Anderson, *Fellow, IEEE*,
Rolf Johannesson, *Fellow, IEEE*, and Chin-foo Lin

***Abstract*—Consider nonbacktracking convolutional decoders that keep a fixed number of trellis survivors. It is shown that the error performance of these depends on the early part of the distance profile and on the number of survivors kept, and not on the free distance or the details of the code generators. Particularly, the encoder may be feed-forward systematic without loss. Furthermore, this kind of encoder solves the correct path loss problem in reduced-search decoders. Other kinds do not. Therefore, with almost any other decoding method than the Viterbi algorithm, systematic feed-forward encoders should be used. The conclusions in this correspondence run counter to much accepted wisdom about convolutional codes.**

***Index Terms*— Correct-path loss problem, list decoder, M-algorithm decoder, systematic feed-forward encoders.**

I. Introduction

Growing evidence in recent research has shown that the error performance of a channel decoder depends primarily on the decoder's resources, and only secondarily on the qualities of the error-correcting code itself, such as its free distance and the state-space size of its minimal encoder. It is well established, for example, that the performance of a sequential decoder of the Fano or stack algorithm type depends on how many code trellis paths the decoder can visit. The precise choice of the code matters little, so long as the code has a good distance profile and a reasonable memory.

In a traditional sequential decoder, the design begins by assuming a long code and by assuming that the decoding of a data symbol will be correct with certainty; an analysis then finds the expected number of code trellis paths (or tree paths, or tree branches) that are required to do this. So long as the code rate is less than the cutoff rate, this expectation is finite, but the actual path number outcome may on occasion exceed any bound, a phenomenon called erasure. The working storage for these paths and the speed in which they may be viewed comprise the resources of the decoder, and it is the need for these that depends weakly on the code.

In more recent times, this design philosophy has often been reversed. Rather than fix the error probability at zero, the storage of the paths is fixed; by means of analysis and measurement, the error probability or the number of correctable errors is found. Two examples of this philosophy are the Viterbi algorithm and the M-

Manuscript received October 11, 1996; revised August 15, 1997. This work was supported in part by the Swedish Research Council for Engineering Sciences under Grant 91-91. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, San Antonio, TX, Jan. 17–22, 1993.

H. Osthoff is with Ericsson Eurolab Deutschland GmbH, D-90411 Nürnberg, Germany.

J. B. Anderson is with the Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA.

R. Johannesson is with the Department of Information Technology, Information Theory Group, Lund University, Box 118, S-221 00 Lund, Sweden.

C.-f. Lin is with Chung-Shan Institute of Science and Technology, P.O. Box 1-322, Lung-Tan, Touyuan, Taiwan, R.O.C.

Publisher Item Identifier S 0018-9448(98)00834-7.

algorithm (list decoder). In the first, the storage is fixed to the encoder state-space size, and depends therefore critically on the code. In the second, the storage is fixed at M paths, independently of the code. Decoders of the M-algorithm type, whose storage is fixed *a priori*, show a strong independence between error performance and code.

The purpose of this correspondence is to relate the storage and error performance of simple breadth-first decoding of convolutional codes, and to investigate how dependent they really are. We will show that the important aspect of a code is the early growth of its distance profile [1], and that apart from this, the error performance of M-algorithm type decoders depends to a first approximation on their storage only. We do this by two methods, for best-d_{free}, QLI [2], and feed-forward systematic encoders. First, we construct decoders and test them over a simulated binary-symmetric channel (BSC); second, we estimate error performance by counting the error sequences that each decoder corrects. All of these turn out to have the same event error performance, so long as the path storage and their distance profiles approximately match. Conversely, codes with markedly different distance profiles have markedly different decoder error rates.

As a consequence, an encoding by a feed-forward systematic encoder with a proper distance profile decodes incorrectly about as often as with any other encoder in a fixed-storage decoder. Such encoders are easily designed. More important, they address a major shortcoming of convolutional decoders that do not search the entire code trellis. These decoders occasionally lose the correct path completely from the storage and need a long time to find it again, an event that leads to a long error burst in the estimated data. feed-forward systematic encoders make it easy to get the correct path at least back into the storage, and from there, correct decoding rapidly follows. Consequently, they should perform much better than other encoder types with a reduced-search trellis decoder, and we show in later sections that this is indeed the case.

The phenomena in this correspondence are easiest to see when the test data are presented in terms of an underlying Gaussian channel. Our channel can be thought of as an antipodal binary modulation (such as binary PSK) passed through additive white Gaussian noise, modeled as a BSC. For noise density $N_0/2$, data bit energy E_b, and code rate R, the crossover probability p in the BSC is given by [3]

$$p = Q(\sqrt{2RE_b/N_0}) \tag{1}$$

with

$$Q(x) \triangleq (1/\sqrt{2\pi}) \int_x^\infty \exp(-u^2/2)\, du, \qquad x \geq 0. \tag{2}$$

Although the decoder sees a BSC, its error performance is a function of E_b/N_0 through these formulas, and the performance takes on the familiar "waterfall" shape when plotted against E_b/N_0 in decibels. Error performance has two meanings in this correspondence. The error event probability P_{ev} of a decoder is the probability that a trellis error begins at a certain trellis level, given that decoding has been correct so far. Trellis error events begin when the decoded trellis path splits from the correct path and end when it merges again. Events can be short, if for example the decoder chooses a nearest neighbor in the trellis, or they can be long. The second measure of decoder error in this correspondence is the overall bit-error rate (BER). Most decoder properties are easiest to see from P_{ev}. P_{ev} is thus an important analysis tool, although, of course, BER remains a measure of practical importance.

Data-bit errors are caused, of course, by error events, but the BER and P_{ev} can differ widely, depending on how many data-bit errors occur in the events. After losing the correct path, a reduced-search decoder can flounder about for an indefinite number of trellis levels trying to find it again. During this time, the BER is essentially 50%, and the BER is only barely related to the P_{ev}, which is defined independently of any correct-path loss event. We will show that the choice of a feed-forward systematic encoder solves the correct-path loss problem.

As we have mentioned, much of the recent work in reduced-search decoding focuses on decoders with fixed storage or with some other strict limitation to the search. Early work on M-algorithm convolutional decoders was performed by Zigangirov and Kolesnik [4]. Simmons and Wittke [5], Aulin [6], Balachandran [7], and Palenius [8], among others, have applied the M-algorithm to CPM modulation codes. They all report a *saturation* phenomenon, in which the plot of P_{ev} versus E_b/N_0 improves steadily with M only up to a certain small M, after which there is little improvement. In the removal of intersymbol interference with the M-algorithm, Seshadri and Anderson [9], and Gozzo [10], also find an even stronger saturation effect. The effect appears as well in decoders whose search is confined to the range of path metrics, rather than by a limit to the number of paths. Aulin has discussed vector Euclidean distance and some other related matters in [11] and [12].

