General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

NSG-5025

(NASA-CR-147974) CONCATENATED CODING SYSTEMS EMPLOYING A UNIT-MEMORY CONVOLUTIONAL CODE AND A BYTE-ORIENTED DECODING ALGORITHM (Notre Dame Univ.) 35 p HC $4.00 CSCL 09B G3/61

N76-31932

Unclas 02450

Dame, Notre Dame,

ts Universal General-
pp. 432-436, July 1976.
eport was sent to you

n," IEEE Trans. Infor-
A preprint of this
ou on August 1, 1975.].

rs,

sey
vestigator

SEP 1976
RECEIVED
NASA STI FACILITY
INPUT BRANCH

Department of

ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

Concatenated Coding Systems Employing a Unit-Memory Convolutional Code and a Byte-Oriented Decoding Algorithm*

Technical Report No. EE-769

Lin-nan Lee**
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

July 12, 1976

ABSTRACT

Concatenated coding systems utilizing a convolutional code as the inner code and a Reed-Solomon code as the outer code are considered. In order to obtain very reliable communications over a very noisy channel with relatively small coding complexity, it is proposed to concatenate a byte-oriented unit-memory convolutional code with an RS outer code whose symbol size is one byte. It is further proposed to utilize a real-time minimal-byte-error probability decoding algorithm, together with feedback from the outer decoder, in the decoder for the inner convolutional code. The performance of the proposed concatenated coding system is studied, and the improvement over conventional concatenated systems due to each additional feature is isolated.

* This research was supported by the National Aeronautics and Space Administration under NASA Grant NSG 5025 at the University of Notre Dame in liaison with the Goddard Space Flight Center. This paper forms part of a dissertation submitted to the Graduate School of the University of Notre Dame, in partial fulfillment of the requirements for the Ph.D. degree.
** The author is presently with the LINKABIT Corporation, San Diego, Calif.

I. INTRODUCTION

The complexity of conventional coding systems grows exponentially with the block length of block codes (or with the constraint length of convolutional codes). To circumvent the prohibitive complexity of directly using very long codes, the idea of cascading two or more codes of less complexity to achieve highly reliable communications was considered first by Elias [1], and later by Forney [2]. Forney's technique of using two or more block codes over different alphabets to obtain a very low error rate over noisy channels is known as concatenated coding.

Guided by the premise that a convolutional code generally performs better than a block code of the same complexity, Falconer [3], and later Jelinek and Cocke [4], considered cascading an outer block code with an inner convolutional code. Figure 1 shows a general representation of such a block-convolutional concatenated coding system. In both the Falconer and Jelinek-Cocke schemes, sequential decoding was used for the inner decoder; the outer block coding system was used only to intervene when the sequential decoder experienced computational overflow. Therefore, these systems can be regarded, more or less, as primarily sequentially-decoded convolutional coding systems.

Maximum likelihood (i.e., Viterbi [5]) decoding of convolutional codes with a moderate constraint length can provide an error rate of less than 10^{-2} at a rate slightly higher than R_{comp} of the noisy channel. Forney's work [2] suggested that a concatenated coding system with a powerful outer code can perform reasonably well when its inner decoder is operated with a probability of error in the range between 10^{-2} and 10^{-3}. It was natural then for Odenwalder [6] to choose a Viterbi decoder for the inner coding system in his block-convolutional concatenated coding system.

Because the output error patterns of Viterbi-type decoders for convolutional codes are bursty, block codes over a large alphabet, such that many bits of the inner code form one symbol of the outer code, appear very attractive for the outer coding system. The Reed-Solomon (RS) block codes are particularly appealing because they can be decoded by relatively simple procedures (such as the Berlekamp-Massey [7], [8] algorithm) and have optimum distance properties. Because the lengths of the bursts of output errors made by Viterbi decoders are widely distributed, it is generally necessary to interleave the inner convolutional code so that errors in the individual RS-symbols of one block are independent; otherwise, a very long block code would be required to operate the system efficiently. Because the most likely length of the output error patterns made by the inner decoder are on the order of the constraint length, K, of the convolutional code, Odenwalder chose the RS symbol alphabet to be $GF(2^K)$.

In a block-convolutional concatenated coding system such as Odenwalder's employing a Viterbi decoder with conventional convolutional codes, it is very unlikely that the beginning of a decoding error burst is always aligned with the boundary between two RS symbols; in fact, such a burst only two bits long may affect two RS symbols. This fact led us to consider using good convolutional codes which are symbol-oriented rather than bit-oriented. In [9], we reported a class of unit-memory convolutional codes for which k_o-bit information segments are encoded into n_o-bit encoded segments. It was shown there that an (n_o, k_o) convolutional code with unit memory always achieves the largest free distance possible for codes of the same rate k_o/n_o and the same number 2^{Mk_o} of encoder states, where M is the encoder memory. The unit-memory codes are naturally byte-oriented with byte size equal to k_o information bits. It will be shown that the improved free

distance and the symbol-oriented nature of these codes provides an improvement of approximately 0.3db in the overall performance of the concatenated coding system when these codes replace bit-oriented convolutional codes.

Another improvement is to modify the decoder for the convolutional code so that the decoder emits not only the most-likely estimated symbol, but also reliability information about the estimated symbol. The outer decoder may then use this reliability information to perform either "erasures-and-errors" decoding or "generalized-minimum-distance" (GMD) decoding as suggested by Forney [2]. Zeoli [10] and Jelinek [11] proposed to extract reliability information by annexing a long tail to the original convolutional code and using this added tail to provide an error detection capability for the estimate made by the Viterbi decoder for the original shorter convolutional code. This approach requires the feedback of symbols previously decoded by the Viterbi decoder and, more importantly, uses the output of the outer decoder to restart the inner Viterbi decoder whenever an error is corrected by the outer decoder. It will be shown that the error detecting capability used with an "erasures-and-errors" outer decoder provides an improvement of 0.2 db and that the feedback from the outer decoder further improves the performance by 0.3 db.

An alternative approach to extracting reliability information from the inner decoder is to compute the a posteriori probability of correctness for each decoded symbol from the decoder for the short constraint length convolutional code and then use this probability as the reliability information provided to the outer coding system. It will be shown that, when used with an errors-and-erasures outer decoder, this scheme improves performance by only 0.05 dB to 0.1 dB compared to hard-decision decoding and hence is less powerful than Zeoli's tail annexation scheme; yet its performance is

undoubtedly optimal among all schemes employing only the short constraint length convolutional code (with no annexed tail). However, it will be shown that, in conjection with the use of feedback from the outer decoder, the a posteriori probability inner decoder provides about 0.2 db more improvement than does the Viterbi decoder aided by feedback. In fact, the a posteriori inner decoder, used with feedback from the outer decoder, offers a slight improvement over Zeoli's scheme; moreover the inner encoder and the inner decoder have the same constraint length so that the inner decoder generally and automatically returns to normal operation only a few branches after making an error.

The plan of this paper is as follows. In Section II, a "real time" decoding algorithm for unit-memory convolutional codes is developed which calculates the a posteriori probability for each value of the byte being decoded. In Sections III, IV, and V, the performances of several block-convolutional concatenated coding systems having unit-memory convolutional inner codes are compared with similar systems having conventional bit-oriented convolutional inner codes. In each case, we chose the (18,6) unit-memory convolutional code as the inner code because it has practically minimum complexity in terms of decoder implementation, and because of its reasonably large free distance (d_{free}=16). We chose the Reed-Solomon codes over GF(2^6), with block length 63 symbols, as the outer codes so that the symbol size of the RS codes would be matched to the byte-size (six bits) of the unit-memory code. In Section VI, the degradation of performance, when the rate 1/3 inner convolutional code is replaced by a rate 1/2 convolutional code, is considered in order to demonstrate the tradeoff between bandwidth expansion and signal-energy-to-noise ratio. In Section VII, the 95% confidence intervals for the simulation results are obtained and interpreted.

II. REAL-TIME MINIMAL-BYTE-ERROR PROBABILITY DECODING OF UNIT-MEMORY CODES

We now develop an algorithm for real-time minimal-byte-error probability decoding of the unit-memory convolutional codes described in [9].

Let $\underline{a}_t$ $(t = 1,2,\ldots)$ denote the byte (or subblock) of k_o information bits to be encoded at time t, and let $\underline{b}_t$ $(t = 1,2,\ldots)$ be the corresponding encoded subblock of n_o bits. For a unit-memory code,

$$\underline{b}_t = \underline{a}_t G_0 + \underline{a}_{t-1} G_1 \tag{1}$$

where G_0 and G_1 are $k_o \times n_o$ matrices and where, by way of convention, $\underline{a}_0 = \underline{0}$. We assume that the sequence $\underline{b}_1, \underline{b}_2, \ldots$ has been transmitted over a discrete memoryless channel and that $\underline{r}_t$ $(t = 1,2,\ldots)$ is the received subblock corresponding to the transmitted subblock $\underline{b}_t$. We shall write $\underline{a}_{[t,\ t']}$ to denote $[\underline{a}_t, \underline{a}_{t+1}, \ldots \underline{a}_{t'}]$; similarly for $\underline{b}_{[t,\ t']}$ and $\underline{r}_{[t,\ t']}$.

By <u>real-time decoding with delay</u> Δ, we mean that the decoding decision for $\underline{a}_t$ is made from the observation of $\underline{r}_{[1,\ t+\Delta]}$. The real-time minimal-byte-error probability (RTMBEP) decoding rule then is that which chooses its estimate $\hat{\underline{a}}_t$ as that value of $\underline{a}_t$ which maximizes $P(\underline{a}_t|\underline{r}_{[1,\ t+\Delta]})$ for $t = 1,2, \ldots$. To find a recursive algorithm for this decoding rule, we begin by noting that

$$P(\underline{a}_t|\underline{r}_{[1,t+\Delta]}) = P(\underline{a}_t, \underline{r}_{[1,t+\Delta]})/\sum_{\underline{\alpha}} P(\underline{\alpha}, \underline{r}_{[1,t+\Delta]}) \tag{2}$$

where we have used $\underline{\alpha}$ to denote a running variable for $\underline{a}_t$. It suffices then to find a recursive method for calculating $P(\underline{a}_t, \underline{r}_{[1,\ t+\Delta]})$.

