Viterbi Decoding Modified for Sources With Memory

V. Korwar
TDA Planning

This article investigates the gain in signal-to-noise ratio that may be realized by using redundancies inherent in TV data to modify the decoding metric of the DSN Viterbi decoders. This modification would take into account the memory in a TV scan line to change the transition probabilities as originally computed using independent data. The resulting data rate gain or error probability decrease is achieved without requiring any spacecraft modifications or additions. A preliminary examination of this concept on a binary symmetric channel rather than a Gaussian channel and using a simplified Markov source model involving two-level or hard-clipped TV shows that very substantial decreases in error probability may be achieved.

I. Introduction

Viterbi decoding for the case where binary messages from a memoryless source are transmitted over a memoryless channel implements maximum likelihood (ML) decoding. In this case, this is equivalent to maximum *a posteriori* probability (MAP) decoding since all messages are equally likely. However, when the source has memory, such as for video, ML decoding, and hence the Viterbi decoding algorithm, as it is usually implemented, is not optimal. For such cases work has been done (see, e.g., Ref. 1) on encoding the source before transmission and then using a source decoder at the receiver. This method utilizes the channel more efficiently. However, in cases where the transmitter complexity needs to be minimized, such as when it is on a spacecraft, it is sometimes undesirable to have a source encoder, which may be a complex piece of equipment. In such a case, it could be worthwhile modifying the Viterbi decoder at the receiving end so as to increase the transmission rate, while leaving the transmitter untouched.

In the present work, we consider the simple case of a Markov source of arbitrary order S, and a memoryless channel, and show how to modify the metric in the Viterbi decoder to implement MAP decoding. Calculations of bit error probability (BEP) for a decoder using this new metric become message-dependent. An upper bound on the BEP can be obtained for any given message by a method to be described, but elegant bounds similar to those in conventional Viterbi decoding (see, e.g., Refs. 2 and 3) are only possible for a few messages, and these are derived. An upper bound on overall BEP has also been similarly derived for low-entropy sources. For a specific $K = 3$ constraint length code, computer simulations of the decoder for a first-order Markov source and a binary symmetric channel (BSC) indicate that the BEP with the new metric are lower than the BEP with the conventional metric, as predicted, with the improvement being more noticeable at rates well below channel capacity. However, many more runs of these simulation programs are needed to determine more exactly the extent of the improvement for various parameters.

Also, we are interested in the Gaussian channel and not the binary symmetric channel for which the simulations in this work have been done.

It has been observed that the Viterbi algorithm (VA), in its most general form, is "a solution to the problem of MAP estimation of the state sequence of a finite-state discrete-time Markov process observed in memoryless noise" (Ref. 4). But the Viterbi Algorithm does not seem to have been used as a MAP estimator in the context of decoding convolutionally encoded messages from a Markov source. It has, however, been used in the problem of text recognition (Refs. 5 and 6). Also, error bounds like the ones given here, for cases where the source has memory, do not seem to have been obtained, nor are there available comparisons between the MAP and ML estimator applications of the VA to decoding problems for sources with memory.

II. The New Metric

A. Definition for a BSC

Consider a BSC with channel error or transition probability p_e and a convolutional code of constraint length K, where b bits are transferred at a time into the encoder and n symbols are output at a time (i.e., a rate b/n code). Here, "at a time" means at each clock pulse. There are bK bits in the encoder register.

A Markov source, which we define below, is a simple example of a source with memory. A Markov source of order S is defined as one having the property that the probability distribution of the n^{th} bit it outputs (for $n > S$) depends only on the values of the previous S bits output. That is, if the sequence of bit output by the source is $\{u_1, u_2, \ldots, u_n, \ldots\}$, then for $n > S$, the conditional probability $p(u_n|u_{n-1}, \ldots, u_1)$ can be expressed as

$$p(u_n|u_{n-1}, \ldots, u_1) = p(u_n|u_{n-1}, \ldots, u_{n-S}) \quad (1a)$$

A symmetric first-order Markov source has

$$p(1|0) = p(0|1) \triangleq p_s \quad (1b)$$

and

$$p(0|0) = p(1|1) \triangleq q_s = 1 - p_s \quad (1c)$$

Here p_s is called the source transition probability. In this paper, whenever we refer to a first-order Markov source, we mean this symmetric one.

We first consider a Markov source of order $S \leqslant bK - 1$, and, for this case, we show that MAP decoding is achieved by using the Viterbi decoding algorithm with modified branch metrics M'_{mi} defined below. Each branch defines a transition out of a state s involving $b(K - 1)$ bits to a new state t uniquely defined by the b incoming bits and the $b(K - 1)$ bits of state s. Thus, if the Markov source memory extends over fewer than bK previous bits, there is a unique *a priori* probability p_{mi} of occurrence of each branch in the state diagram given by Eq. (2b) below.

