Erasure Declaring Viterbi Decoders

F. Pollara

Communications Systems Research Section

Several methods for realizing erasure declaring Viterbi decoders for the (7, 1/2) NASA code are discussed. Only bit oriented algorithms are considered. When such decoders are used in a concatenated system with a (255,223) Reed-Solomon decoder, improvements on the probability of word error of at most 0.1 dB were obtained.

I. Introduction

Reliable deep space communication can be realized with concatenated coding systems based on an inner convolutional code and an outer Reed-Solomon code. A $(7, 1/2)$ inner convolutional code and an 8-bit $(255,223)$ outer Reed-Solomon code are used in NASA's Voyager mission and as an international coding standard.

This Reed-Solomon code can correct any word such that $2e + E \leqslant 2t$, where e is the number of symbol errors in the word, E the number of erasures, and $2t = 32$ the number of parity symbols. While Reed-Solomon decoders which can correct erasures can be easily implemented (Ref. 1), methods for estimating symbol quality and criteria for declaring symbol erasures are open to discussion.

There are two distinct classes of methods: one extracts quality information from the Viterbi decoder, the other examines adjacent symbols of interleaved Reed-Solomon words and erases symbols next to incorrect symbols. Both these classes of methods have been considered in Ref. 2. We will concentrate on further variations of the first class methods, which give better results than those reported in Ref. 2, and comparable results to methods belonging to the second class.

II. Reed-Solomon Code Performance

Our $(255,223)$ Reed-Solomon code over $GF(2^8)$ has a word error probability given by Ref. 3:

$$P_w = \sum_{i=0}^{n} \sum_{\substack{j=2(t-i)+1 \\ j \geqslant 0}}^{n-i} \alpha p^i s^j (1 - p - s)^{n-i-j} \qquad (1)$$

where

$$\alpha = \frac{n!}{i!j!(n-i-j)!}$$

and $n = 255$; p is the symbol error rate at the Reed-Solomon decoder input; and s is the symbol erasure rate. The only ingredients needed to compute the performance of the concatenated system are therefore p and s, which are measured at the output of a software simulated Viterbi decoder, driven by convolutionally encoded data in additive Gaussian noise. The values of E_b/N_0 shown in this report have been increased by 0.58 dB to take into account the Reed-Solomon code rate. These values represent the correct SNR of the concatenated channel.

III. Erasure Declaring Viterbi Decoders

The general problem is that of modifying the Viterbi algorithm so that an estimate on the quality of decoded bits or symbols (8 bits in our case) can be provided.

Theoretically we should be able to compute the a posteriori probability for each bit or symbol and then compare it to a threshold to declare erasures. Since the Reed-Solomon decoder operates on symbols, we are ultimately interested in the a posteriori symbol probability. Lee (Ref. 4) devised a decoding algorithm called real-time minimal byte error probability (RTMBEP) decoder, which actually provides the a posteriori symbol probability, but is unfortunately too complex to implement. A similar MAP bit decoding algorithm has been proposed by P. L. McAdam (Ref. 5), but suffers from similar implementation complexity problems.

Byte-oriented decoders should offer better erasure information on Reed-Solomon symbols, but due to their complexity we restrict ourselves to conventional bit-oriented Viterbi decoders.

Several modified Viterbi algorithms will be described as follows.

Method A. Extraction of quality information for Viterbi decoded bits from the rate of metric renormalization has been considered in Ref. 2, and will not be repeated here. This method's performance is severely limited by the fact that the renormalization rate cannot resolve precisely enough in time which bits are less reliable.

Method B. The conventional Viterbi algorithm searches all possible code sequences (paths) and finds the most likely transmitted sequence. Accumulated metrics can be viewed as distances between surviving paths and the received sequence, where the closest path has metric normalized to zero and all other paths have some positive metric. These metrics do not contain enough information to reconstruct the a posteriori probabilities of paths, since some paths are pruned at each stage and their effect is thereafter ignored, together with the probabilities of all the paths that could have departed from them. Yet the accumulated metrics do contain "some" information about the reliability of each surviving path. Typically the metric values will be spread over a wide range of positive values up to approximately $2(K - 1)$ (Ref. 6), where K is the constraint length of the code, if SNR $\gg$ 1, while they will tend to accumulate around zero if SNR is low. Therefore the reliability of a chosen path (zero metric) can be estimated according to the *spread* of the metrics.

Another basic problem is that we are interested in the quality of a decoded bit and not in that of an entire path. Different paths may or may not yield identical decoded bits.

Once the quality of a bit has been decided, we will declare erasure by comparing it with a threshold. Ultimately, we will have to decide on symbol (byte) erasures, which can be declared if one or more bits are erased in the symbol. Although this study is limited to bit-oriented Viterbi decoders, symbol erasure criteria based on multiple bits erasures or multiple bit cumulative quality have been tested. No measurable improvements were found as compared to declaring symbol erasure based on a single bit erasure in the symbol.