We next observe that

$$\begin{aligned} P(\underline{a}_t|\underline{r}_{[1,t+\Delta]}) &= P(\underline{a}_t, \underline{r}_{[1,t]})P(\underline{r}_{[t+1,\ t+\Delta]}|\underline{a}_t, \underline{r}_{[1,t]}) \\ &= P(\underline{a}_t, \underline{r}_{[1,t]})P(\underline{r}_{[t+1,\ t+\Delta]}|\underline{a}_t) \end{aligned} \tag{3}$$

where we have used the facts that the channel is memoryless and that the code has unit memory. It remains to find recursive rules for obtaining the two probabilities on the right in (3).

Obtaining the recursion for $P(\underline{a}_t, \underline{r}_{[1,t]})$ is quite standard [12]-[14];

$$P(\underline{a}_t, \underline{r}_{[1,t]}) = \sum_{\underline{a}_{t-1}} P(\underline{a}_{[t-1,\ t]}, \underline{r}_{[1,t]})$$

$$= \sum_{\underline{a}_{t-1}} P(\underline{a}_{t-1}, \underline{r}_{[1,\ t-1]}) P(\underline{a}_t, \underline{r}_t | a_{t-1}, \underline{r}_{[1,\ t-1]}). \quad (4)$$

But also

$$P(\underline{a}_t, \underline{r}_t | \underline{a}_{t-1}, \underline{r}_{[1,\ t-1]}) = P(\underline{a}_t | \underline{a}_{t-1}, \underline{r}_{t-1}) P(\underline{r}_t | \underline{a}_{[t-1,\ t]}, \underline{r}_{[1,\ t-1]})$$

$$= 2^{-k_o} P(\underline{r}_t | \underline{b}(\underline{a}_{[t-1,\ t]})) \quad (5)$$

where we have written $\underline{b}(\underline{a}_{[t-1,\ t]})$ for the value of $\underline{b}_t$ determined by (1) from $\underline{a}_{[t-1,\ t]}$, and where we assume here and hereafter that all information sequences are equally likely (as corresponds to maximum-likelihood decoding.) Substituting (5) into (4), we have our desired recursion

$$P(\underline{a}_t, \underline{r}_t) = 2^{-k_o} \sum_{\underline{a}_{t-1}} P(\underline{a}_{t-1}, \underline{r}_{[1,\ t-1]}) P(\underline{r}_t | \underline{b}(\underline{a}_{[t-1,\ t]})). \quad (6)$$

We now turn to the quantity $P(\underline{r}_{[t+1,\ t+\Delta]} | \underline{a}_t)$ which we note is the $i = 1$ value of

$$P(\underline{r}_{[t+i,\ t+\Delta]} | \underline{a}_{t+i-1}) = \sum_{\underline{a}_{t+i}} P(\underline{a}_{t+i}, \underline{r}_{[t+i,\ t+\Delta]} | \underline{a}_{t+i-1}). \quad (7)$$

Proceeding in the same manner that (6) was obtained from (4), we find the desired recursion

$$P(\underline{r}_{[t+i,\ t+\Delta]} | \underline{a}_{t+i-1}) = 2^{-k_o} \sum_{\underline{a}_{t+i}} P(\underline{r}_{[t+i+1,t+\Delta]} | \underline{a}_{t+i}) P(\underline{r}_{t+i} | \underline{b}(\underline{a}_{[t+i-1,t+i]})). \quad (8)$$

This recursion is initialized with its $i = \Delta$ value

$$P(\underline{r}_{t+\Delta}|\underline{a}_{t+\Delta-1}) = \sum_{\underline{a}_{t+\Delta}} P(\underline{a}_{t+\Delta}, \underline{r}_{t+\Delta}|\underline{a}_{t+\Delta-1})$$

$$= 2^{-k_o} \sum_{\underline{a}_{t+\Delta}} P(\underline{r}_{t+\Delta}|\underline{b}(a_{[t+\Delta-1,\ t+\Delta]})), \tag{9}$$

and evaluated with $i = \Delta-1, \Delta-2, \ldots, 1$. It should be noted that, because of the restriction to unit-memory codes, the recursion (8) is much simpler than the corresponding one required for RTMBEP decoding of general convolutional codes [14].

An algorithm to carry out the recursive rules given by (6) and (8) requires, for each byte (or "state" in the usual Viterbi decoding terminology) $\underline{\alpha}$, the storage of two real numbers, $f(\underline{\alpha})$ and $h(\underline{\alpha})$; namely,

$$f(\underline{\alpha}) = P(\underline{a}_t = \underline{\alpha}, \underline{r}_{[1,t]}) \tag{10}$$

and

$$h(\underline{\alpha}) = P(\underline{r}_{[t+i,\ t+\Delta]}|\underline{a}_{t+i-1} = \underline{\alpha}) \tag{11}$$

where i will be decremented from Δ to 1 as the algorithm progresses. (Of course, the received segment $\underline{r}_{[t+1,\ t+\Delta]}$ must also be stored so that $P(\underline{r}_{t+i}|\underline{b}(\underline{a}_{[t+i-1,\ t+i]}))$ can also be found for $i = \Delta, \Delta-1, \ldots, 1$.) We may now state:

<u>The RTMBEP Decoding Algorithm for Unit-Memory Codes</u>

<u>Step 0</u>: Set $f(\underline{0}) = 2^{k_o}$ and set $f(\underline{\alpha}) = 0$ for $\underline{\alpha} \neq \underline{0}$. Set $t = 1$.

<u>Step 1</u>: Make the replacement, for all states $\underline{\alpha}$,

$$f(\underline{\alpha}) \leftarrow 2^{-k_o} \sum_{\underline{\alpha}'} f(\underline{\alpha}')\, P(\underline{r}_t|\underline{b}(\underline{\alpha}', \underline{\alpha})).$$

<u>Step 2</u>: Set $i = \Delta$ and, for all states $\underline{\alpha}$, set

$$h(\underline{\alpha}) = 2^{-k_o} \sum_{\underline{\alpha}'} P(r_{t+\Delta}|b(\underline{\alpha}, \underline{\alpha}')).$$

<u>Step 3</u>: Decrease i by 1 and make the replacement, for all states $\underline{\alpha}$,

$$h(\underline{\alpha}) \leftarrow 2^{-k_o} \sum_{\underline{\alpha}'} h(\alpha')P(\underline{r}_{t+i}|\underline{b}(\underline{\alpha}, \underline{\alpha}')).$$

If now $i = 1$, go to Step 4. Otherwise, return to Step 3.

<u>Step 4</u>: Emit, as the estimate of $\underline{a}_t$, that byte $\underline{\alpha}_o$ which maximizes $f(\underline{\alpha})h(\underline{\alpha})$, and emit, as the reliability indicator, the probability

$$P(\underline{a}_t = \underline{\alpha}_o|\underline{r}_{[1,\ t+\Delta]}) = f(\underline{\alpha}_o)h(\underline{\alpha}_o)/\sum_{\underline{\alpha}} f(\underline{\alpha})h(\underline{\alpha}).$$

Increase t by 1 and return to Step 1.

The only feature of the algorithm that should require any comment is the initialization of $f(\underline{0})$ at 2^{k_o}. This is required so that the first time step 1 is performed one obtains the correct initial value $f(\underline{\alpha}) = P(\underline{r}_1|\underline{b}(\underline{0}, \underline{\alpha}))$. In fact, however, it makes no difference in the output from the algorithm if the f and h values are scaled by fixed positive constants, so that $f(\underline{0}) = 1$ is permissible in Step 0 and the factors 2^{-k_o} can be removed in Steps 1, 2 and 3.

Note that Step 3 of the algorithm, which has the same complexity as Step 1, is performed $\Delta - 1$ times for each time that Step 1 is performed. It is clearly desirable then to keep Δ as small as possible. Table I shows the variation of the decoding byte-error probability, P_{BE}, with the decoding delay, Δ, for the ($n_o = 18$, $k_o = 6$) unit-memory code of [9] used on a simulated three-bit- quantized additive white Gaussian noise (AWGN) channel. We see that $\Delta = 8$ gives virtually the same P_{BE} as the "optimum" choice $\Delta = \infty$.

We now point out, however, that one can reduce the ratio of Step 3 operations to Step 1 operations to as close to unity as desired without any degradation in performance but at the cost of additional storage. The "trick" is to use a <u>variable decoding delay</u> Δ. Each $\underline{a}_t$ is decoded from

Table I. Variation of Decoding Byte-Error Probability p with Decoding Delay Δ for RTMBEP Decoding of the (18,6) Unit-Memory Code on a Simulated AWGN Channel with an E_b'/N_o of 1.25 db. (4000 bytes decoded for each Δ).

Δ (bytes)	4	6	8	16
p	.0285	.0248	.0193	.0193

Table II. Byte-Error Probability, p, for Viterbi Decoding of Three $R = k_o/n_o = 1/3$ Convolutional Codes on a Simulated AWGN Channel. (8000 bytes decoded for each point shown, decoding delay Δ in bits of 48 in all cases.)

E_b'/N_o	1.00 db	1.25 db	1.50 db	1.75 db
p(95% confidence) (18,6) unit-memory code	.0305 (±.0053)	.0200 (±.0044)	.0118 (±.0033)	.0065 (±.0025)
p(95% confidence) M = 6, (3,1) code	.0488 (±.0068)	.0325 (±.0056)	.0233 (±.0048)	.0128 (±.0035)
p(95% confidence) M = 7, (3,1) code	.0400 (±.0062)	.0225 (±.0047)	.0140 (±.0037)	.0103 (±.0032)

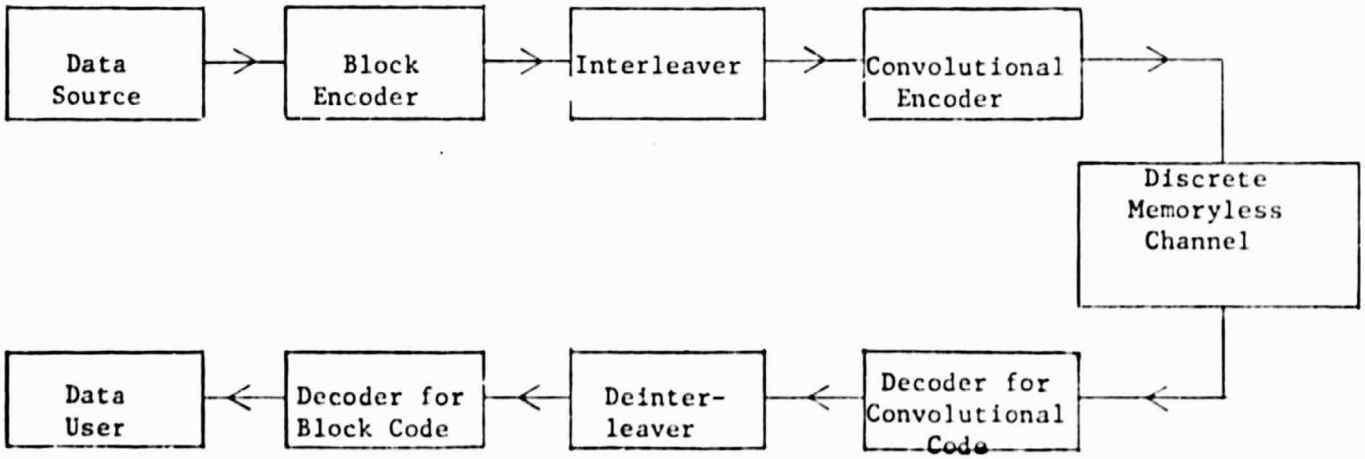

Figure 1. A concatenated coding system employing a convolutional code as the inner code and a block code as the outer code.