In our previous work [13], we treat M-algorithm decoding of convolutional codes over the BSC. That work reports the saturation phenomenon, but reports a new one as well. For the QLI convolutional encoders tested and M below the saturation value, the P_{ev} versus E_b/N_0 plot does not depend on the encoder, but only on M. Our purpose now is to extend this work and determine what quality of a convolutional code, if any, sets the P_{ev}, when M is below the saturation M.

Since the M-algorithm and its strict limit to path storage play a central role in the correspondence, we summarize the state of knowledge about it here. A practical implementation keeps in storage M paths of length t_D, where t_D, the decoder *decision depth*, is the number of trellis levels behind the front of the search at which a data bit is released as output. In each basic cycle of the algorithm, a data bit is released at t_D behind the front, all paths are extended one trellis level deeper, and the best M survivors are retained. We also delete any paths that do not correspond to the released data bit and that represent state duplications of a better path in storage, although these features do not much change the decoder performance. Further implementation details appear in [3].

The proper value for t_D (see [3] or [14]) for nonbacktracking decoders and known, concrete codes is approximately

$$t_D \approx 2e/\rho c \tag{3}$$

where $R = b/c$ is the code rate, e is the number of errors the decoder is assumed to correct, and ρ is the Gilbert–Varshamov parameter, i.e., the solution of

$$h_B(\rho) = 1 - R \tag{4}$$

where $h_B(\cdot)$ is the binary entropy function. We use this t_D in our decoders, although considerably shorter values may be used without compromising P_{ev} very much. Regarding the size of M, certain properties are known. If the design of the decoder is to guarantee the correction of any e errors in the length t_D, a *bounded- distance decoder* (BDD) design, then M must satisfy [15]–[17]

$$M \approx (2^{1-R} - 1)^{-e}. \tag{5}$$

On the other hand, if the design specifies that the decoder must *achieve a certain* $P_{\rm ev}$, then the path numbers are entirely different; this is discussed in [3] and [18].

It is well known that a good computational performance for sequential decoding requires a rapid initial growth of the column distances. This led to the introduction of the *distance profile* [1]:

Definition: Let $\mathcal{C}$ be a convolutional code encoded by a minimal-basic generator G [19] of memory m. The $(m+1)$-tuple

$$\boldsymbol{d}^p = \left(d_0^c, d_1^c, \cdots, d_m^c\right)$$

where d_j^c, $0 \le j \le m$, is the jth-order column distance of G, is called the distance profile of the code $\mathcal{C}$.

A generator matrix of memory m is said to have a distance profile $\boldsymbol{d}^p$ superior to a distance profile $\boldsymbol{d}^{p\prime}$ of another generator matrix of the same rate R and memory m, if there is some l such that

$$d_j^c \begin{cases} = d_j^{c\prime}, & j = 0, 1, \cdots, l-1 \\ > d_j^{c\prime}, & j = l. \end{cases} \tag{6}$$

Moreover, a convolutional code is said to have an *optimum distance profile* (ODP) if there exists no generator matrix of the same rate and memory whose code has a better distance profile. An ODP generator causes the fastest possible initial growth of the minimal separation between the encoded paths diverging at the root in a code tree.

Furthermore, it has been shown [20] that there exists a binary, rate $R = b/c$, time-invariant convolutional code with a generator matrix of memory m whose column distances satisfy the inequality

$$d_j^c \ge \rho c(j+1) \tag{7}$$

for $0 \le j \le m$, and where ρ is the Gilbert–Varshamov parameter. From (7) it follows that at length $t_D = j+1$ we can have a distance d between paths out of a trellis node that is at least $\rho c t_D$, or equivalently

$$t_D \le d/\rho c. \tag{8}$$

This again suggests rule (3) for the decoder decision depth.

The organization of the correspondence is as follows. In Section II we give the observed $P_{\rm ev}$ versus E_b/N_0 for an M-algorithm decoder with best-$d_{\rm free}$, QLI, and feed-forward systematic encoders, and find that $P_{\rm ev}$ depends only on the distance profiles and M. In Section III we estimate $P_{\rm ev}$ by an error sequence counting argument, and come to the same conclusion. Section IV compares the $P_{\rm ev}$ of the M and Viterbi algorithms. In Section V we turn to the BER, a quantity that depends critically on how the decoder recovers from losses of the correct path; we conclude that only the systematic encoder can improve upon the Viterbi BER.

II. Distance Profile and M: Simulations

In this section we will establish by tests of an actual encoder/decoder that the error event performance of an M-algorithm type decoder depends for the BSC almost entirely on the distance profile and apparently not on the details of the code generators or on the free distance. The tests measure the error event frequency, as discussed in Section I, and thus do not measure the efficiency of recovery from an error event (this is studied in Section V). (In the following simulations we used frames consisting of 1024 information bits. The number of simulated frames was chosen such that at least 100 distinct error events occurred; for example, with BSC crossovers $p = 0.037$ and 0.033, the numbers of frames were respectively 250 000 and 500 000 or more. In addition, there were selected tests of very long frames to confirm that catastrophic events occur with negligible probability.)

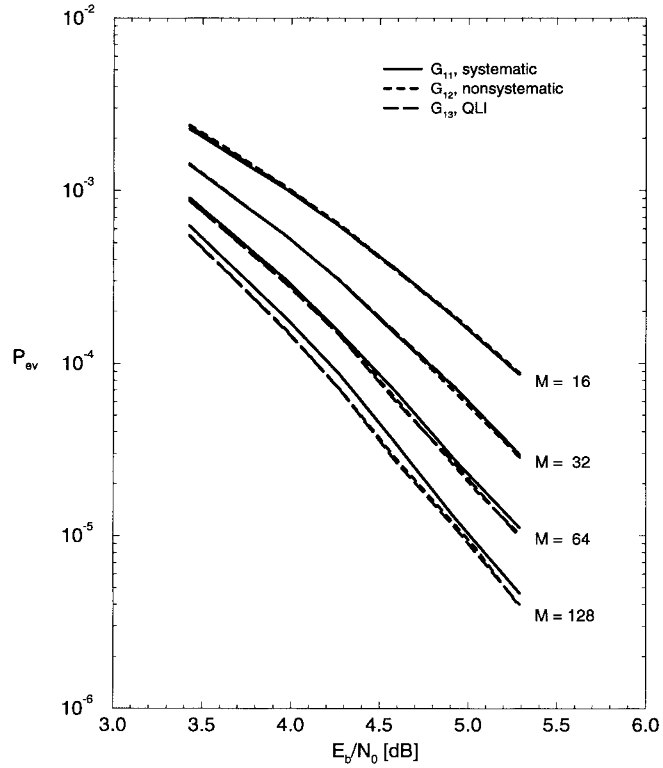

Fig. 1. Event error probabilities observed in tests of three different types of encoders with approximately the same distance profiles.