Let m_j be the accumulated metric of state j at a given time t; $b_j \in (+1, -1)$ be the information bit belonging to the surviving sequence into state j at time $t - \tau$; and j^* be the state such that $m_{j^*} = 0$, i.e., the state chosen as most likely at time t, as shown in Fig. 1. Then b_{j^*} will be the decoded bit at time t.

In a Viterbi decoder with $M = 2^{K-1}$ states, Method B forms the sum,

$$B = \sum_{j=0}^{M-1} b_j, \qquad b_j \in (-1, +1)$$

and then declares erasure if $Bb_{j^*} < 0$. This corresponds to a simple majority rule, where an erasure is declared if less than half of the b_j agree with b_{j^*}.

Method C. Let

$$m_{\max} = \underset{j}{\text{Max}} \, \{m_j\}$$

and $m_j' = m_{\max} - m_j, j = 0, \ldots, M - 1$. This method defines B as

$$B = \sum_{j=0}^{M-1} m_j' b_j$$

and declares erasure if

$$Bb_{j^*} < -\frac{1}{2} \sum_{j=0}^{M-1} m_j'$$

This corresponds to a weighted majority rule, where a higher weight is assigned to more reliable paths.

Method D. This method (proposed by D. Divsalar) is substantially different from all others. It stores survivors as strings of symbols from a ternary alphabet $(+1, -1, E)$, where E represents erasure and $b_j \in (+1, -1, E)$. An erasure is stored when the relative difference of the two competing accumulated metrics is less than a threshold T. Let these two metrics be denoted by $m_j^{(a)}$ and $m_j^{(b)}$, where

$$m_j = \underset{j}{\text{Min}} \{m_j^{(a)}, m_j^{(b)}\}$$

then an erasure is stored if

$$\frac{|m_j^{(a)} - m_j^{(b)}|}{m_j^{(a)} + m_j^{(b)}} < T$$

where T is small constant. An erasure is declared if $b_{j*} = E$.

Method E. This method defines B as

$$B = \sum_{j=0}^{M-1} e^{-m_j} b_j$$

and declares erasure if

$$Bb_{j*} < T \sum_{j=0}^{M-1} e^{-m_j},$$

where T is a threshold value. This corresponds to a nonlinearly weighted majority rule.

Method F. Let J_1 and J_0 be two sets of values of j such that $b_j = 1$ and $b_j = 0$, respectively. Then, this method defines B_1 and B_0 as

$$B_1 = \sum_{j \in J_1} e^{-m_j}$$

$$B_0 = \sum_{j \in J_0} e^{-m_j}$$

and declares erasure if

$$b_{j*} \log\left(\frac{B_1}{B_0}\right) < T,$$

which is equivalent to a log-likelihood ratio test.

IV. Results and Conclusion

All the above methods have been tested extensively by simulation, for various threshold values. Thresholds showed a mild dependency on E_b/N_0, and have been optimized by repeated trials.

Symbols containing at least one reliable bit have been erased. Erasing only symbols with at least two unreliable bits has been tried and proved inferior.

Among all methods described, Method F performed consistently, though slightly, better than any other. The superiority of Method F is due to the fact that this method yields the closest possible approximation of true a posteriori probabilities, based on the Viterbi algorithm. The performance of this method in terms of probability of word error at the output of Reed-Solomon decoder (Eq. [1]) vs. concatenated E_b/N_0 is shown in Fig. 2, where the performance of the usual (no erasure) system is also shown for comparison. The curve denoted as "lower bound" is the performance of an hypothetical system, where a "genie" knows exactly all symbols in error, which are then erased.[1] Aside from the lower bound, the two sets of curves in Fig. 2 represent two different truncation lengths L of survivors in the Viterbi decoder.

These simulation results indicate that an improvement of approximately 0.1 dB can be obtained at $P_w = 10^{-5}$ by an erasure declaring Viterbi decoder, based on Method F, if $L = 32$ bits. This result compares favorably with that obtained in Ref. 2. However, the improvement becomes negligible if the truncation length is increased to 64 bits.

These results seem to suggest that the additional gain available may be achievable only with the more complex, symbol oriented, algorithms based on true a posteriori symbol probabilities.

---

[1]This method was devised by L. Deutsch, and reported in JPL IOM 331-83-132A (internal document), by L. Swanson, Jet Propulsion Laboratory, Pasadena, Calif., April 12, 1983.

References

1. Blahut, R. E., *Theory and Practice of Error Control Codes*. New York: Addison-Wesley, 1983.

2. Pitt, G. H., III and Swanson, L., "Erasure Information for a Reed-Solomon Decoder," *TDA Progress Report 42-83*, pp. 39–44, Jet Propulsion Laboratory, Pasadena, Calif., November 1985.