$P(\underline{a}_t|\underline{r}_{[1,\ t+\Delta]})$ but Δ, depending on the value of t, takes some value in the range $\Delta_m \leq \Delta \leq \Delta_M$. The minimum decoding delay, Δ_m, is chosen large enough to ensure negligible degradation, say $\Delta_m = 8$, while the maximum decoding delay, Δ_M, is chosen small enough to make the increased memory tolerable as will soon become apparent.

In this variable real-time minimal-byte-error probability (VRTMBEP) decoding, one stores $\Delta_M - \Delta_m + 2$ real numbers for each state $\underline{\alpha}$, namely: $f_i(\underline{\alpha})$ for $i = 1,2,\ldots, \Delta_M - \Delta_m + 1$ and $h(\underline{\alpha})$ where

$$f_i(\underline{\alpha}) = P(\underline{a}_{t+i-1} = \underline{\alpha},\ r_{[1,\ t+i-1]}) \tag{12}$$

and where $h(\underline{\alpha})$ is as in (11) with Δ replaced by Δ_M.

Observe now that, in the process of executing Step 2 of the RTMBEP algorithm with $\Delta = \Delta_M$, one would obtain sequentially the quantities

$$P(\underline{r}_{[t+i,\ t+\Delta_M]}|\underline{a}_{t+i-1} = \underline{\alpha}) \tag{13}$$

for $i = \Delta_M-1, \Delta_M-2, \ldots, 1$. But the product of the quantity in (13) with $f_i(\underline{\alpha})$ as in (12) is, according to (3), equal to $P(\underline{a}_{t+i-1} = \underline{\alpha}|\underline{r}_{[1,\ t+\Delta_M]})$; this is precisely the statistic needed to estimate $\underline{a}_{t+i-1}$ with a decoding delay of $\Delta = \Delta_M - i + 1$. Hence, if we had the foresight to perform Step 1 of the RTMBEP algorithm $\Delta_M - \Delta_m + 1$ times and to store the resulting $f_i(\underline{\alpha})$, then we could make $\Delta_M - \Delta_m + 1$ decoding decisions during the $\Delta_M - 1$ times that Step 3 is performed. Thus, for each time we use Step 1, we would be using Step 3 only $(\Delta_M-1)/(\Delta_M - \Delta_m + 1)$ times. For instance, with $\Delta_m = 8$ and $\Delta_M = 13$, we would perform Step 3 only twice for each time we performed Step 1; and we would be storing only $\Delta_M - \Delta_m + 2 = 7$ real numbers per state rather than 2 as in the original RTMBEP algorithm in which Step 3 is performed $\Delta - 1 = 7$ times for each time that Step 1 is performed.

It should now be obvious that the following algorithm is the necessary modification to the RTMBEP decoding algorithm for obtaining reduced computation

at the price of additional storage as has just been described.

The VRTMBEP Decoding Algorithm for Unit-Memory Codes

Step 0: Set $f_{\Delta_M - \Delta_{m+1}}(\underline{0}) = 2^{k_o}$ and set $f_{\Delta_M - \Delta_m+1}(\underline{\alpha}) = 0$ for $\underline{\alpha} \neq \underline{0}$.

Set $t = 1$.

Step 1: Set $f_1(\underline{\alpha}) = 2^{-k_o} \sum_{\underline{\alpha}'} f_{\Delta_M - \Delta_m+1}(\underline{\alpha}')P(\underline{r}_t|\underline{b}(\underline{\alpha}', \underline{\alpha}))$,

and set $f_{i+1}(\underline{\alpha}) = 2^{-k_o} \sum_{\underline{\alpha}'} f_i(\underline{\alpha})P(\underline{r}_{t+i}|\underline{b}(\underline{\alpha}', \underline{\alpha}))$

for $i = 1,2,\ldots, \Delta_M - \Delta_m$ in order.

Step 2: Set $i = \Delta_M$ and, for all states $\underline{\alpha}$, set

$$h(\underline{\alpha}) = 2^{-k_o} \sum_{\underline{\alpha}'} P(\underline{r}_{t+\Delta_M}|\underline{b}(\underline{\alpha}, \underline{\alpha}')).$$

Step 3: Decrease i by 1 and make the replacement, for all states $\underline{\alpha}$,

$$h(\underline{\alpha}) \leftarrow 2^{-k_o} \sum_{\underline{\alpha}'} h(\underline{\alpha}')P(\underline{r}_{t+i}|\underline{b}(\underline{\alpha}, \underline{\alpha}')).$$

If now $i \leq \Delta_M - \Delta_m + 1$, go to Step 4. Otherwise, return to Step 3.

Step 4: Emit, as the estimate of $\underline{a}_{t+i-1}$, that byte $\underline{\alpha}_o$ which maximizes $f_i(\underline{\alpha})h(\underline{\alpha})$, and emit, as the reliability indicator, the probability $P(\underline{a}_{t+i-1} = \underline{\alpha}_o|\underline{r}_{[1,\ t+\Delta_M]}) = f_i(\underline{\alpha}_o)h(\underline{\alpha}_o)/\sum_{\underline{\alpha}} f(\underline{\alpha})h(\underline{\alpha})$.

If $i = 1$, increase t by $\Delta_M - \Delta_m + 1$ and return to Step 1.

Otherwise, decrease i by 1 and return to Step 3.

It is satisfying to note that the VRTMBEP decoding algorithm reduces to the RTMBEP algorithm when $\Delta_M = \Delta_m$. It should be pointed out that when only a finite number, L, of information bytes are encoded and one takes $\Delta_M = L$, the largest possible choice, then the VRTMBEP algorithm reduces to

that given by Bahl et al. [12] (when the latter algorithm is specialized to unit-memory codes) and does about twice the computation of the usual Viterbi decoder; but this case also maximizes the memory requirements. The chief advantage which both RTMBEP and VRTMBEP decoding of unit-memory codes have over Viterbi decoding is in their providing reliability information about the decoding decisions; information of considerable value to the outer decoder in a concatenated coding system.

Because the resulting performance of the RTMBEP and VRTMBEP algorithms are indistinguishable when $\Delta = \Delta_m$ is chosen large enough for negligible degradation compared to $\Delta = \infty$, say $\Delta_m = 8$, we will not hereafter distinguish between the two algorithms in our discussion of concatenated coding systems.

III. ODENWALDER'S CONCATENATED CODING SYSTEM AND SOFT-DECISION MODIFICATION WITH THE RTMBEP DECODING ALGORITHM

The concatenated coding system proposed by Odenwalder [6], which we shall call System I, is as shown in Figure 1 where the inner decoder is a hard-decision Viterbi decoder and where the outer decoder is a t-error correcting decoder for the RS outer block code. Here and hereafter, we assume that the interleaving is "perfect", i.e., that the symbols in each RS block at the output of the interleaver have been independently decoded by the inner Viterbi decoder. Thus, we can then calculate the probability of a decoding error in an RS block, P_{ERS}, as

$$P_{ERS} = \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}, \tag{14}$$

where n is the RS block length (in bytes) and p is the byte-error probability at the Viterbi decoder output. Further, since almost all the incorrectly decoded RS codewords are $d_{min} = 2t + 1$ symbols away from the correct codeword (where d_{min} is the minimum distance of the RS code), the byte-error

probability, P_{BE}, of the concatenated coding system is given closely by

$$P_{BE} = \frac{2t+1}{n} \; P_{ERS}. \tag{15}$$

For a byte size of 6 bits, as will be assumed hereafter, the RS code has length $n = 2^6 - 1 = 63$ bytes. For convenient reference, we give in Table II the byte-error probability of a Viterbi decoder for the three different convolutional codes of rate $R_{CON} = k_o/n_o = 1/3$ that will be used in our subsequent comparisons when used on four different AWGN channels; this data is taken from [9]. The AWGN channels are specified by the ratio of channel energy per encoder input bit to one-side noise power spectral density, E_b'/N_o. Note that the energy per channel input bit (decoder output bit), E_s, is given by $E_s = R_{CON}E_b'$. But also $E_b' = R_{RS}E_b$ where R_{RS} is the rate of the RS code and E_b is the channel energy per information bit entering the RS encoder. Thus, the channel energy per information bit to one-sided noise power spectral density ratio for the overall concatenated coding system, E_b/N_o is given by

$$E_b/N_o = \frac{1}{R_{RS}\,R_{CON}} \; (E_s/N_o). \tag{16}$$

Using the results of Table II together with (14) and (15), we can calculate the byte-error probability for Odenwalder's System I for various RS outer codes. The results of this calculation are shown in Fig. 2 for the three different $R_{CON} = 1/3$ convolutional codes, namely (i) the conventional (3,1) code with $M = 6$, i.e., $K = 7$; (ii) the conventional (3,1) code with $M = 7$, i.e., $K = 8$; and (iii) the (18,6) unit-memory code. Codes (i), (ii) and (iii) have free distances of 15, 16 and 16, respectively, and their corresponding Viterbi decoders have 64, 128 and 64 states, respectively. We see, from Fig. 2, that the use of the unit-memory code provides an advantage of about 0.3 db over the conventional code with the same state complexity, part of which gain is attributable to the larger free distance of the unit-