If $S > bK - 1$, we can follow exactly the same decoding procedure with the metrics which we define below, with the exception that the state diagram and the trellis diagram for the decoder now have $2^{(S-b+1)}$ states, with each state represented by $(S - b + 1)$ bits rather than $2^{b(K-1)}$ states, each represented by $b(K - 1)$ bits. (Each node will, as usual, have 2^b branches merging at it and 2^b branches emanating from it.) However, since $S - b + 1 > b(K - 1)$ for $S > bK - 1$, this increases the decoder complexity. So, if b is small, we might as well increase K, which increases the encoder complexity only slightly, but reduces BEP considerably. Thus, it may often be worthwhile choosing b, K, S such that $S \leqslant bK - 1$ holds, if some other constraints are not violated in doing so. If b is large, however, increasing K even by 1 increases the number of states by a factor of 2^b. This may increase the decoder complexity much more than if we were to keep K fixed and have $2^{(S-b+1)}$ states in the decoder to take care of the source memory S being greater than $bK - 1$.

The branch metric of conventional Viterbi decoding for the i^{th} branch of the m^{th} possible path through the trellis is $-d_{mi}$ where d_{mi} is the Hamming distance between the n-dimensional code subvector for the branch and the corresponding n-dimensional received subvector (see Ref. 2). We define the new metric as

$$M'_{mi} = -d_{mi} + \frac{\ln p_{mi}}{\ln \dfrac{(1 - p_e)}{p_e}} \quad (2a)$$

p_{mi} can be written as the conditional probability

$$p_{mi} = p(u_{mi}|u_{m(i-1)}, \ldots, u_{m(i-S)}) \quad (2b)$$

for an S^{th} order Markov source, where u_{mi} is the i^{th} bit of the m^{th} possible message of block length B. Note that m runs from 0 to $2^B - 1$. We assume throughout that $B >> K$, and that the last $b(K - 1)$ bits transmitted are always 0.

B. Definition for an Arbitrary Memoryless Channel

The conventional branch metric M_{mi} for the i^{th} branch of the m^{th} path is the logarithm of the conditional probability of the i^{th} received subvector (Ref. 2):

$$M_{mi} = \ln p(\bar{y}_i|\bar{x}_{mi}) \tag{3}$$

where $\bar{x}_{mi}$ is the n-dimensional code subvector of the m^{th} message sequence for the i^{th} branching level, and $\bar{y}_i$ is the corresponding n-dimensional noisy received vector. The modified metric in this case is

$$M'_{mi} = \ln p(\bar{y}_i|\bar{x}_{mi}) + \ln p_{mi} \tag{4}$$

where p_{mi} is defined in Eq. (2b).

C. Derivation of New Metric

We now derive the new metric defined above for the case where the source memory is S.

Let $\{u_{m1}, \ldots, u_{mB}\}$ be the m^{th} possible message sequence of block length B. Then the *a priori* probability of the sequence $\{u_{m1}, \ldots, u_{mB}\}$ is

$$\begin{aligned} p(m) &= p(u_{m1}, \ldots, u_{mB}) \\ &= p(u_{mB}|u_{m(B-1)}, \ldots, u_{m(B-S)})\, p(u_{m(B-1)}|u_{m(B-2)}, \\ &\quad \ldots u_{m(B-S-1)}) \ldots p(u_{m1}|u_{m0}, u_{m(-1)}, \ldots, u_{m(1-S)}) \\ &= p_{mB} p_{mB-1} \ldots p_{m1} \end{aligned} \tag{5a}$$

where p_{mi} is defined as in Eq. (2b), and we define

$$\{u_{m0}, u_{m(-1)}, \ldots, u_{m(1-S)}\} = \{0, 0, \ldots, 0\} \tag{5b}*$$

Then the MAP decoder should choose that B-bit block corresponding to message m that has the maximum *a posteriori* probability given by

$$p_{ap} = \prod_{i=1}^{B} p(\bar{y}_i|\bar{x}_{mi}) \frac{\{p(m)\}}{p(\bar{y})} \tag{6}$$

*Alternatively, we may take care of the initial bits by defining $p_{m\bar{i}}, \ldots, p_{mS}$ each to be 0.5.