$G_{11} = (4000000\;7144655), m = 20, d_{\rm free} = 13;$
$G_{12} = (6055051\;4547537), m = 20, d_{\rm free} = 22;$
$G_{13} = (7404155\;5404155), m = 20, d_{\rm free} = 18.$

Fig. 1 plots the observed $P_{\rm ev}$ against E_b/N_0 for a systematic and a nonsystematic encoder that have the same distance profile, as well as a best-$d_{\rm free}$ QLI encoder, whose distance profile is approximately the same. All encoders have the same memory, which means that they all use the same resources. At the M's in the figure $(16, 32, 64, 128)$ the difference in the measured $P_{\rm ev}$ is statistically insignificant across the three encoders. This occurs despite the fact that $d_{\rm free}$ varies widely among the three encoders. We have noticed similar behavior for smaller M's in our earlier work [13].

A very different result occurs when encoders with widely varying early distance profiles are tested. Fig. 2 compares the observed $P_{\rm ev}$ for the memory 12 best-$d_{\rm free}$ QLI encoder, decoded with $M = 32$, to $P_{\rm ev}$ for the encoder with both generators reversed. This reversal creates an encoder with the same $d_{\rm free}$, but with a distance profile that develops late in the trellis rather than early. Despite the identical free distance, $P_{\rm ev}$ for the reversed encoder is nearly tenfold worse.

Fig. 3 compares $P_{\rm ev}$ tests at $M = 32$ for a sequence of five encoders, all with $d_{\rm free} = 10$, each of whose distance profile successively underbounds the others. The result is a sequence of widely varying $P_{\rm ev}$ curves arranged in the same order.

Almost all the variation in decoder $P_{\rm ev}$ for the encoders that we tested can be traced to variations in the early distance profile. It is reasonable that variation later in the distance profile might affect $P_{\rm ev}$ for large M, but these M are of less practical interest. The following hardly affect $P_{\rm ev}$: The type of encoder (systematic or not, QLI or

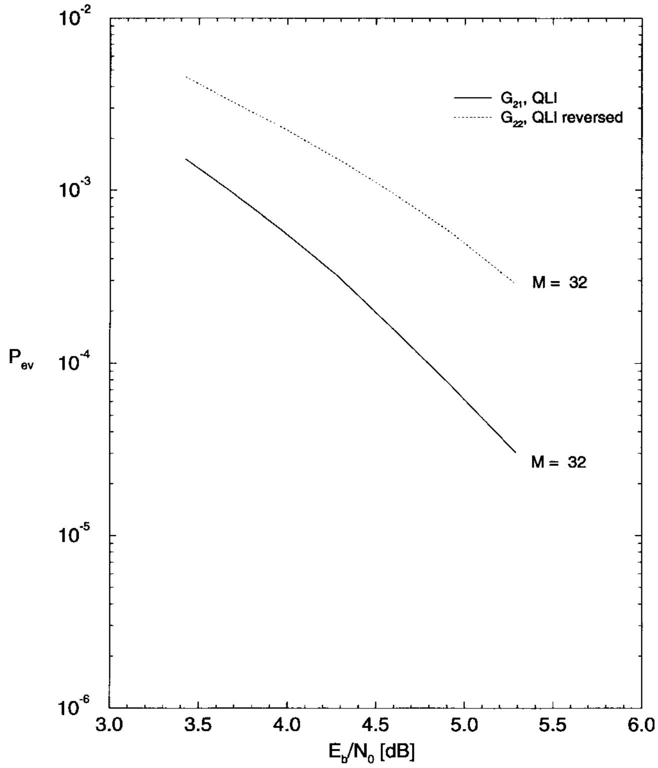

Fig. 2. Event error probabilities observed in tests of a QLI encoder and the corresponding reversed encoder when decoded with $M = 32$.

$G_{21} = (76414\ 56414), m = 12, d_{\text{free}} = 14;$
$G_{22} = (60574\ 60564), m = 12, d_{\text{free}} = 14.$

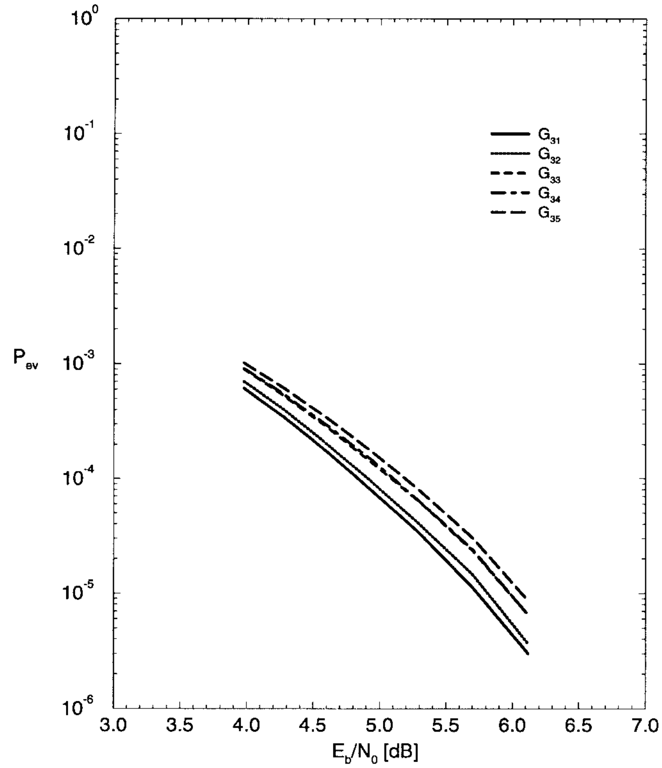

Fig. 3. Simulated event error probabilities observed in tests of a set of systematic encoders, each of whose distance profile $\boldsymbol{d}^p$ successively underbounds the others, when decoded with $M = 32$. The heavy line is the ODP feed-forward systematic encoder G_{31} which always performs best.

$G_{31} = (400000\ 714474), m = 15, d_{\text{free}} = 10,$
$\boldsymbol{d}^p = (2,3,3,4,4,5,5,6,6,6,7,7,8,8,8,8);$
$G_{32} = (400000\ 552234), m = 15, d_{\text{free}} = 10,$
$\boldsymbol{d}^p = (2,2,3,4,4,4,5,5,5,6,6,6,7,7,8,8);$
$G_{33} = (400000\ 447254), m = 15, d_{\text{free}} = 10,$
$\boldsymbol{d}^p = (2,2,2,3,3,3,3,4,5,6,6,6,6,7,7,7);$
$G_{34} = (400000\ 427654), m = 15, d_{\text{free}} = 10,$
$\boldsymbol{d}^p = (2,2,2,2,3,3,4,5,5,5,5,5,5,6,6,6);$
$G_{35} = (400000\ 417354), m = 15, d_{\text{free}} = 10,$
$\boldsymbol{d}^p = (2,2,2,2,2,3,4,5,5,6,6,6,6,6,6,7)$

not), the free distance, the encoder memory. Depending on M, the last two must exceed a minimum threshold.