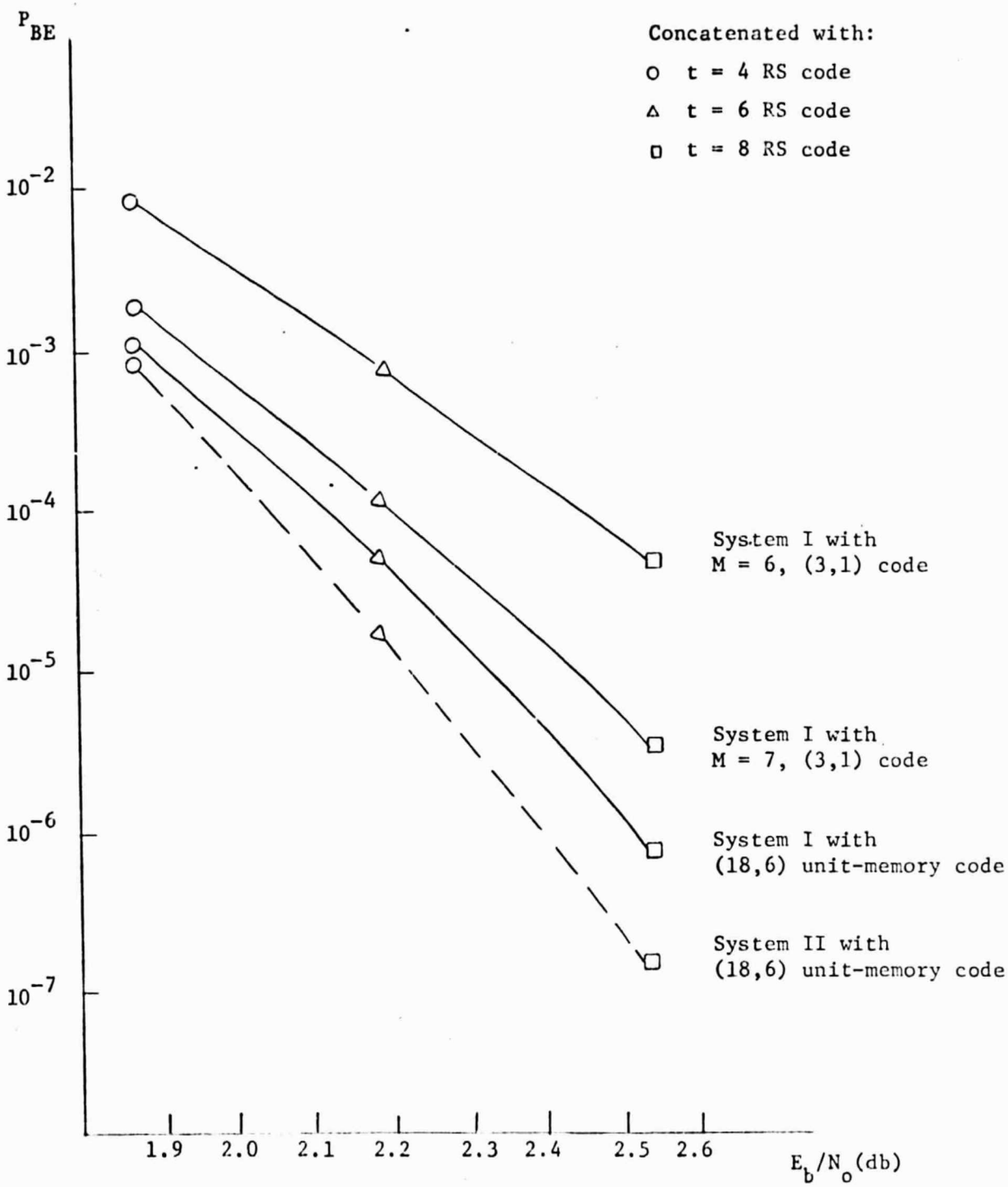

Figure 2. The performance of Concatenated Coding Systems I and II with RS codes over $GF(2^b)$ on a simulated AWGN channel with E_b'/N_o = 1.25 db.

memory code. But the unit-memory code is also about 0.1 db superior to the conventional code with the same free distance (and doubled number of decoder states); this gain is attributable entirely to the byte-oriented structure of the unit-memory code.

It should be mentioned that gains of 0.1 db are not insignificant in concatenated coding systems. As can be seen from Fig. 3, a gain of 0.1 db corresponds to a reduction of P_{BE} by nearly an order of magnitude, such steepness of the P_{BE} vs. E_b/N_o curves being characteristic of well-designed concatenated coding systems.

The inner decoder, i.e., the Viterbi decoder, in System I makes "hard decisions" on the decoded bytes. The system performance can be improved by using a "soft decision" decoder which passes along to the outer decoder a reliability indicator for each decoded byte. Such a system, in which the inner decoder is a RTMBEP decoder and the outer decoder is an errors-and-erasures decoder for the RS code, will be called System II. (For ease of reference, we summarize in Table III the characteristics of each of the six concatenated coding systems that will be considered in this paper.) When the reliability indicator, $P(\underline{a}_t | \underline{r}_{[1,t+\Delta]})$ for a decoded byte is less than some specified T, the outer decoder treats the byte as having been "erased." The erasures-and-errors decoder for the RS code can correct t errors and e erasures, whenever $2t + e < d_{min}$. Thus, the block error probability for the outer decoder is given by

$$P_{ERS} = \sum_{d=d_{min}}^{n} \sum_{\substack{t=0 \\ e=d-2t}}^{d/2} \binom{n}{t,e} p^t q^2 (1-p-q)^{n-t-e} \tag{17}$$

where p is again the byte-error probability for the inner decoder, where q is the byte-erasure probability for the inner decoder, and where

$$\binom{n}{t,e} = \frac{n!}{t!e!(n-t-e)!}.$$

Table III. The Six Bloc -Convolutional Concatenated Coding Systems Studied. (EO = errors only decoder, E + E = errors and erasures decoder, FBTID = feedback to inner decoder.)

	Inner Decoder Type	Outer Decoder Type	Inner Code Tail Annexation
System I	Viterbi hard-decision	EO	NO
System II	RTMBEP soft-decision	E + E	NO
System III	Viterbi hard-decision	EO with FBTID	NO
System IV	RTMBEP hard-decision	EO with FBTID	NO
System V	RTMBEP soft-decision	E + E with FBTID	NO
System VI	Viterbi soft-decision	E + E with FBTID	YES

Table IV. Variation of Inner Decoder Byte-Error Probability p and Byte-Erasure Probability q and of Outer Decoder Byte-Error Probability P_{BE} with the Erasure Threshold T for the (18,6) Unit-Memory Code on a Simulated AWGN Channel and with the Minimum Distance d_{min} of the Outer RS Code.

E_b'/N_o in db	T	p	q	P_{BE} for $d_{min} = 9$	P_{BE} for $d_{min} = 13$	P_{BE} for $d_{min} = 17$
1.00	.70	.01325	.04150	$.740\times10^{-2}$	$.477\times10^{-3}$	$.128\times10^{-4}$
"	.50	.02100	.01950	$.677\times10^{-2}$	$.555\times10^{-3}$	$.213\times10^{-4}$
"	.80	.00675	.03400	$.123\times10^{-2}$	$.244\times10^{-4}$	$.193\times10^{-6}$
1.25	.70	.00800	.02650	$.902\times10^{-3}$	$.179\times10^{-4}$	$.149\times10^{-6}$
"	.50	.01350	.01125	$.107\times10^{-2}$	$.332\times10^{-4}$	$.481\times10^{-6}$
"	.80	.00425	.02125	$.112\times10^{-3}$	$.691\times10^{-6}$	$.173\times10^{-8}$
1.50	.70	.00525	.01625	$.981\times10^{-4}$	$.684\times10^{-6}$	$.204\times10^{-8}$
"	.50	.00900	.00400	$.113\times10^{-3}$	$.136\times10^{-5}$	$.774\times10^{-8}$
"	.80	.00250	.01050	$.416\times10^{-5}$	$.636\times10^{-8}$	$.416\times10^{-11}$
1.75	.70	.00250	.00825	$.249\times10^{-5}$	$.334\times10^{-8}$	$.196\times10^{-11}$
"	.50	.00400	.00250	$.336\times10^{-5}$	$.816\times10^{-8}$	$.927\times10^{-11}$

The byte-error probability of the overall system is again obtained from (15).

The byte-error-probability, p, and the erasure probability, q, depend on the particular threshold, T, specified. The optimal threshold is a function of E_b'/N_o and the minimum distance, d_{min}, of the Reed-Solomon code. Roughly speaking, for a given block length n, as d_{min} gets larger, the overall block error probability is minimized at a higher erasure rate. We have found no simple way to determine the optimal threshold analytically. Instead, we have found p and q for T = 0.5, 0.7 and 0.8 by simulation and have used these values of p and q to calculate the byte-error probability of the coding system. In Table IV, we show the result of this calculation. We see, for E_b'/N_o in the range from 1.25 db to 1.75 db, that T = 0.7 is the best threshold among the three candidates.

The performance of System II with T = 0.7 is also plotted in Figure 2. The improvement over System I of the performance due to the erasure scheme, as observed from Figure 2, is dependent on the error correcting capability of the outer coding system as well as on E_b'/N_o and is approximately 0.1 dB. This slight improvement is probably not significant enough to justify the increased complexity of the RTMBEP decoder over the Viterbi decoder. However, as we shall soon see, the RTMBEP decoder coupled with an "erasures-and-error" block decoder performs much better than the Viterbi decoder <u>when feedback from the outer decoder is utilized</u>.

IV. FEEDBACK FROM THE OUTER DECODER TO THE INNER DECODER

Because of the nature of a convolutional code and the Viterbi decoding algorithm, once an "error event" occurs the decoder often makes a number of closely spaced erroneous estimations before it recovers to correct operation. Since the outer decoder of a concatenated coding system is designed in such a way that it is able to detect and correct almost all of the errors made by the inner decoder, it is then of significant advantage if the corrected estimates of the outer decoder are fed back to restart the inner decoder from the point where it first erred in order to eliminate the "burst" of errors. Figure 3 illustrates the general concept of such a block-convolutional concatenated coding system.

To study the gain provided by feedback from the outer decoder, we first implemented a software Viterbi decoder and a software RTMBEP decoder which can be restarted with feedback. Assuming that the outer decoder always makes correct decisions, a justifiable assumption since the probability of byte-error at the outer decoder output is at least several orders of magnitude less than that at the inner decoder's output, we obtained the results shown in Table V for the (18,6) unit-memory convolutional code on a simulated AWGN channel with an E_b'/N_o of 1.25 db. From Table V, we see that the RTMBEP decoder receives a considerably greater benefit from the feedback than does the Viterbi decoder. We then considered the following block-convolutional concatenated coding systems:

System III: A hard-decision Viterbi inner decoder with feedback from the errors-only RS outer decoder, i.e., System I with feedback.