where $p(m)$ is given by Eq. (5), $\bar{x}_{mi}, \bar{y}_i$ have the same meaning as in Eq. (4), and $p(\bar{y})$ is the probability of receiving $\bar{y}$. This probability is constant for all messages m, and hence can be dropped from the maximization; or, equivalently, we can maximize the log of this probability (after dropping $p(\bar{y})$), i.e.,

$$\ln p'_{ap} = \sum_{i=1}^{B} \ln \left\{ p(\bar{y}_i|\bar{x}_{mi}) \right\} + \ln \{p(m)\} \tag{7}$$

$$= \sum_{i=1}^{B} \left\{ \ln p(\bar{y}_i|\bar{x}_{mi}) + \ln p_{mi} \right\} \tag{8}$$

From this it follows that the branch metric for the i^{th} branch of the m^{th} message path is given by Eq. (4). For a BSC with transition probability p_e, we can further simplify Eq. (4) using the fact that (Ref. 2)

$$p(\bar{y}_i|\bar{x}_{mi}) = p_e^{\,d_{mi}} (1 - p_e)^{n - d_{mi}} \tag{9}$$

Thus, Eq. (4) becomes

$$M''_{mi} = \left\{ -d_{mi} \ln \left(\frac{1 - p_e}{p_e} \right) + n \ln (1 - p_e) + \ln p_{mi} \right\} \tag{10}$$

Or, dropping the $n \ln (1 - p_e)$, which is constant for all branches, and dividing by

$$\ln \left(\frac{1 - p_e}{p_e} \right)$$

we get

$$M'_{mi} = -d_{mi} + \frac{\ln p_{mi}}{\ln \left(\frac{1 - p_e}{p_e} \right)} \tag{11}$$

which establishes Eq. (2a).

III. Error Probability Bounds With New Metric

We now derive upper bounds on the probability of error with the new metric in a manner paralleling the derivation of

Eq. 4.4-8 of Ref. 2, which is repeated as Eq. R1 below:

$$E[n_b(j)] \leqslant \sum_{i=1}^{\infty} \sum_{d=d_f}^{\infty} ia(d,i)Z^d \qquad \text{(R1)}$$

where $E[n_b(j)]$ is the expected number of bit errors caused by an incorrect path diverging at node j; $a(d,i)$ is the number of paths diverging from the all-zeros path (at node j) at distance d and with i "1's" in its data sequence over the unmerged segment; d_f is the minimum distance of any path from the correct one, called the free distance; and Z is defined in Eq. (R2):

$$Z = \sum_{y} \sqrt{p_1(y)\, p_0(y)} \qquad \text{(R2a)}$$

Here $p_1(y)$, $p_0(y)$ are the probabilities of receiving a given value of a bit y when the corresponding transmitted bit is a 1 or a 0, respectively. For the special case of a BSC, Z reduces to

$$Z = \sqrt{4p_e(1-p_e)} \qquad \text{(R2b)}$$

A. Extension of the Bhattacharya Bound

We first extend the Bhattacharya bound to the case in which MAP detection occurs rather than ML detection.

If $p_E(m \to m')$ denotes the probability that message m' is decoded when message m is sent and only two alternatives (m and m') exist, and if $\bar{x}_m$ is the signal vector sent and $\bar{y}$ the one received, and if $\bar{x}'_m$ is the other possible signal vector, then the Bhattacharya bound states (Eq. 2.3.15 of Ref. 2) that

$$p_E(m \to m') \leqslant \sum_{y} \sqrt{p(\bar{y}|\bar{x}'_m)\, p(\bar{y}|\bar{x}_m)} \qquad \text{(R3)}$$

We now extend this bound to the case of MAP detection. We have

$$p_E(m \to m') = \sum_{\bar{y} \in \Lambda_{mm'}} p(\bar{y}|\bar{x}_m) \qquad \text{(12a)}$$

where

$$\Lambda_{mm'} = \left\{ \bar{y}: \frac{p_{m'}\, p(\bar{y}|\bar{x}'_m)}{p_m\, p(\bar{y}|\bar{x}_m)} \geqslant 1 \right\} \qquad \text{(12b)}$$

or

$$\Lambda_{mm'} = \sum_{\bar{y}} f(\bar{y})\, p(\bar{y}|\bar{x}_m) \qquad (13)$$

where

$$f(\bar{y}) = \begin{cases} 1, & \text{for } y \in \Lambda_{mm'} \\ 0, & \text{otherwise} \end{cases} \qquad (14)$$

Now

$$f(\bar{y}) = \begin{cases} 1 \leqslant \sqrt{\dfrac{p'_m\, p(\bar{y}|\bar{x}'_m)}{p_m\, p(\bar{y}|\bar{x}_m)}}, & \text{for } y \in \Lambda_{mm'} \\ 0 \leqslant \sqrt{\dfrac{p'_m\, p(\bar{y}|\bar{x}'_m)}{p_m\, p(\bar{y}|\bar{x}_m)}}, & \text{otherwise} \end{cases} \qquad (15)$$

where the first part of the inequality follows from the definition of $\Lambda_{mm'}$ and the second is trivial.