As a final example, we compare a nonsystematic encoder to its equivalent systematic encoder. Consider the memory $m = 31$ QLI ODP nonsystematic convolutional generator matrix (octal notation) $G_{\text{nonsys}} = (74041567512\ 54041567512)$ with $d_{\text{free}} = 25$. By long division of the generator polynomials and truncation after degree m, we obtain the memory $m = 31$ ODP systematic convolutional generator matrix $G_{\text{sys}} = (40000000000\ 67115143222)$ with $d_{\text{free}} = 16$. These two generator matrices are equivalent over the first memory length, and consequently they have the same distance profile. In Fig. 4 we compare for various M the error event probability P_{ev} observed at the root of the code tree for these two encoders. The outcomes of the two simulations are virtually identical at each M.

III. Distance Profile and M: Counting Estimate

Because the error rates among different encoders with the same distance profile are so close, a more accurate measurement than experimental observations is needed for P_{ev}. Also, experiments do not give accurate estimates at small P_{ev}. We therefore estimate P_{ev} combinatorically by counting the number of channel error sequences that can lead to a decoding error in the first data symbol. Let the BSC crossover probability be p, the decoder decision depth be t_D trellis levels, or $n_D = ct_D$ bits, and suppose the encoder, M, and t_D are sufficient to guarantee the correction of all combinations of e or fewer channel errors. Furthermore, for this M-algorithm decoder suppose that $N(d)$ channel error sequences of Hamming weight d can potentially lead to decoder error. Then the error event probability is overbounded by

$$P_{\text{ev}} \leq \sum_{d=e+1}^{n_D} N(d)p^d(1-p)^{n_D-d}. \tag{9}$$

The remaining task is to count the sequences that can lead to decoder error. This is done by a search of the decoder trellis with M paths retained, once for each candidate error sequence. As the search progresses, at each level there will be a certain number of trellis paths at or below the Hamming distance of the correct path. If this number never exceeds M, the correct path cannot be dropped, it eventually is closer to the received path than any incorrect-subset path, and decoding of the first trellis branch is correct with certainty. If the number exceeds M, the correct path can be worse than M paths at some trellis level, or it can be tied for worst among the M. In either case, we count the error sequence in the total $N(d)$, which makes (9) an overbound.

It can happen that the errors in a sequence concentrate later in the sequence, and because the decoder keeps only M paths, all incorrect subset paths are dropped before the errors are reached. In this case, correct decoding of the first symbol is certain; the reduced search has in fact rescued the decoder, so far as the first branch is concerned. If the candidate error sequences are tested in the right order, most can

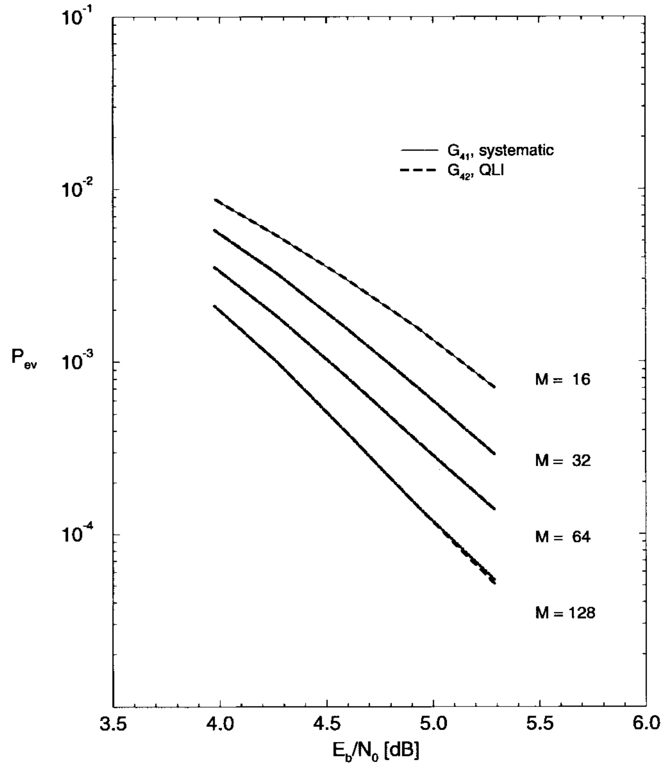

Fig. 4. Event error probabilities observed in tests of a systematic and nonsystematic encoder that are equivalent over the first memory length.

$G_{41} = (40000000000\ 67115143222), m = 31, d_{\rm free} = 16;$
$G_{42} = (74041567512\ 54041567512), m = 31, d_{\rm free} = 25.$

be skipped over for this reason and the counting becomes a relatively efficient procedure.

If the correct path fails to be unquestionably within the best M paths, the most likely event is that it is one Hamming unit too heavy and that it is therefore tied with others as a candidate to be dropped. We assign a probability of $1/2$ to the event that the tie-break keeps the correct path; we thus apply a factor of $1/2$ to (9). Finally, the tail of (9) beyond, say, weight w is easy to compute, if we assume a decoder error is certain

$$P_{\rm tail} = 1 - \sum_{d=0}^{w} \frac{n_D!}{(n_D - d)!d!} p^d (1-p)^{n_D - d}. \tag{10}$$

Beyond a certain w, this expression decays rapidly. In order to reduce the calculation of $N(d)$, it can be substituted for the tail in (9) at the earliest convenient moment; the effect will make (9) a little more of an overbound.

We turn now to results obtained with (9). Fig. 5 plots the counting estimate of $P_{\rm ev}$ for another collection of encoders with similar good distance profiles and relatively small M. This time, the encoders are the best-$d_{\rm free}$ encoders of memories 4, 6, and 10, in addition to an ODP feed-forward systematic encoder, denoted ODP-FFS. The $P_{\rm ev}$ estimates are almost identical for the same M, and in fact the traces of the encoders resemble the intertwined strands of a rope. We claim this is because the early ranges of the distance profiles are very similar.

The same progression occurs for these encoders in the counting estimate, as shown in Fig. 6, which treats $M = 4, 8, 16, 32$. Occasionally, the ordering of a single pair of encoders reverses at a particular M (e.g., at $M = 4$, the order is $\rm acbde$). Overall, however, the pattern is clear: A nested set of distance profiles leads to a nested set of $P_{\rm ev}$ curves.