System IV: A hard-decision RTMBEP inner decoder with feedback from the errors-only RS outer decoder.

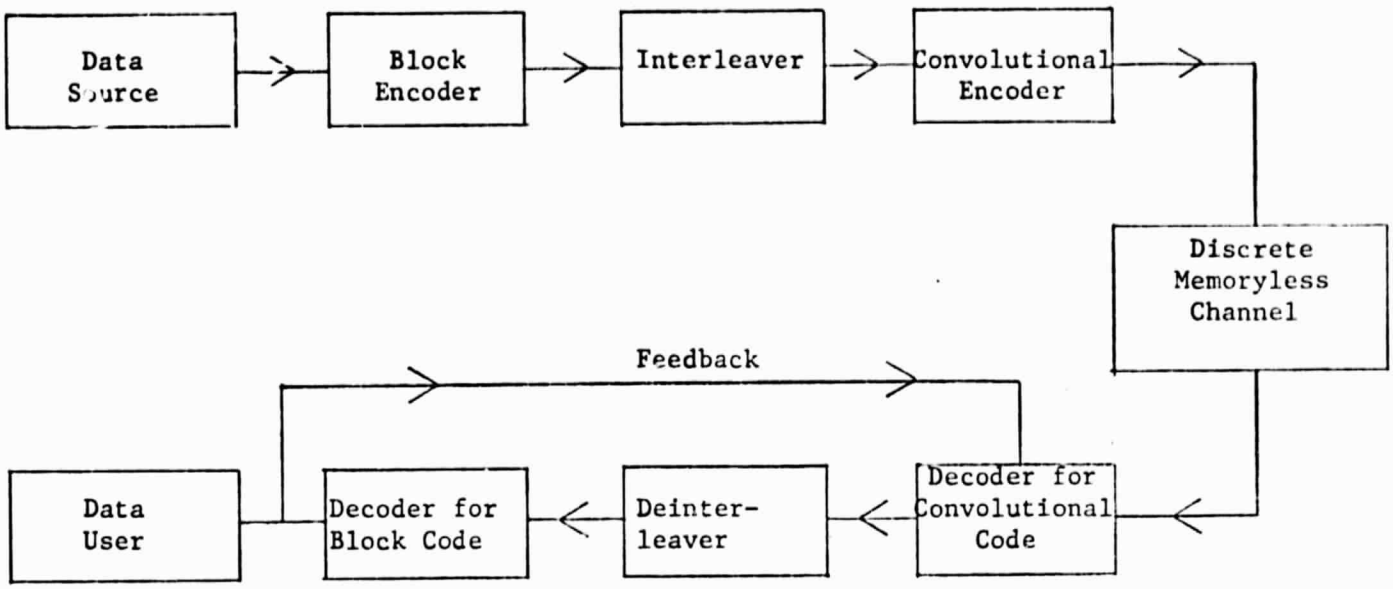

Figure 3. A block/convolutional concatenated coding system with feedback from the outer decoder to the inner decoder.

Table V. The Effect of Feedback from the Outer Decoder on the Byte-Error Probability for a Viterbi Decoder and an RTMBEP Decoder on a Simulated AWGN Channel with an E_b'/N_o of 1.25 db. (8000 bytes decoded for each point shown, decoding delay Δ of 48 bits in each case.)

	p for (18,6) unit-memory code (95% confidence)		p for M = 7, (3,1) code (95% confidence)	
	No feedback	With feedback	No feedback	With Feedback
Viterbi Decoder	.0200 ($\pm$.0032)	.0110 ($\pm$.0023)	.0225 ($\pm$.0034)	.0133 ($\pm$.0025)
RTMBEP Decoder	.0193 ($\pm$.0031)	.0075 ($\pm$.0019)		

System V: A soft-decision RTMBEP inner decoder with feedback from the erasures-and-errors RS outer decoder, i.e., System II with feedback.

The performances of Systems III, IV and V when used with the (18,6) unit-memory code on the AWGN channel are shown in Fig. 4. For ease of comparison, the corresponding performances of Systems I and II, given in Fig. 2, are repeated in Fig. 4. By comparing performances between Systems I and III, we see from Fig. 4 that feedback from the outer decoder improves the system by about 0.3 db for a hard-decision Viterbi inner decoder. As can be seen from Table V, the performance of a hard-decision RTMBEP inner decoder is virtually indistinguishable from that of a Viterbi inner decoder for a unit-memory code; thus, the performance of System I in Fig. 4 is also the performance of the system with an RTMBEP inner decoder without feedback from an errors-only RS outer decoder. Hence, by comparing the performances of Systems I and IV in Fig. 4, we can conclude that feedback from the outer decoder improves the system by a full 0.5 db for a hard-decision RTMBEP inner decoder. By comparing the performances of Systems IV and V in Fig. 4, we can further conclude that, when feedback from the outer decoder is used, an additional 0.1 db improvement can be gained by using a soft-decision RTMBEP inner decoder rather than a hard-decision one--the same improvement as was observed in the previous section when there was no feedback from the outer decoder.

V. ZEOLI'S TAIL ANNEXATION SCHEME APPLIED TO A UNIT-MEMORY CONVOLUTIONAL CODE

In [10], Zeoli proposed a concatenated coding system that employed a rather long constraint length (K = 32, i.e., M = 31) convolutional code obtained by annexing a long tail to the M = 7, (3,1) convolutional code. The longer code is then decoded by the same Viterbi decoder as for the short

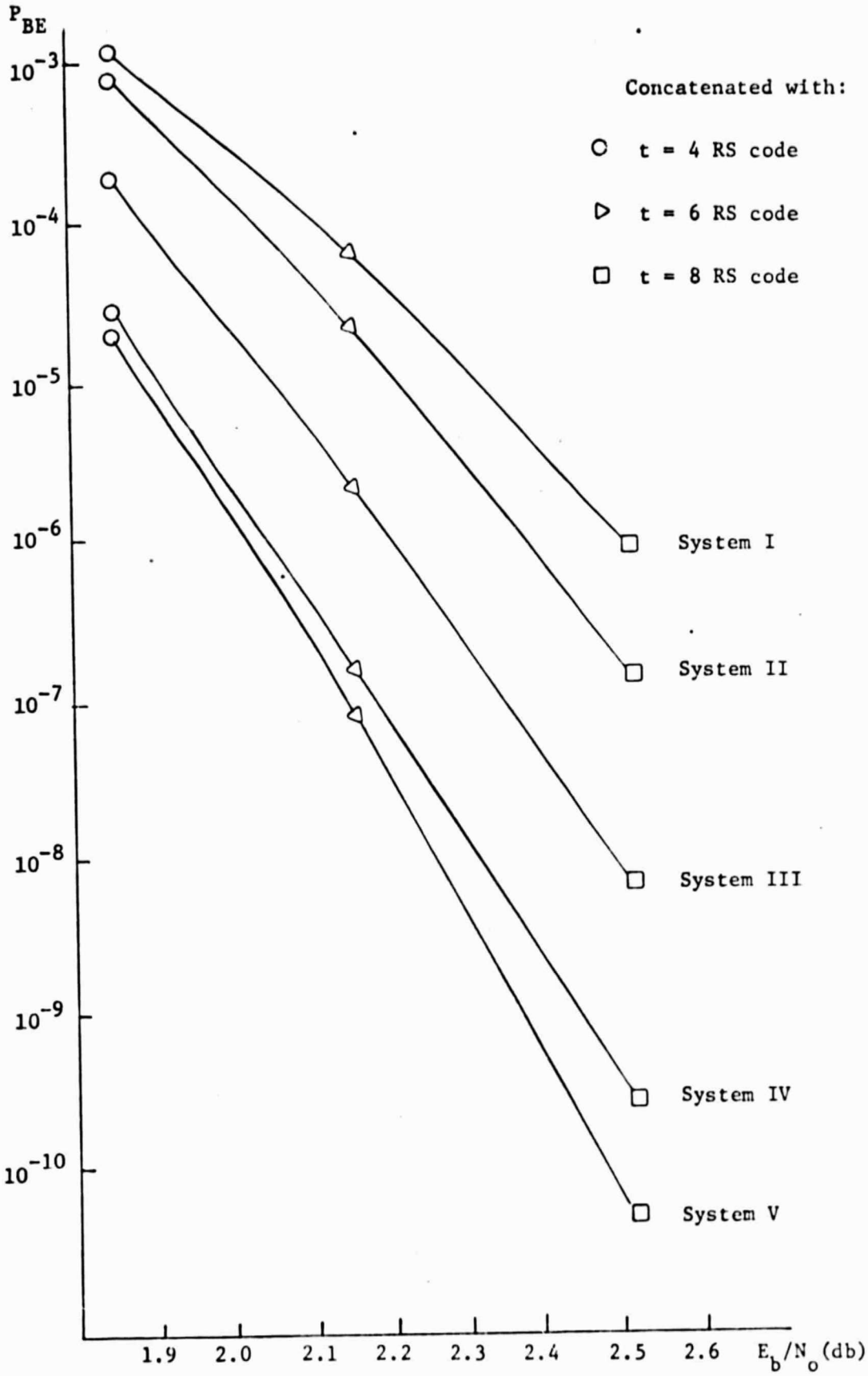

Figure 4. Performance of Concatenated Coding Systems I-V employing the (18,6) unit-memory convolutional code and RS codes over $GF(2^6)$ on a simulated AWGN channel with $E_b'/N_o = 1.25$ db.

code with the exception that the information sequence along the best path to each state is treated as correct and used to "cancel" the effect of the longer tail from the encoded sequence. Thus, the decoder state complexity remains the same as that for the original code and the annexed tail has absolutely no effect on the hard-decision decoding error probability until after an error has been made. But the tail provides excellent "error-detection" once the Viterbi decoder starts to make mistakes. Because the tail is not cancelled when a decoding error is made, the state metrics become extremely ominous after a few decoded branches and can be used as the basis for excellent erasure rules for the output of the inner decoder. However, feedback from the outer decoder is no longer an option, but now a necessity in order to reset the decoder to the correct state and thus to terminate the very "error propagation" used to trigger the erasure alarm.