Thus, for all $\bar{y}$,

$$f(\bar{y}) \leqslant \sqrt{\frac{p'_m\, p(\bar{y}|\bar{x}'_m)}{p_m\, p(\bar{y}|\bar{x}_m)}} \qquad (16)$$

Using this in Eq. (13) gives

$$p_E(m \to m') \leqslant \sum_{\bar{y}} \sqrt{\frac{p'_m}{p_m}\, p(\bar{y}|\bar{x}'_m)\, p(\bar{y}|\bar{x}_m)} \qquad (17)$$

Thus, the only difference between this inequality and the corresponding one for ML decoding (Eq. R3), i.e., the usual Bhattacharya bound, is the extra factor $\sqrt{p'_m/p_m}$ in Eq. (17). This bound is tighter than the usual Bhattacharya bound whenever the decoded message m' is less probable than the one sent, m, i.e., if

$$p'_m \leqslant p_m \quad \ldots (17)'$$

B. MAP Decoding Error Bound

This bound is derived using the union bound and the extended Bhattacharya bound (Eq. 17). The fact that the *a priori* probabilities of the various possible messages are unequal changes the locations of the MAP receiver decision

boundaries in signal space but the expression for the union bound is unchanged. Thus, we get, for the message m actually being sent, that the total probability of error considering all other possible messages is

$$p_{Em} \leqslant \sum_{m' \neq m} p_E(m \to m') \tag{18}$$

$$\leqslant \sum_{\bar{y}} \sum_{m' \neq m} \sqrt{\frac{p'_m}{p_m} p(\bar{y}|\bar{x}_{m'}) p(\bar{y}|\bar{x}_m)} \tag{19}$$

after using Eq. (17).

Now for memoryless channels we can show that (see Appendix)

$$p_E(m \to m') \leqslant \sqrt{\frac{p'_m}{p_m}} \left\{ \sum_y \sqrt{p_1(y) p_0(y)} \right\}^{w_{mm'}}$$

$$\triangleq \sqrt{\frac{p'_m}{p_m}} Z^{w_{mm'}} \tag{20a}$$

where $w_{mm'}$ is the Hamming distance between messages m and m', and $p_1(y)$, $p_0(y)$ are the probabilities of receiving a given value of a bit y given that the corresponding transmitted bit is a 1 or 0, respectively. Here

$$Z \triangleq \sum_y \sqrt{p_1(y) p_0(y)} \tag{20b}$$

which, for the special case of a BSC, reduces to

$$Z = Z_{\text{BSC}} = \sqrt{4 p_e (1 - p_e)} \tag{20c}$$

We now apply these equations to the calculation of upper bounds on event and bit error probabilities. We know that a necessary condition for an event error to begin occurring at node j is that an incorrect path diverging from the correct one at node j accumulates higher total metric than the correct one over the unmerged segment. If we denote by Γ the set of all such incorrect paths when the input message is m, using the union bound as in Eq. (18), we get, for the probability of an event error occurring at node j with m input,

$$p_j(e|m) \leqslant \sum_{m' \in \Gamma} \sqrt{\frac{p'_m}{p_m}} Z^{w_{mm'}}$$

$$\leqslant \sum_{m' \neq m} \sqrt{\frac{p'_m}{p_m}} Z^{w_{mm'}} \tag{21}$$

In the second inequality above, the summation is over all $m' \neq m$.

We can bound the probability of bit error for the input message m, $p(b|m)$, by weighting each of the terms in the sum in Eq. (21) by the number of bit errors $i_{mm'}$ occurring in choosing that incorrect path m', i.e.,

$$p(b|m) \leqslant \sum_{m' \neq m} \sqrt{\frac{p'_m}{p_m}} i_{mm'} Z^{w_{mm'}} \tag{22}$$

Accounting for all possible transmitted messages m, the overall BEP is given by

$$p_b(\text{MAP}) = \sum_m p(b|m) p_m$$

$$\leqslant \sum_m \sum_{m' \neq m} \sqrt{p'_m p_m}\, i_{mm'} Z^{w_{mm'}} \tag{23}$$

Equations (22) and (23) can, in theory, be evaluated for a specific code, but, except for certain cases, as in those of Subsection III-F, are very cumbersome in practice.