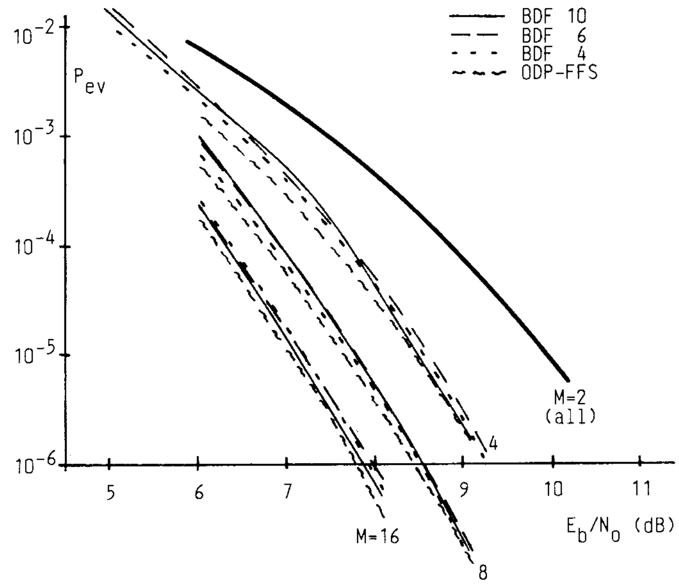

Fig. 5. Counting estimate of the event error probabilities for a set of encoders with different memories but with similar early ranges of the distance profiles.

BDF10	$= G_{51} = (4672\ 7542),$	$m = 10,$	$d_{\rm free} = 14;$
BDF6	$= G_{52} = (554\ 744),$	$m = 6,$	$d_{\rm free} = 10;$
BDF4	$= G_{53} = (46\ 72),$	$m = 4,$	$d_{\rm free} = 7;$
ODP-FFS	$= G_{54} = (4000\ 7153),$	$m = 11,$	$d_{\rm free} = 9.$

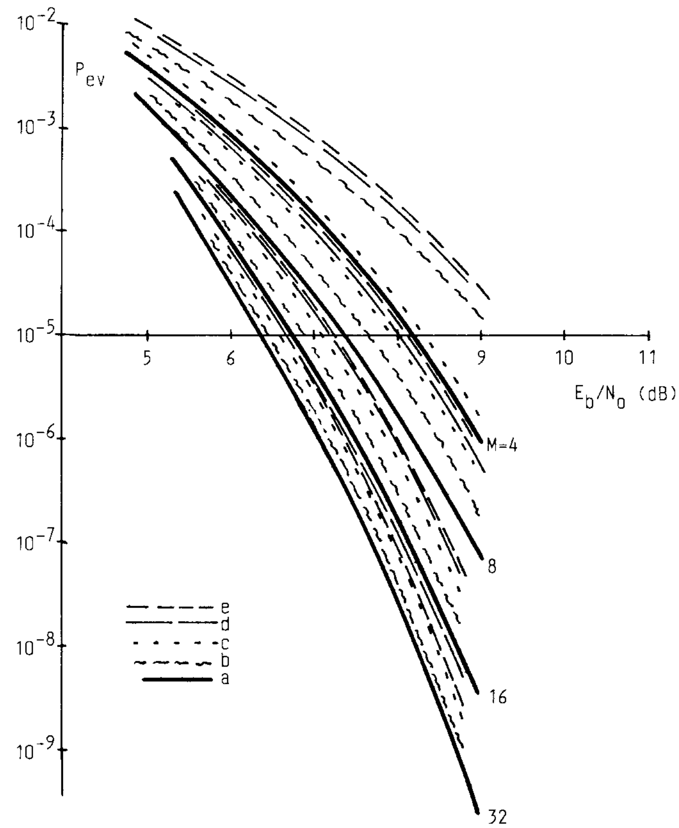

Fig. 6. Counting estimate of the event error probabilities for the same encoders as in Fig. 3.

$a = G_{31},\ b = G_{32},\ c = G_{33},\ d = G_{34},$ and $e = G_{35}.$

IV. Comparison of the M and Viterbi Algorithms

The object of a reduced-search decoder is to reduce the survivor number—and hopefully, the computation—for the same error perfor-

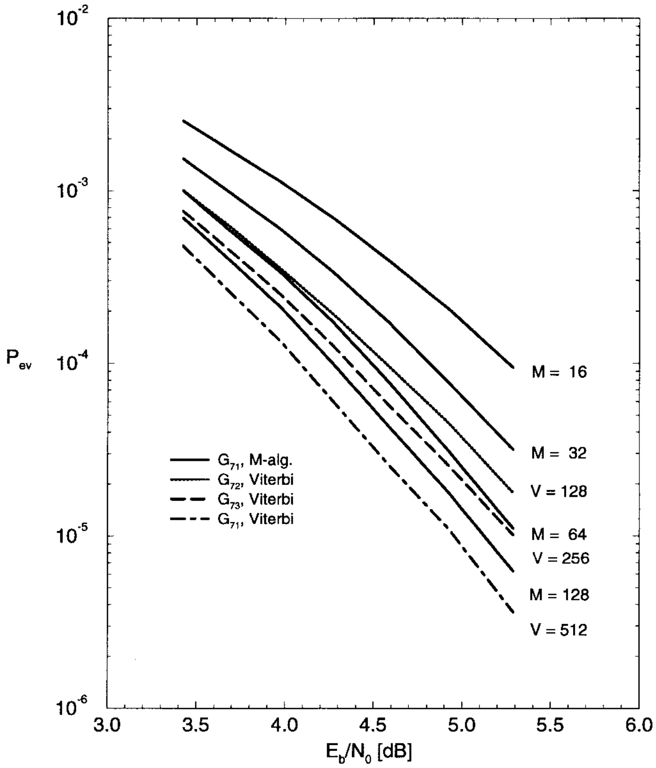

Fig. 7. Observed event error probabilities for a nonsystematic encoder G_{71} with the M-algorithm and three different best-$d_{\rm free}$ nonsystematic encoders G_{71}, G_{72}, G_{73} with Viterbi decoders.