To study the improvement resulting from Zeoli's scheme, we annexed, to the (18,6) unit-memory convolutional code, a three-branch-long "random tail" such that the resultant code is actually an M = 4, (18,6) convolutional code. The encoding matrices of this latter convolutional code are shown in Table VI. The length of the tail was chosen to be comparable in memory to the M = 31, (3,1) code used in [10]. (Because the decoder is intended to make mistakes continually after its first error, it makes no difference whether the annexed M = 4, (18,6) code is catastrophic [15] or not.) The last of the systems to be considered in this paper, System VI, is that of Zeoli [10], namely a soft-decision Viterbi inner decoder with feedback from an errors-and-erasures RS outer decoder, with the M = 4, (18,6) code replacing his conventional M = 31, (3,1) code.

The state metric used in the "real time Viterbi decoder" [14] of System VI, namely $\mu(t+\Delta) = \log P(\underline{\hat{a}}_{[1,\ t+\Delta]} | \underline{r}_{[1,\ t+\Delta]}$ when $\underline{\hat{a}}_{[1,\ t+\Delta]}$ is the "best path" at time $t+\Delta$, can be used as the basis for an effective erasure

Table VI. The Encoding Matrices of the M = 4 (18,6) Convolutional Code Obtained by Annexing a Randomly-Chosen Tail to the (18,6) Unit-Memory Code.

$$G_0 = \begin{bmatrix} 111000 & 110100 & 110000 \\ 011100 & 011010 & 011000 \\ 001110 & 001101 & 001100 \\ 000111 & 100110 & 000110 \\ 100011 & 010011 & 000011 \\ 110001 & 101001 & 100001 \end{bmatrix} \qquad G_1 = \begin{bmatrix} 000011 & 000111 & 001011 \\ 000110 & 001110 & 010110 \\ 001100 & 011100 & 101100 \\ 011000 & 111000 & 011001 \\ 110000 & 110001 & 110010 \\ 100001 & 100011 & 100101 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 000110 & 000001 & 101111 \\ 100011 & 000011 & 010011 \\ 110001 & 100110 & 100001 \\ 111000 & 110101 & 001000 \\ 011000 & 011010 & 011100 \\ 001100 & 011100 & 110110 \end{bmatrix} \qquad G_3 = \begin{bmatrix} 011000 & 111001 & 011000 \\ 110001 & 110010 & 110000 \\ 100011 & 100101 & 100001 \\ 000111 & 000011 & 001011 \\ 000110 & 001110 & 010110 \\ 001100 & 011100 & 101100 \end{bmatrix}$$

$$G_4 = \begin{bmatrix} 111111 & 010100 & 000000 \\ 000111 & 111010 & 100000 \\ 000000 & 111111 & 010100 \\ 100000 & 000111 & 111010 \\ 010100 & 000000 & 111111 \\ 111010 & 100000 & 000111 \end{bmatrix}$$

Table VII. Variation of Inner Decoder Byte-Error Probability p and Byte-Erasure Probability q and of Outer Decoder Byte-Error Probability P_{BE} with the Erasure Parameter λ for the M = 4 (18,6) Code Obtained by Annexing a Tail to the (18,6) Unit-Memory Code on a Simulated AWGN Channel with an E_b'/N_o of 1.25 db and with the Minimum Distance d_{min} of the Outer RS Code.

λ	p	q	P_{BE} for $d_{min} = 9$	P_{BE} for $d_{min} = 13$	P_{BE} for $d_{min} = 17$
1.50	.00125	.03788	2.095×10^{-4}	8.175×10^{-7}	9.465×10^{-10}
1.80	.00263	.02088	3.602×10^{-5}	1.074×10^{-7}	1.245×10^{-10}
2.00	.00425	.01450	4.168×10^{-5}	1.899×10^{-7}	3.708×10^{-10}

rule as follows. The difference, $\mu(t+\Delta) - \mu(t)$, is, along the correct encoded path, the sum of Δn_o statistically independent random variables, each corresponding to one encoded bit. Note that, for System VI, $\Delta n_o = 8(18) = 144$. The central-limit-theorem can thus be invoked to assert that $\mu(t+\Delta) - \mu(t)$ is approximately Gaussian. Letting m and σ be the (easily calculable) mean and standard deviation of $\mu(t+\Delta) - \mu(t)$, it is natural to use the <u>erasure rule</u>: Erase $\underline{a}_t$ whenever $\mu(t+\Delta) - \mu(t)$ is more than λ standard deviations above m. In Table VII , we give the performance of System VI using this erasure rule for $\lambda = 1.5$, 1.8 and 2.0; the value 1.8 is seen to give the best performance. Note that if $\mu(t+\Delta) - \mu(t)$ were truly Gaussian, the probability that it would exceed $m + 1.8\sigma$ (i.e., the probability of an erasure in the Viterbi decoder output) would be .036; the observed value of .021 given in Table VII is rough confirmation of the appropriateness of the Gaussian approximation.

The performance of System VI on the AWGN channel is shown in Fig. 5; for comparison, the performance of Zeoli's original system, taken from [10], is also shown. The performance of Systems III and V, given in Fig. 4, are also repeated in Fig. 5 to indicate how System VI compares to the systems previously considered. By comparing the performance of Systems III and VI, we see that Zeoli's tail annexation scheme (and the resulting erasure capability) has improved the performance of the feedback system with a Viterbi inner decoder by about 0.2 db.

VI. DEGRADATION OF PERFORMANCE FOR EMPLOYING HIGHER RATE INNER CODES

We have studied, rather extensively, block-convolutional concatenated coding systems employing rate 1/3 convolutional codes and Reed-Solomon codes over GF (2^6). However, it is sometimes desired in practice to operate the

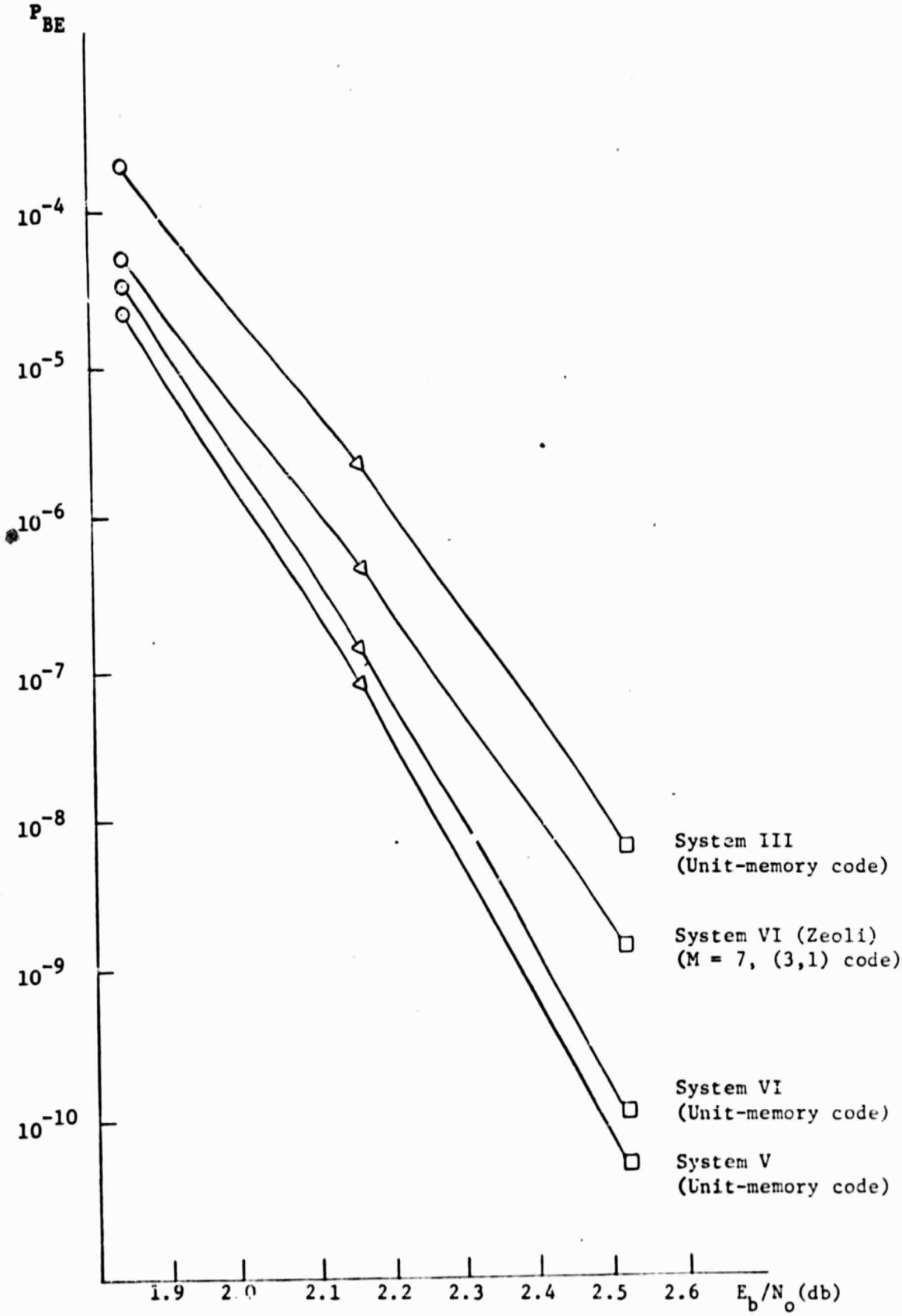

Figure 5. Performance of Zeoli's tail annexation scheme (System VI) on a simulated AWGN channel with E_b'/N_o = 1.25 db, and comparison with other concatenated coding systems.

inner convolutional codes at a higher rate (i.e., narrower bandwidth), rate 1/2 in particular, in order to ease the burden imposed on the phase-lock loops in the receiver. We now describe an heuristic approach to estimate the performance of similar concatenated coding systems with rate 1/2 coding systems from the rate 1/3 results.

From past experience [16], it has been observed that the performance of a rate 1/2 convolutional coding system is about 0.5 db inferior to that of a rate 1/3 convolutional coding system of the same complexity. To verify the general applicability of this rule-of-thumb, we used a hard-decision Viterbi decoder (without feedback) for an M = 6, (2,1) convolutional code on a simulated AWGN channel at E_b'/N_o = 1.75 db, or, equivalently, E_s/N_o = -1.25 db. The results of this simulation and the calculated overall byte-error-probability when this decoder is used with an errors only RS outer decoder concatenated with Reed-Solomon codes are given in Figure 6. For comparison, the performance of the similar R = 1/3 system employing the M = 6, (3,1) code is also shown. We see from Fig. 6 that the latter system is about 0.5 db superior to the former. It seems reasonable then to conclude that a concatenated block-convolutional coding system with a rate 1/2 inner code will be about 0.5 db inferior to that with a rate 1/3 inner code for the same number of decoder states for the Viterbi inner decoder.