We cannot make further simplifications in Eqs. (22) and (23), directly at least, as we can in the usual ML decoding case, because each sum in the bound in Eq. (22) depends on the particular message m and the incorrect message m'. For certain specific messages m it is possible to extend the generating function method (as we do in Subsection III-F) to obtain elegant expressions for $p(b|m)$, but that is not sufficient to enable calculation of the overall BEP $p_b(\text{MAP})$, since all messages do not have the same error probabilities.

C. Comparison of MAP and ML Decoding Error Bounds

If the usual Viterbi or ML decoding were used for the same system, we would have, for the bit error probability given that message m is sent,

$$p(b|m) \leqslant \sum_{m' \neq m} Z^{w_{mm'}} i_{mm'} \tag{24}$$

where $w_{mm'}$ and $i_{mm'}$ are the same as in Eq. (22). The overall BEP would be

$$p_b(\mathrm{ML}) = \sum_m p(b|m) p_m$$

$$\leqslant \sum_m \sum_{m' \neq m} p_m i_{mm'} Z^{w_{mm'}} \tag{25}$$

Now Eq. (23) and Eq. (25) both involve a double summation over all possible pairs of messages m, m'. We compare these term by term. Consider a pair of messages m_1 and m_2 with probabilities p_{m_1}, p_{m_2}. Their contributions to the sums in Eqs. (23) and (25) are

$$C_{\mathrm{MAP}} = Z^{w_{m_1 m_2}} i_{m_1 m_2} \left(\sqrt{p_{m_1} p_{m_2}} + \sqrt{p_{m_1} p_{m_2}} \right)$$

$$= 2 \sqrt{p_{m_1} p_{m_2}} Z^{w_{m_1 m_2}} i_{m_1 m_2} \tag{26}$$

and

$$C_{ML} = \left(p_{m_1} + p_{m_2} \right) Z^{w_{m_1 m_2}} i_{m_1 m_2} \tag{27}$$

Comparing Eqs. (26) and (27), we see that

$$C_{\mathrm{MAP}} \leqslant C_{\mathrm{ML}} \tag{28}$$

for all p_{m1}, p_{m2}, with equality holding for $p_{m1} = p_{m2}$. Thus, we get

(MAP) error upper bound $\leqslant$ (ML) error upper bound (29)

Since the upper bounds in Eqs. (23) and (25) are asymptotically tight, we expect from Eq. (29) that, for low BEP where the bounds are close to the actual error probabilities, the new metric is better than, or as good as, the conventional one. Also, on general principles, we know that MAP decoding is always at least as good as ML decoding.

D. Approximate Magnitude of Improvement Expected From Upper Bound Expressions

From Eqs. (23) and (25) we may obtain an order-of-magnitude estimate of the improvement in BEP using MAP decoding. We assume a low enough BEP so that the actual BEP is replaceable by the upper bounds in Eqs. (23) and (25). Since the BEP is low, we must have Z small, so that the terms in Eqs. (23) and (25) with factors $Z^{w_{mm'}}$, where $w_{mm'}$ is greater than the free distance d_f of the code, are negligible compared to the terms with factors Z^{d_f}. Thus, we may retain only these terms and have

$$\frac{p_b(\mathrm{ML})}{p_b(\mathrm{MAP})} \approx \frac{\sum_{mm' \in \Gamma} p_m Z^{d_f} i_{mm'}}{\sum_{mm' \in \Gamma} \sqrt{p_m p_{m'}} Z^{d_f} i_{mm'}}$$

$$= \frac{\sum_{mm' \in \Gamma} p_m i_{mm'}}{\sum_{mm' \in \Gamma} \sqrt{p_m p_{m'}} i_{mm'}} \tag{30}$$

where Γ is the set of path pairs m, m' with Hamming distance equal to d_f between them.

Further simplification is possible only if we know the probabilities of the message pairs that are at the free distance from each other and this is specific to the code and the source.

E. Further Approximation for First-Order Markov Source

In this section we make an order-of magnitude estimate of the improvement in BEP for the specific case of a first-order Markov source with low transition probability p_s. For such a source with low p_s, there is a lot of redundancy in the message output, and we may expect the MAP method to be most useful in this case. In making our estimate of the improvement in BEP in this section, we make the following assumptions:

(1) We assume that the most probable message transmitted, m_1, has an *a priori* probability so close to 1 that the overall BEP p_b(MAP) may be closely approximated by the BEP given m_1, $p(b|m_1)$.

(2) Given that an error is made in decoding this message m_1, we assume that the BEP given m_1 can be calculated by considering only those possible erroneously decoded messages that are closest in distance to m_1. In doing this, we are again using the arguments leading to Eq. (30) in Subsection III-D.

(3) Among these closest-distance erroneous messages, we further restrict our attention to the one that has the highest *a priori* probability. There may be more than one such message possible but we ignore this small factor in getting our order-of-magnitude estimate.