$G_{71} = (4734\;6624),\;\; m = 9,\;\; d_{\rm free} = 12;$
$G_{72} = (712\;476),\;\; m = 7,\;\; d_{\rm free} = 10;$
$G_{73} = (561\;753),\;\; m = 8,\;\; d_{\rm free} = 12.$

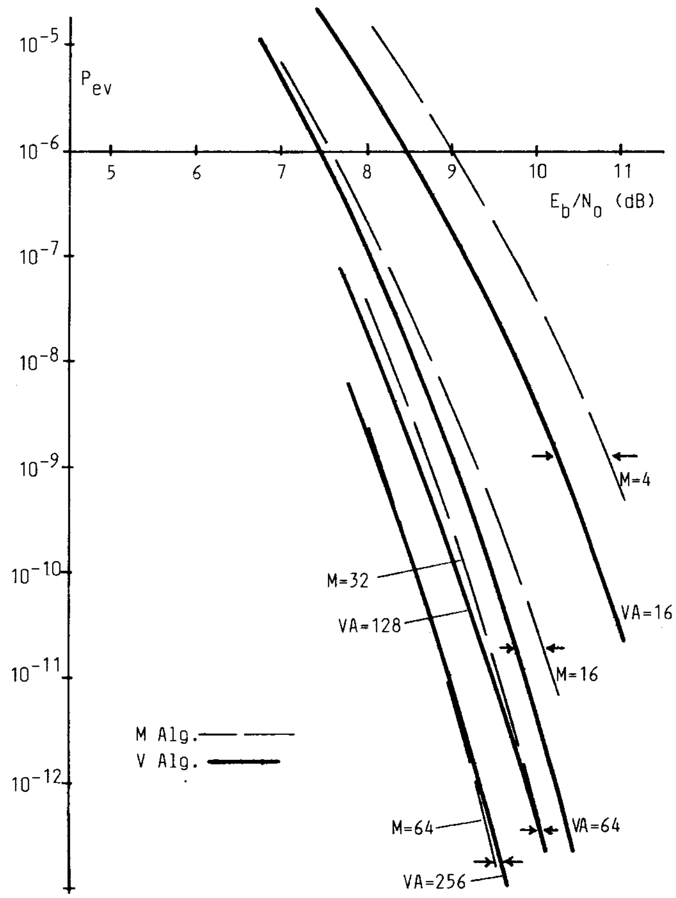

Fig. 8. Counting estimates of the event error probabilities for the M-algorithm decoding of the ODP feed-forward systematic encoder G_{81} with $M = 4, 16, 32$, and 64 and for the four-fold larger Viterbi decoder of sizes $16, 64, 128$, and 256. The Viterbi algorithm (VA) encoders are ODP feed-forward nonsystematic with memory $m = 4, 6, 7$, and 8. The quarter-size M-algorithm has lower error probabilities above VA size 128.

$G_{81} = (400000\;671166),\;\; m = 17,\;\; d_{\rm free} = 12;$
$G_{82} = (62\;56),\;\; m = 4,\;\; d_{\rm free} = 7;$
$G_{83} = (634\;564),\;\; m = 6,\;\; d_{\rm free} = 10;$
$G_{84} = (626\;572),\;\; m = 7,\;\; d_{\rm free} = 10;$
$G_{85} = (751\;557),\;\; m = 8,\;\; d_{\rm free} = 12.$

mance. We have shown that $P_{\rm ev}$ for a decoder of the M-algorithm type will depend almost entirely on the distance profile, but we have not shown how much better it might be than the Viterbi algorithm (VA) for a given number of survivors. Actually, the answer depends strongly on whether the error criterion is bounded distance or error rate.

We have already explored the BDD criterion for all rates in an earlier paper [15]. At rate $1/2$, for example, convolutional codes that correct e or fewer errors closely follow the rule $M \approx (1+\sqrt{2})^e \approx (2.414)^e$, while the VA requires state size $\approx 4^e$; i.e., the disparity in survivors grows exponentially as $\approx (1.7)^e$.

If the decoder is to be judged by its $P_{\rm ev}$, the disparity turns out to be much less. At very large E_b/N_0, the comparison must track the BDD result, since the lightest weight uncorrected error sequence will dominate the $P_{\rm ev}$ calculation. At relatively high $P_{\rm ev}$, we show the test results in Fig. 7. Nonsystematic best-$d_{\rm free}$ encoders are compared; the M-algorithm decodes the $m = 9$ encoder at $M = 16, 32, 64, 128$, while the VA works with the three encoders of memory 7, 8, and 9, whose survivor numbers are 128, 256, and 512. Some study of the figure will show that the VA requires somewhat more than twice the survivors for the same $P_{\rm ev}$.

Fig. 8 aims at relatively low $P_{\rm ev}$ and is based on the counting estimate. The VA error event probability is computed from [3, p. 223]

$$P_{\rm ev,va} \triangleq \sum_{d \geq d_{\rm free}} \alpha(d)[4p(1-p)]^{d/2} \tag{11}$$

in which $\alpha(d)$ is the number of paths of weight d in the trellis. The figure works by comparing the VA with nonsystematic ODP encoders to an M-algorithm (MA) that retains $1/4$ the survivors; the MA works always with the $m = 16$ ODP systematic encoder $(400000, 671166)$. For the combination (VA size 16)/(MA size 4), for which the VA uses an $m = 4$ ODP encoder, the VA is 0.5 dB better. For the combination (VA size 64)/(MA size 16), the VA gain drops to about 0.2 dB. By the time (VA size 256)/(MA size 64) is reached, the MA is actually a little better. Note that all the codes in the figure have nearly the same early distance profile.

As one would expect, we are seeing here that the VA/MA comparison begins to feel the influence of the bounded distance case as $P_{\rm ev}$ drops. We can predict that the VA will require 2–4 times the survivors that the M algorithm does, depending on the decoder error rate. It is widely acknowledged that the MA executes 1.5–2 times the calculations of the VA per survivor kept, so that this rate of 2–4 needs to be reduced somewhat. We can claim that overall the MA has up to half the cost of the VA, when $P_{\rm ev}$ is the criterion.

V. The Path Loss Issue

With the probability of error design criterion, more errors than are guaranteed corrected by the list size M may occur sometimes, or even often. These errors may lead to loss of the correct path from

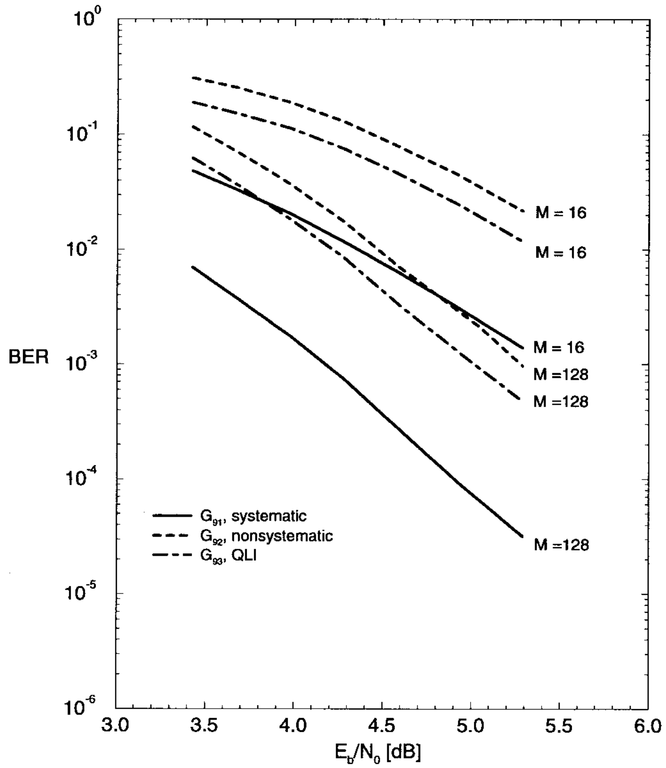

Fig. 9. Bit error probabilities observed in tests of the three types of encoders, all of memory $m = 20$.