VII. CONFIDENCE INTERVALS FOR THE SIMULATION RESULTS

In the preceding, we have reported the performances of numerous block-convolutional concatenated coding systems. The overall byte-error rate was calculated from the byte-error rate of the inner decoder as obtained by simulation. The rather large values of P_{BE} for the inner decoding imply that the simulations require only a modest sample size. Assuming that the decoder makes an error with probability P_{BE} independently for each byte-decision,

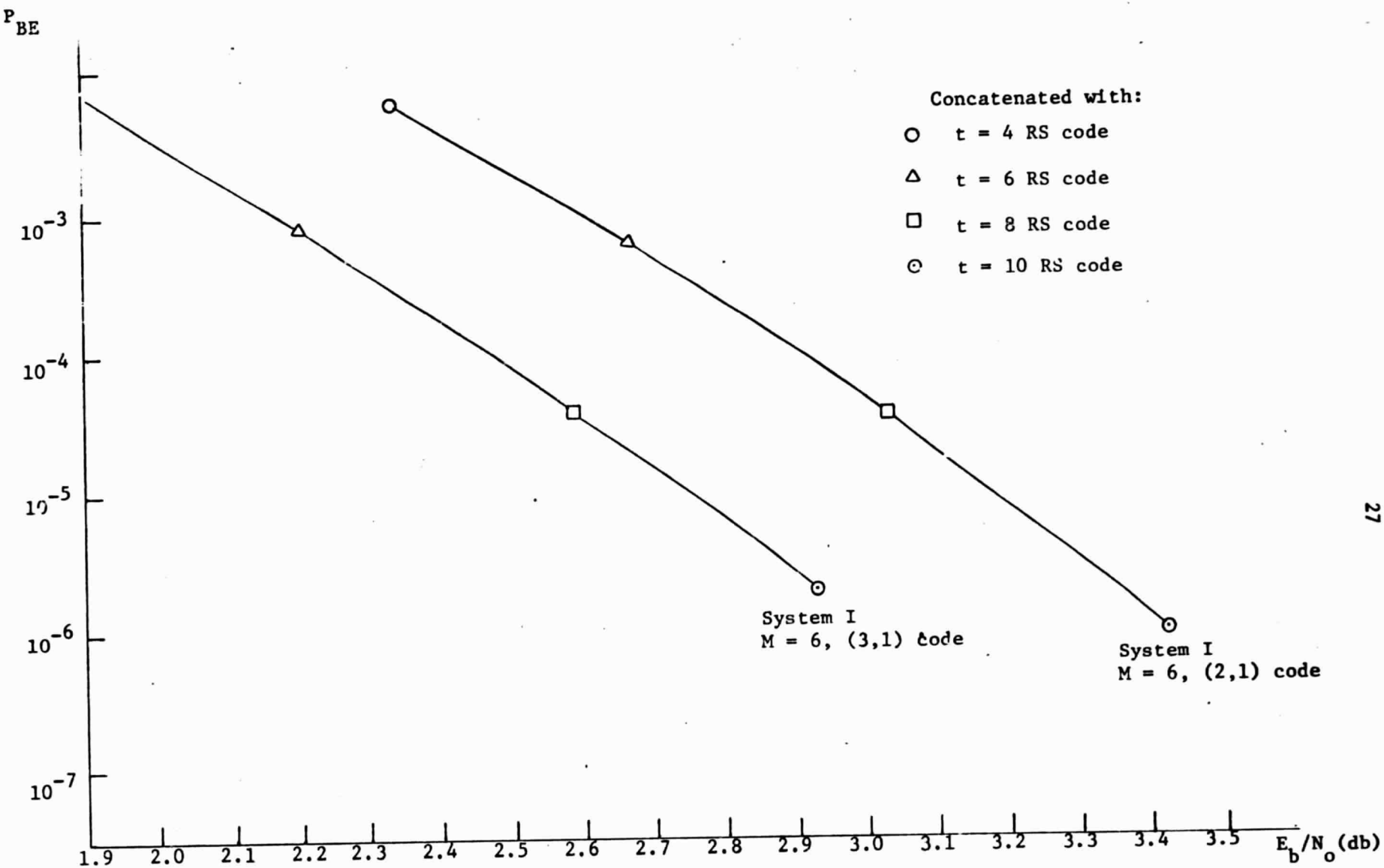

Figure 6. Performance of Concatenated Coding System I on a simulated AWGN channel with E_b'/N_o = 1.75 db when a rate 1/2, M = 6, (2,1) convolutional code is used, and with E_b'/N_o = 1.25 db when a rate 1/3, M = 6, (3,1) convolutional code is used.

the number of byte errors for L decisions is a binomial random variable with parameters L and P_{BE}. The mean value of this random variable, is, LP_{BE}, and the standard deviation is $\sqrt{LP_{BE}(1-P_{BE})}$. L is sufficiently large for this binomial random variable to be well approximated by a Gaussian random variable with the same mean and variance. Since 95.4% of the samples of a Gaussian random variable are within the interval specified by the mean plus and minus twice the standard deviation, we can be 95% confident that the actual byte-error rate for the inner decoder is in the interval $P_{BE} \pm 2\sqrt{LP_{BE}(1-P_{BE})}$. Such 95% confidence intervals are indicated in Tables II and V.

The performances of System I for the M = 6, (3,1) inner code and for the (18,6) unit-memory inner code are shown in Fig. 7 together with their corresponding confidence intervals. We conclude that we may be 95% confident that the actual performance of the concatenated coding system deviates no more than about 0.1 db from our simulation results. Moreover, since all the simulation results are obtained through the same pseudo-random number sequence, the relative differences in performance among various systems are, in fact, much more accurate than the 0.1 db confidnece interval alone would indicate.

VIII. SUMMARY AND CONCLUSIONS

We have extensively studied block-convolutional concatenated coding systems with various modifications. We have found that employing unit-memory convolutional codes rather than conventional codes can improve the performance by nearly 0.3 db. Feedback from the outer decoder to restart a Viterbi inner decoder also contributes an improvement of about 0.3 db. But, surprisingly, feedback from the outer decoder to restart an RTMBEP inner decoder provides an approximately 0.5 db advantage; this might be the principal occasion where the use of RTMBEP decoding rather than Viterbi decoding is justified. Another unexpected result is that soft-decisions by the inner

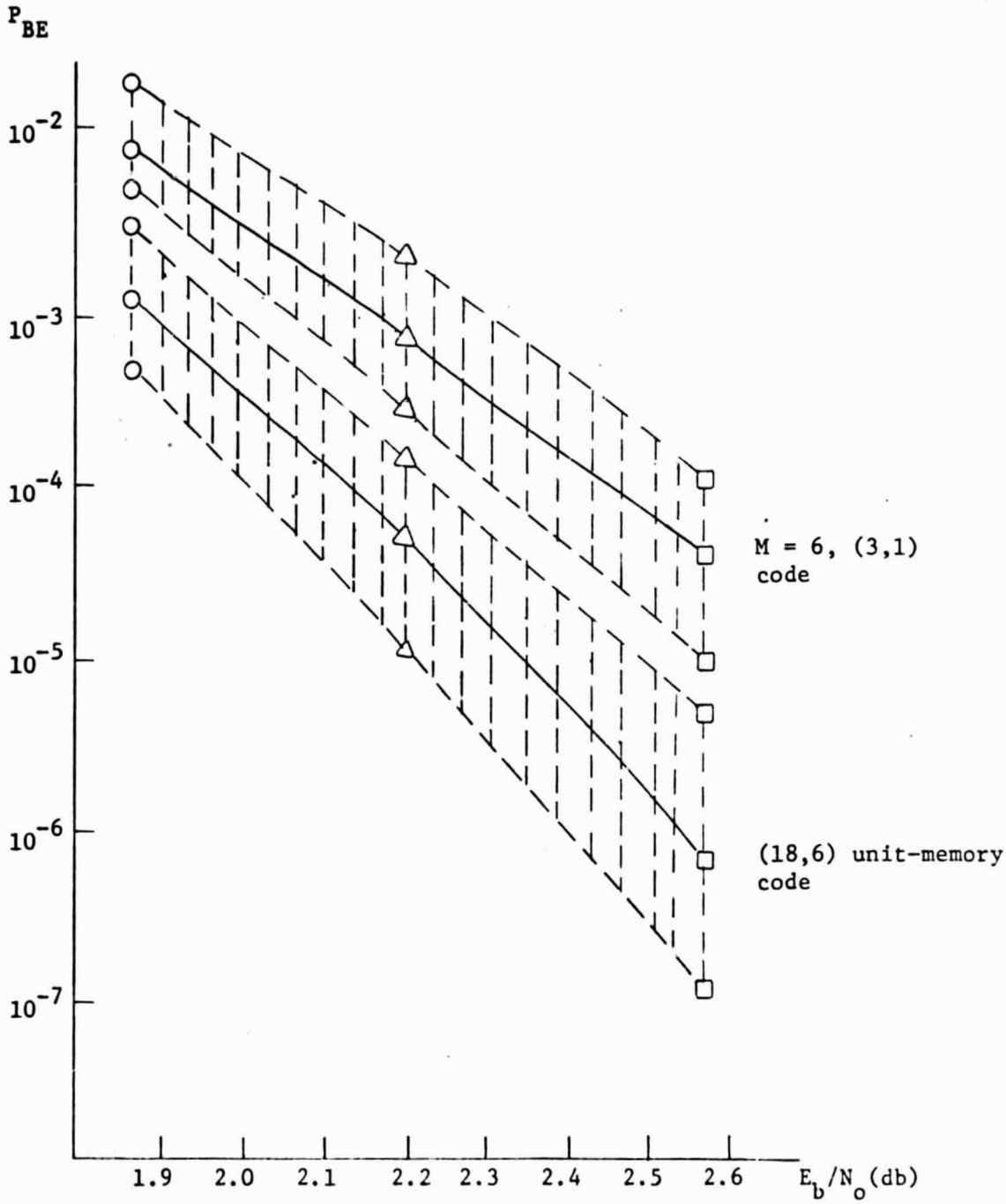

Figure 7. 95% Confidence Intervals for the performance of System I with the M = 6, (3,1) convolutional code and with the (18,6) unit-memory convolutional code.

decoder in conjunction with an erasures-and-errors outer decoder improves the overall performance by only about 0.1 db for RTMBEP decoding. Even with Zeoli's modification, which provides an excellent erasure capability, soft-decisions in conjunction with an erasures-and-errors outer decoder improves performance by about only 0.2 db.