Thus, assumption (1) enables us to consider only one transmitted message m_1 and assumptions (2) and (3) enable us to narrow down the set of possible erroneously decoded messages (given that m_1 is sent) to only one message, which we call m_2.

Since the first-order Markov source has a low transition probability p_s, the most probable messages it puts out will consist of long strings of zeros and long strings of ones with occasional (with probability p_s) transitions from one type of string to the other. We now make the following assumption:

(4) We assume that the most probable message m_1 has long enough runs of zeros and ones that we can replace it, for the purpose of calculating BEPs, by one with no transitions at all, say, the all-zero message.

Now the messages closest to m_1 will differ from it in only a small number of bits (like 1 or 2 depending on the specific code). We can see that the most probable of these, i.e., m_2, will have at least 2 transitions in it. For instance, the relevant portions of m_1 and m_2 may be -000- and -010-, or -0000- and -0110-. Thus, for the specific code discussed in Section IV, if an error occurs in decoding a portion -000- of a message, and if the erroneously decoded message is at the free distance from the correct one, it occurs because -000- is decoded as -010-.

We may now calculate the improvement in BEP considering only the messages m_1 and m_2. For symmetry, we include in p_b(MAP), both $p(b|m_1)$ and $p(b|m_2)$, where the conditional bit error probability for each message is calculated assuming the other one as being the erroneously decoded message. We do this using the method of Subsection III-D, and get

$$\frac{p_b\,(\mathrm{ML})}{p_b\,(\mathrm{MAP})} \approx \frac{p_{m1} + p_{m2}}{2\sqrt{p_{m1}\,p_{m2}}} \tag{31}$$

Defining

$$q_s = 1 - p_s \tag{32}$$

this becomes

$$\frac{p_b\,(\mathrm{ML})}{p_b\,(\mathrm{MAP})} \approx \frac{q_s}{2p_s} + \frac{p_s}{2q_s} \tag{33}$$

For instance, if $p_s = 10^{-3}$, we have from Eq. (33),

$$\frac{\text{old BEP}}{\text{new BEP}} \approx 499.5 \tag{34}$$

Thus, there is about a factor of 500 improvement in BEP in this case using the new metric.

F. Generating Function Method for Low p_S Case

In this section, we show how to calculate the improvement in BEP more exactly than we did in Subsection III-E. The method to be described holds for all p_s if the only message considered is the all-zeros or the all-ones message and also possibly for certain specific messages depending on the code. For low p_s, the most probable messages consist of strings of zeros and ones with occasional transitions between strings so that this analysis holds for most of the probable message sequences.

We have for 2 messages m, m',

$$p_E(m \to m') \leqslant \sqrt{\frac{p_{m'}}{p_m}}\; Z^{w_{mm'}} \tag{20}$$

which differs from the corresponding equation for the conventional decoder only in the factor $\sqrt{p_{m'}/p_m}$. Here, p_m is the probability of the transmitted message sequence in consideration and p'_m is the probability of the erroneously decoded message m'. We can express p_m and p'_m using Eq. (5) for the special case of the first-order Markov source as

$$p_m = p_{mB}p_{m(B-1)}\cdots p_{m1} \tag{35}$$

$$p_{m'} = p_{m'B}p_{m'(B-1)}\cdots p_{m'_1} \tag{36}$$

where

$$p_{mj} = p\left(u_{mj}|u_{m(j-1)}\right) \tag{35a}$$

$$p_{m'j} = p\left(u_{m'j}|u_{m'(j-1)}\right) \tag{36a}$$

So we have

$$p_E(m \to m') \leqslant Z^{w_{mm'}} \sqrt{\frac{p_{m'B}}{p_{mB}}\cdots\frac{p_{m'1}}{p_{m1}}} \tag{37}$$

We can then rewrite Eqs. (21) and (22) as

$$p_j(e|m) \leqslant \sum_{m' \neq m} Z^{w_{mm'}} \sqrt{\frac{p_{m'B}}{p_{mB}} \cdots \frac{p_{m'1}}{p_{m1}}} \tag{21}'$$

$$p(b|m) \leqslant \sum_{m' \neq m} i_{mm'} Z^{w_{mm'}} \sqrt{\frac{p_{m'B}}{p_{mB}} \cdots \frac{p_{m'1}}{p_{m1}}} \tag{22}'$$