3. Clark, G. C., and Cain, J. B., *Error-Correction Coding for Digital Communications*. New York: Plenum Press, 1981.

4. Lee, L. N., "Concatenated Coding Systems Employing a Unit-Memory Convolutional Code and a Byte-Oriented Decoding Algorithm," *IEEE Trans. on Communications*, Vol. COM–25, No. 10, pp. 1064–74, October 1977.

5. McAdam, P. L., *MAP Bit Decoding of Convolutional Codes*, Ph.D. Disseration, University of Southern California, Los Angeles, February 1974.

6. Viterbi, A. J., and Omura, J., *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
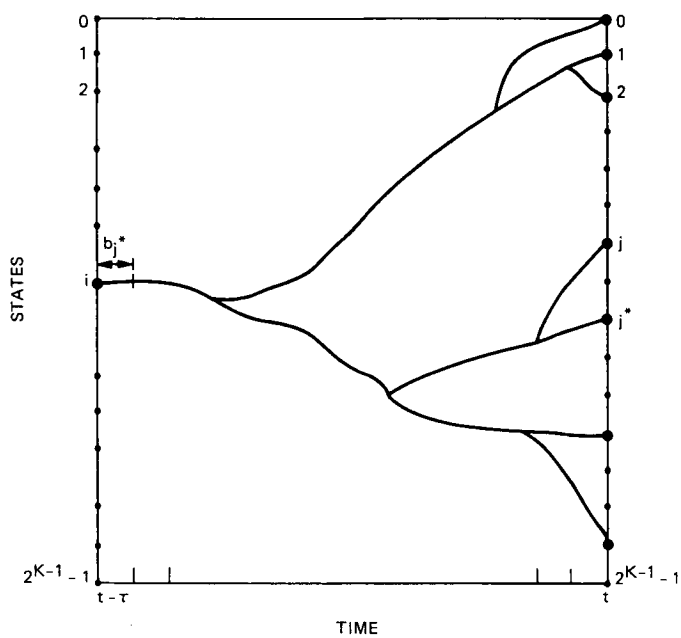
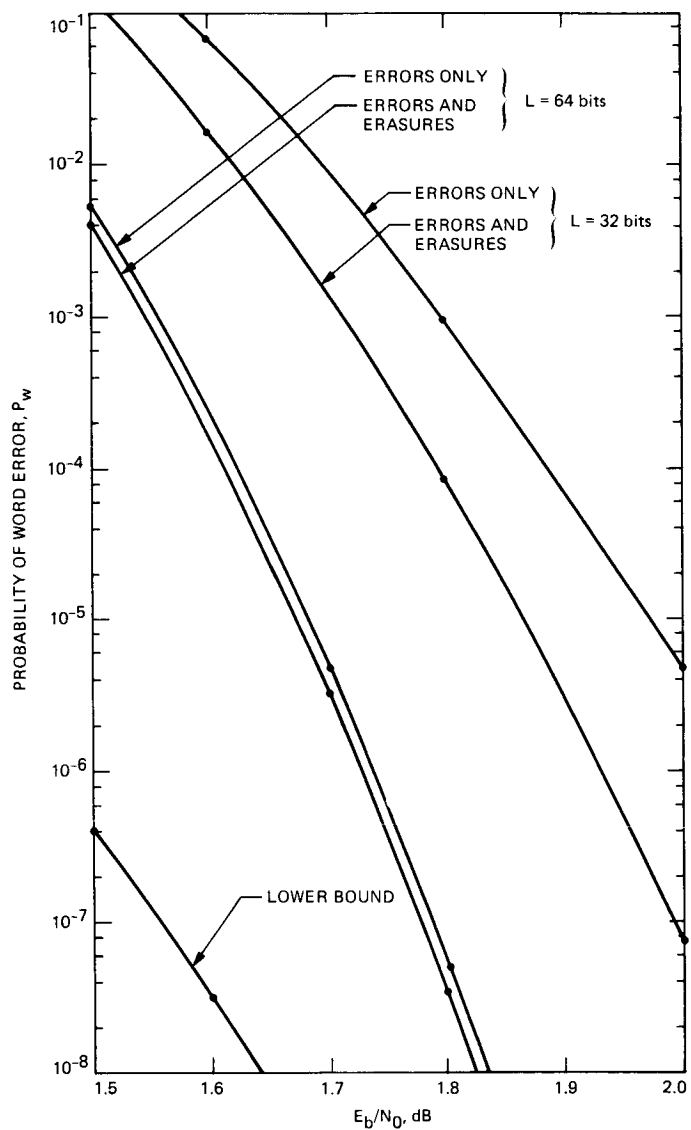Fig. 1. Example of surviving sequences at time t

Fig. 2. Probability of word error vs E_b/N_0 for Method F