In Fig. 8, we summarize the effects of each feature discussed above on the performance of block-convolutional concatenated coding systems. The figure is drawn in terms of a db scale. As a communications engineer starts to choose a coding system, the first question he faces is whether his phase-locked-loop can tolerate the burden of a rate 1/3 coding system, if the answer is positive, he gains 0.5 db over that of a rate 1/2 inner coding system. Then, he decides which inner code to employ; to choose the M = 7, (3,1) code gives a 0.2 db advantage over the M = 6, (3,1) code but requires twice the number of states in the decoder, whereas to choose the M = 1, (18,6) code gives a 0.3 db advantage with same number of states, but more branch connections required in the inner decoder. The third question is whether he will allow the decisions of the outer decoder to be fed back to the inner decoder; if not, the obvious choice is Viterbi decoding, otherwise, he can gain 0.3 db or 0.5 db depends on whether a Viterbi decoder or an RTMBEP decoder is utilized. And finally, if a soft-decision inner decoder is used, he can gain 0.2 db through Zeoli's erasure scheme if he uses a Viterbi decoder, or gain about 0.05 db if an RTMBEP decoder is employed.

The leading contenders for a good concatenated system are Zeoli's annexation scheme with the unit-memory code (System VI), or either hard decision (System IV) or soft-decision (System V) RTMBEP decoding of the unit-memory code with feedback from the outer decoder. Among them, the soft-decision RTMBEP decoder with feedback performs the best. In terms of hard-ware implementation, Zeoli's modification with the unit-memory code and the

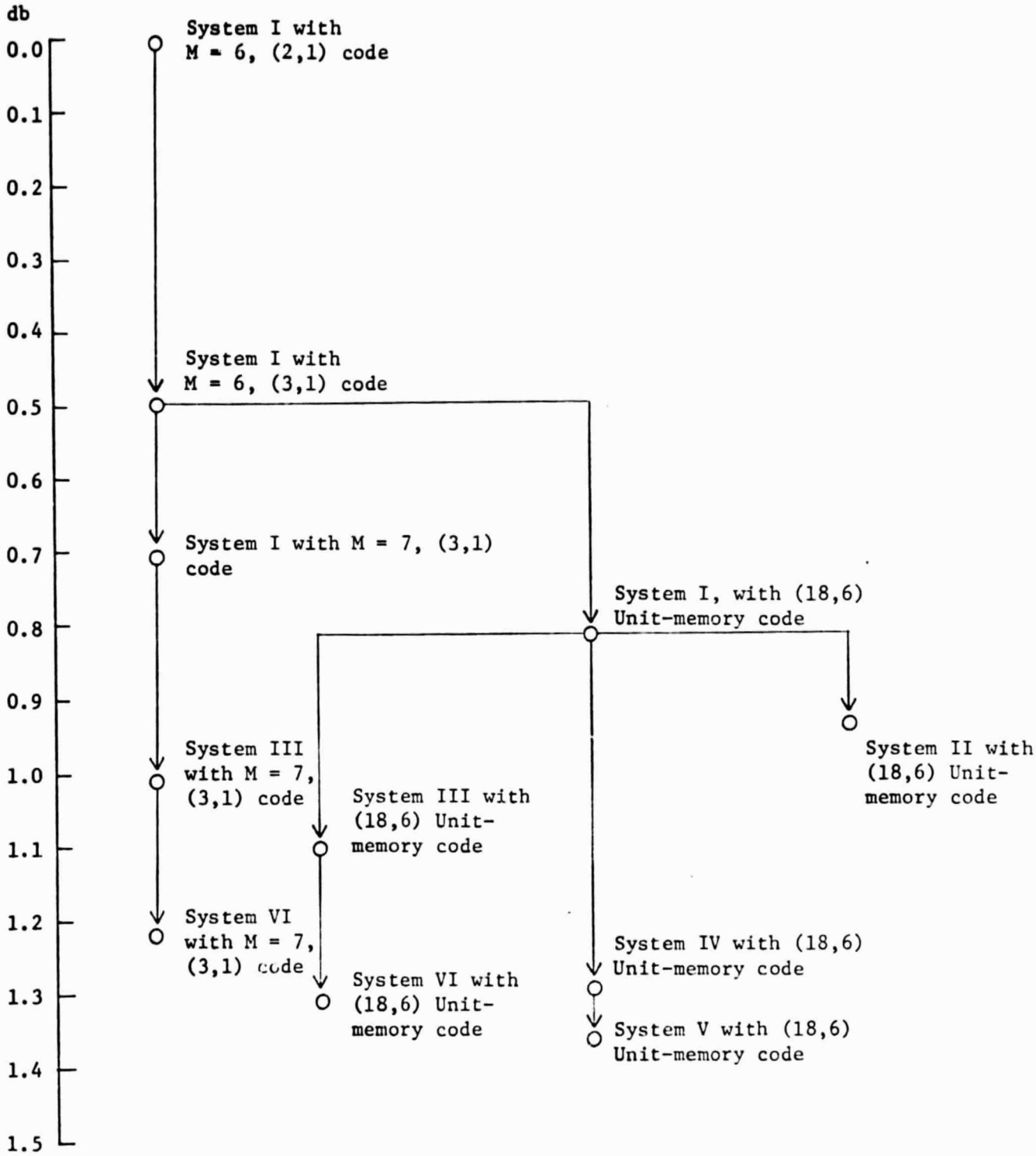

Figure 8. The relative db gains among the concatenated coding systems st died.

hard-decision RTMBEP decoder are of approximately the same complexity. However, since the operation of the Viterbi decoder for Zeoli's system depends on the correct feedback from the outer decoder, there is always a slim chance that the outer decoder fails to provide correct decisions to the Viterbi decoder. Since the encoder constraint length is much larger than the decoder constraint length, this can cause endless errors as if a catastrophic convolutional code were used. Thus, it is necessary to send synchronization signals periodically to reset the Viterbi decoder to guarantee restoration of normal operation. The RTMBEP decoder has the same constraint length as that of the encoder, therefore the decoder is able to recover from errors in a few branches by itself without feedback. The feedback from the outer decoder only speeds this process up; therefore, when an error is fed back, the most damage it can cause is for the RTMBEP decoder to make a few more errors before it recovers by itself. This is certainly a very desirable advantage for a concatenated coding system. Moreover, because the decoder can restore its normal operation quickly, the degree of interleaving required for this scheme is considerably less than the full Reed-Solomon block length interleaving required for the Zeoli's scheme.

Finally, as a remark to information theorists, we note that for System III (the RTMBEP inner decoder for the rate 1/3 (18,6) unit-memory code concatenated with the (63,51), 6-error-correcting RS code with feedback from the RS errors-only decoder) we can achieve a byte-error-probability of 10^{-7} at E_b/N_o of 2.25 db, or, equivalently, at E_s/N_o of 3.25 db. The cut-off rate, R_{comp}, of this 8-level quantized AWGN channel is 0.275 whereas its channel capacity is 0.44. The overall rate of the concatenated coding system is 0.27. It seems that the cut-off rate, rather than the channel capacity, is still the practical limit of rate for reliable communications, even for a very

sophisticated concatenated coding system, just as it is in a conventional convolutional coding system employing sequential decoding [16]. The advantage of the concatenated coding system resides only in the elimination of "deleted data" such as is always present in a sequential decoding system because of the latter's highly variable computation.

ACKNOWLEDGEMENT

The author would like to express his gratitude to his Dissertation Advisor, Professor James L. Massey for his patient and generous guidance of this research and for his suggestions to match byte-oriented convolutional codes with Reed-Solomon codes and to develop the "real-time minimal-byte-error probability (RTMBEP) decoding algorithm."

REFERENCES

[1] P. Elias, "Error-Free Coding," IRE Transactions on Information Theory, Vol. IT-4, pp. 29-37, Sept. 1954.

[2] G. D. Forney, Jr., Concatenated Codes. Cambridge, Mass.: M.I.T. Press, 1966.

[3] D. D. Falconer, "A Hybrid Sequential and Algebraic Decoding Scheme," Ph.D. Dissertation, Dept. of Electrical Engineering, M.I.T., Cambridge, Mass., 1966.

[4] F. Jelinek and J. Cocke, "Bootstrap Hybrid Decoding for Symmetrical Binary Input Channels," Information and Control, Vol. 18, pp. 261-298, March 1971.

[5] A. J. Viterbi, "Error Bounds for Convolutional Codes and An Asmptotically Optimal Decoding Algorithm," IEEE Transactions on Information Theory, Vol. IT-13, pp. 260-269, April 1967.

[6] J. P. Odenwalder, "Optimal Decoding of Convolutional Codes," Ph.D. Dissertation, School of Engineering and Applied Sciences, University of California, Los Angeles, 1970.

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

[8] J. L. Massey, "Shift-Register Synthesis and BCH Decoding," IEEE Transactions on Information Theory, Vol. IT-15, pp. 122-125, Jan. 1969.

[9] L. N. Lee, "Short, Unit-Memory, Byte-Oriented, Binary Convolutional Codes Having Maximal Free Distance," IEEE Transactions on Information Theory, Vol. IT-22, No. 3, pp. 349-352, May 1976.

[10] G. W. Zeoli, "Coupled Decoding of Block-Convolutional Concatenated Codes," IEEE Transactions on Communications, Vol. COM-21, pp. 219-226, March 1973.

[11] F. Jelinek, "Bootstrap Trellis Decoding," IEEE Transactions on Information Theory, Vol. IT-21, pp. 318-325, May 1975.

[12] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," IEEE Transactions on Information Theory, Vol. IT-20, pp. 284-288, March 1974.

[13] P. L. McAdam, L. R. Wlech and C. L. Weber, "M.A.P. Bit Decoding of Convolutional Codes," presented at 1972 International Symposium on Information Theory, Asilomar, California, January 1972.

[14] L. N. Lee, "Real-Time Minimal-Bit-Error Probability Decoding of Convolutional Codes," IEEE Transactions on Communications, Vol. COM-22, pp. 146-151, Feb. 1974.

[15] J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits," IEEE Transactions on Computers, Vol. C-17, pp. 330-337, April 1968.

[16] I. M. Jacobs, "Sequential Decoding for Efficient Communications from Deep Space," IEEE Transactions on Communication Technology, Vol. COM-15, pp. 492-501, Aug. 1967.