Now each of the factors $\sqrt{p_{m'i}/p_{mi}}$ in Eq. (37) corresponds to a particular branch in the trellis. If the input message is such (e.g., the all-ones or the all-zeros message) that all possible incorrect paths can be accounted for by tracing varying numbers of loops on a diagram obtained by opening up the state diagram at some node, then we know that all the paths at various distances from the correct one can be expressed by terms of the series expansion of the generating function $T(D)$ (§4.3 of Ref. 2). Here, D has the usual meaning that its exponent for any branch represents the distance of that branch from the corresponding code subvector of the input message. In order to weight each of these terms by the factor $\sqrt{p_{m'i}/p_{mi}}$, all we need to do is to associate with each branch of the opened-up state diagram the appropriate factor $\sqrt{p_{m'i}/p_{mi}}$ and then evaluate the resulting generating function which we denote by $T_m(D)$. We then set $D = Z$ in the expression $T_m(D)$ to evaluate the bound in Eq. (21)'. Similarly, to get the bound in Eq. (22)', we obtain the generating function $T_m(D,I)$, where I has the usual meaning that $I = 1$ for a given branch if its bit differs from the corresponding bit in the input message and $I = 0$ otherwise, and we set $D = 2$, $I = 1$ in $\partial T_m/\partial I\,(D,I)$ to evaluate the bound in Eq. (22)'. This generating function method allows us to take into account all possible erroneous paths and their error contributions rather than just one as in Subsection III-E.

In the special case of the first-order Markov source, we get an even simpler expression for event and bit error probability bounds. Each of the p_{mi} and $p_{m'i}$ is either p_s or q_s in this case. So each term in the sums in Eqs. (21)' and (22)' has a factor of M^k evaluated at $M = \sqrt{p_s/q_s}$, where k is an integer. This integer can be determined as follows. We first associate with each branch of the opened-up state diagram either M or 1 or M^{-1}, depending on whether $p_{m'i}/p_{mi} = p_s/q_s$ or 1 or q_s/p_s for the particular branch. Thus, for the all-zeros or the all-ones message, we write M for branches corresponding to a transition in the message and 1 for branches with no transition, since in these cases, p_{mi} is always equal to q_s and $p_{m'i}$ can be either p_s or q_s. We then evaluate the transfer function $T(D,M)$ (this takes the place of the transfer function $T_m(D)$ we had in the previous paragraph for the higher order Markov cases) and the transfer function $T(D,M,I)$ (to take the place of $T_m(D,I)$ of the previous paragraph). We can then see that the event error probability at any node, say, j, is given by

$$p_j(e|m) \leqslant T(D,M)\Big|_{\substack{M = \sqrt{\frac{p_s}{q_s}} \\ D = Z}} \tag{38}$$

and the BEP given m is given by

$$p(b|m) \leqslant \frac{\partial T}{\partial I}(D,M,I)\Big|_{\substack{D = Z \\ I = 1 \\ M = \sqrt{\frac{p_s}{q_s}}}} \tag{39}$$

It is not easy to generalize these results to get the overall BEP considering all possible messages, because the functions $T_m(D)$, $T_m(D,I)$ here are message-dependent unlike the usual Viterbi decoding case. Thus, for non-repetitive messages, a general expression like those in Eqs. (38) and (39) that identifies all other possible paths and their error contributions are not easy to obtain. A slight generalization of one method is possible for repetitive messages like, say, ... 1000, 1000, 1000 ... for the code considered in the next section, by opening up, not just one node, but a path in the state diagram corresponding to the message unit repeated. But even in this case, it is necessary to consider each possible erroneously decoded message individually and obtain its contribution to the BEP.

As we have stated before, if p_s is low, then just considering the all-zeros or all-ones message should be good enough to obtain an estimate of the overall BEP.

IV. Illustration With a Specific Code

A. The Code

We assume a BSC, a first-order Markov source with transition probability p_s and a specific $K = 3$, $b = 1$, $n = 2$ (rate 1/2) code. The encoder is shown in Fig. 1, and the state diagram is shown in Fig. 2. Bits come in at the left in Fig. 1 and the states (written in boxes at the nodes in Fig. 2) are represented by the rightmost 2 bits of the encoder. The dotted and solid lines in Fig. 2 correspond to a 1 and a 0 input bit, respectively.

B. Calculations

The state diagram with the 00 node opened up and the branches marked with the appropriate powers of D,I,M are shown in Fig. 3. This diagram thus corresponds to the all-zeros input message.