$$G_{91} = G_{11},\ G_{92} = G_{12}, \text{and } G_{93} = G_{13}.$$

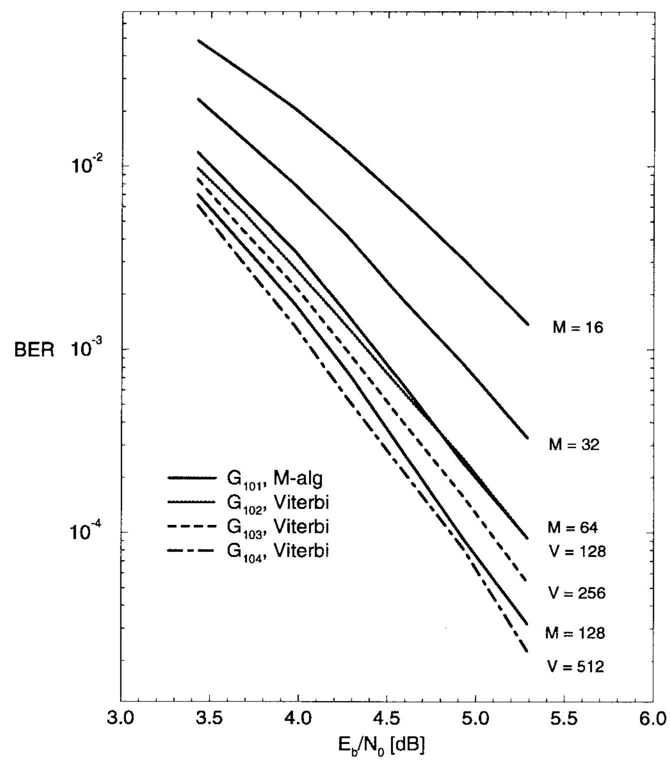

Fig. 10. Comparisons of the bit error probabilities observed indecoder tests.

$$\begin{aligned} G_{101} &= G_{11}; \\ G_{102} &= (712\,476), \quad m = 7,\ d_{\rm free} = 10; \\ G_{103} &= (561\,753), \quad m = 8,\ d_{\rm free} = 12; \\ G_{104} &= (4734\,6624),\ m = 9,\ d_{\rm free} = 12. \end{aligned}$$

the decoder memory, an event that the decoder must deal with. Tests show that the overall bit-error rate of an M-algorithm decoder with a nonsystematic encoder is much worse than the same decoder with a systematic encoder having a similar distance profile. Yet the error event rate is almost identical. The reason is that once the error event occurs the rest of the frame almost always follows an incorrect trellis path. This is the correct-path loss problem. Tests show that with systematic encoders the M decoder quickly recovers a lost correct path, and can even outperform the Viterbi decoder that works with a matching nonsystematic encoder. An example is shown in Fig. 9, which compares the BER of systematic, nonsystematic, and QLI encoders at $M = 16$ and 128. All have approximately the same initial distance profile. The free distance of the systematic encoder is by far the least, yet its BER is more than ten times better.

Fig. 10, which is a companion figure to Fig. 7, compares the M and Viterbi decoders in terms of the BER measure. It shows that the M algorithm is again 2–4 times more efficient than the Viterbi algorithm in terms of survivor numbers, just as it was against the $P_{\rm ev}$ measure, but only when the encoder is systematic.

Fig. 11 shows both the $P_{\rm ev}$ and BER for some systematic and nonsystematic rate $R = 2/3$ encoders whose memories and early distance profiles are similar (the nonsystematic encoder has much larger $d_{\rm free}$). As we saw with rate $R = 1/2$, the $P_{\rm ev}$'s for both encoders are almost identical at the same M, but the BER's are grossly different. Only the systematic encoder leads to an acceptable decoder BER.

Many other demonstrations of the superiority of feed-forward systematic encoders may be seen in [21]. Apparently, the only advantage of nonsystematic (and feedback) encoders is the larger $d_{\rm free}$ they offer at a given encoder state-space size. Yet this extra distance seems to have almost no effect on $P_{\rm ev}$ or the BER. Nor does it change the M needed to correct e errors, so long as e falls within the powers of the systematic encoder.

A suggestion of why systematic encoders offer rapid recovery of a lost correct path may be found by considering the trellises of rate $R = 1/2$ random systematic and nonsystematic encoders. Suppose the correct path is the all-zero one and no errors occur for a time, and consider an arbitrary trellis node. The 0-branch (the one that inserts a zero into the encoder shift register) is the one leading back to the correct path. For a systematic encoder, the distance increment of this "correct" branch is 0.5 on the average, while the incorrect branch has increment 1.5. For a nonsystematic encoder, these average increments are both 1, and give no particular direction of the search back to the correct path. Finally, we should use a feed-forward, rather than a feedback, systematic encoder, because a driving sequence of zeroes does not in general lead back to state 0 with the feedback encoders.

VI. Conclusion

We have given strong evidence that only feed-forward systematic encoders should be used with reduced-search decoders. To review the train of logic, we first showed that nonbacktracking decoder error performance depends almost entirely on the number of survivors kept and on the early part of the distance profile. We then compared tests of systematic and nonsystematic encoders with the same distance profile, and showed that because the systematic encoder allows quick recovery of a lost correct path, it has a much better overall error rate than a nonsystematic encoder has. Since both kinds of encoders have the same error rate in the absence of correct-path loss, systematic encoders are clearly superior to nonsystematic ones.

We have also reached other conclusions. Apparently, the survivor numbers required in a bounded-distance decoder also depend only

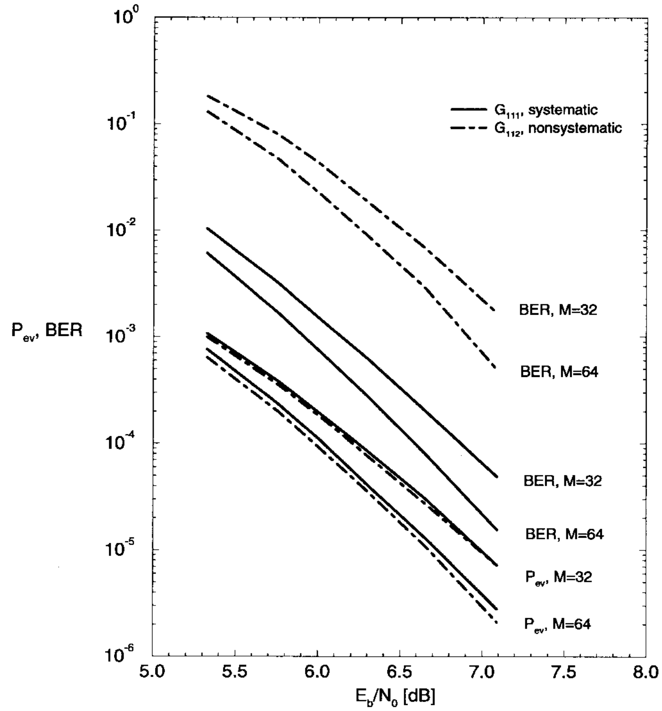

Fig. 11. Observed bit and event error probabilities for $R = 2/3$ systematic and nonsystematic ODP encoders.

$$G_{111} = \begin{pmatrix} 40000 & 00000 & 57371 \\ 00000 & 40000 & 63225 \end{pmatrix}, m = 14, \nu = 28, d_{\text{free}} = 8;$$
$$G_{112} = \begin{pmatrix} 51630 & 25240 & 42050 \\ 05460 & 61234 & 44334 \end{pmatrix}, m = 12, \nu = 23, d_{\text{free}} = 16.$$

very weakly on free distance, encoder state-space size, and the precise digits of the code generators; instead, they are simply and directly related to the desired error correction.

While we have studied only the performance of the M algorithm in this correspondence, the estimates of the maximum BDD storage size such as (5) are known to hold for any breadth-first decoder. In addition, our conclusions about distance profiles in Sections II and III are known to apply to backtracking decoders [22]. We can thus conjecture that our conclusions extend quite widely.

The conclusions here do run counter to much accepted wisdom about convolutional codes. For instance, much of the search for good codes has focused on maximizing the free distance for a given encoder memory. Yet we show that the free distance has almost no effect on the performance of a reduced decoder with a fixed M survivors, so long as the distance exceeds a certain minimum. Second, the widespread use of the Viterbi decoder has motivated the design of codes with the smallest possible encoder state spaces, but the complexity of a reduced-search decoder for a given error performance depends hardly at all on the encoder state-space size, again so long as it exceeds a certain minimum. We have shown that the BER of an M-algorithm decoder is several times better than a Viterbi algorithm, for the same decoder complexity. There is thus little practical reason to search at length for encoders, e.g., feedback encoders, with a minimal encoder state space or with the absolute largest free distance. What really matters is the early distance profile, and the search for a good profile can concentrate on systematic feed-forward encoders only.

While we have tested only a handful of code rates, the mechanism of how decoders perform seems clear and not really difficult. With just a little prudence in the encoder design, error performance simply depends on how many survivors are stored in the decoder.

References

[1] R. Johannesson, "Robustly optimal rate one-half binary convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 464–468, July 1975.

[2] J. L. Massey and D. J. Costello, Jr., "Nonsystematic convolutional codes for sequential decoding in space applications," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 806–813, Oct. 1971.

[3] J. B. Anderson and S. Mohan, *Source and Channel Coding*. Boston, MA: Kluwer, 1991.

[4] K. Sh. Zigangirov and V. D. Kolesnik, "List decoding of trellis codes," *Probl. Contr. Inform. Theory*, no. 6, 1980.

[5] S. J. Simmons and P. Wittke, "Low complexity decoders for constant envelope digital modulation," in *Conf. Rec. Global Communication Conf.* (Miami, FL, Nov. 1982), pp. E7.7.1–E7.7.5.

[6] T. Aulin, "Study of a new trellis decoding algorithm and its applications," Final Rep., ESTEC Contract 6039/84/NL/DG/, European Space Agency, Noordwijk, The Netherlands, Dec. 1985.

[7] K. Balachandran, "Design and performance analysis of constant envelope and non-constant envelope digital phase modulation schemes," Ph.D. dissertation, ECSE Dept., Rensselaer Polytech. Inst., Troy, NY, Feb. 1992.

[8] T. Palenius, "On reduced complexity noncoherent detectors for continuous phase modulation," Ph.D. dissertation, Telecommunication Theory Dept., Lund Univ., Lund, Sweden, Nov. 1991.

[9] N. Seshadri and J. B. Anderson, "Decoding of severely filtered modulation codes using the (M, L) algorithm," *J. Select. Areas Commun.*, vol. 7, pp. 1006–1016, Aug. 1989.

[10] F. Gozzo, "Robust sequence estimation in the presence of channel mismatch," Ph.D. dissertation, ECSE Dept., Rensselaer Polytech. Inst., Troy, NY, May 1992.

[11] T. Aulin, "Optimal trellis decoding at given complexity," in *Conf. Proc., 1993 IEEE Int. Symp. Information Theory* (San Antonio, TX, Jan. 1993), p. 272.

[12] ——, "Application of the SA(B) detection algorithm to codes-PSK modulation," Final Rep., ESTEC Contract 7108/87/NL/PR, European Space Agency, Noordwijk, The Netherlands, Dec. 1991.

[13] C.-F. Lin, "A truncated Viterbi algorithm approach to trellis codes," Ph.D. dissertation, ECSE Dept., Rensselaer Polytech. Inst., Troy, NY, Sept. 1986.

[14] J. B. Anderson and K. Balachandran, "Decision depths of convolutional codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 455–459, Mar. 1989.

[15] J. B. Anderson, "Limited search trellis decoding of convolutional codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 944–956, Sept. 1989.

[16] K. Sh. Zigangirov and H. Osthoff, "Analysis of global-list decoding for convolutional codes," *Europ. Trans. Telecommun.*, vol. 4, pp. 165–173, Mar.–Apr. 1993.

[17] R. Johannesson and K. Sh. Zigangirov, "Towards a theory for list decoding of convolutional codes," *Probl. Inform. Transm.*, no. 1, 1996.

[18] J. B. Anderson, "Sequential decoding based on an error criterion," *IEEE Trans. Inform. Theory*, vol. 38, pp. 987–1001, May 1992.

[19] R. Johannesson and Z.-x. Wan, "A linear algebra approach to minimal convolutional encoders," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1219–1233, July 1993.

[20] R. Johannesson and K. Sh. Zigangirov, "Distances and distance bounds for convolutional codes," in *Topics in Coding Theory—In Honour of Lars H. Zetterberg*. Berlin, Germany: Springer-Verlag, 1989, pp. 109–136.

[21] H. Osthoff, "Reduced complexity decoding with systematic encoders," Ph.D. dissertation, Information Theory Dept., Lund Univ., Lund, Sweden, May 1993.

[22] P. R. Chevillat and D. J. Costello, Jr., "Distance and computation in sequential decoding," *IEEE Trans. Commun.*, vol. 44, pp. 440–447, Apr. 1996.