The generating function is

$$T(D,I,M) = \frac{D^5 IM^2}{1 - DI - DIM^2} \tag{40}$$

so that the BEP given $m = \ldots 00 \ldots$ is

$$p(b|m)|_{MAP} \leq \left.\frac{\partial T}{\partial I}\right|_{\substack{I=1 \\ M=\sqrt{\frac{p_s}{q_s}} \\ D=Z}}$$

$$= \left.\frac{D^5 M^2}{(1 - D - DM^2)^2}\right|_{\substack{M=\sqrt{\frac{p_s}{q_s}} \\ D=Z}} \tag{41a}$$

$$= D^5 M^2 \left\{1 + 2D(1+M^2) + 3D^2(1+M^2)^2 + 4D^3(1+M^2)^3 + \cdots\right\}\Big|_{\substack{M=\sqrt{\frac{p_s}{q_s}} \\ D=Z}} \tag{41b}$$

For the case of the conventional ML decoder, we would obtain a similar expression with $M = 1$, i.e.,

$$p(b|m)_{ML} \leq \left.\frac{D^5}{(1 - 2D)^2}\right|_{D=Z} \tag{42a}$$

$$\leq D^5 \{1 + 2D \cdot 2 + 3D^2 \cdot 2^2 + 4D^3 \cdot 2^3 + \cdots\}\Big|_{D=Z} \tag{42b}$$

The expansion of Eq. (41b) shows that the path with minimum distance (=5) has 2 transitions and one bit error, corresponding to the term D^5M^2; of the two paths with distance 6, one has 2 transitions and the other 4, and each of these has 2 bit errors corresponding to the $2D^5 \cdot M^2 \cdot D(1 + M^2)$ term, and so on.

From the expansions of Eqs. (41b) and (42b), we see that if $M^2 < 1$, i.e.,

$$p_s < 0.5 \tag{43}$$

the bound of Eq. (41b) is smaller than that of Eq. (42b) by a factor of more then M^2, since each term in the brackets in Eq. (41b) is smaller than the corresponding one in brackets in Eq. (42b).

In our example, we assume $M^2 < 1$. If we assume the bounds in Eqs. (41) and (42) to be close to the actual BEP because of our $p_s << 1$ assumption, there is an improvement in BEP by more than a factor of $M^2 = p_s/q_s$. Thus, for $p_s = 0.1$, we have that the new BEP is lower than the old BEP by at least a factor of 1/9.

For the all-ones input message, the 11 node can be opened up. If the details are worked out, we obtain, as expected, exactly the same expression as in Eq. (41).

C. The Simulations

We simulated the encoder and decoder for the code described above and ran several messages of 298-bit block lengths for various sets of parameters p_s and p_e, with both p_s, $p_e < 0.5$. Initially, we checked the program with just a block of zeros or just a block of ones as input and, later, generated the input messages themselves within the program by including a simulation of the Markov source. In this case the input messages are realistic ones and were found to consist of strings of zeros and ones as expected. The simulations show that the BEP with the new metric is always lower (for p_s, $p_e < 0.5$) than with the conventional metric. But a larger number of simulations would be needed to obtain the exact amount of improvement, especially for the cases with low BEP. However, the results shown in Table 2 clearly demonstrate the improvement. We also see that for low enough p_e, the improvement in BEP is by a factor $> p_s/q_s$, as predicted by Eqs. (41) and (42).

Another fact that was indicated by the simulations (see Table 1), but which again needs further verification, is that the decoder with the new metric needs a trellis truncation depth of at least 20 constraint lengths for negligible truncation error, as opposed to the 5 constraint lengths required for the old decoder (Ref. 3). The truncation scheme we are talking about

here is the simple one described in §4.7 of Ref. 2, in which, as each set of b bits enters the registers of each state, the b bits which entered 20K branches earlier are removed, after the decoder has made a final decision on these bits by setting them equal to the appropriate survivor bits of an arbitrary state. In obtaining the results described in Table 2, we used a 40K truncation length rather than 20K, to be on the safe side.

D. Channel Capacity Limitations

From the results of the simulations (Table 2) we see that for given p_s, if the probability of channel transition p_e is high, the BEP is high for both the new and the old decoding algorithm, and the ratio of the new BEP to the old BEP increases with p_e. Similarly, for a given p_e, this ratio increases as p_s increases. For the given source and channel, we can calculate the source entropy and the channel capacity, which are given, respectively, by

$$H(p_s) = -p_s \log_2 p_s - (1 - p_s) \log_2 (1 - p_s) \tag{44}$$

and

$$\begin{aligned} C(p_e) &= 1 - H(p_e) \\ &= 1 + p_e \log_2 p_e + (1 - p_e) \log_2 (1 - p_e) \end{aligned} \tag{45}$$

Figure 4 shows the nature of these two curves, on the same plot for convenience.

Since the channel capacity represents the upper limit on the reliable communication rate, we should expect that no scheme would allow any substantial reduction in BEP above channel capacity. Thus, if we set

$$H(p_s) \leqslant C(p_e) \tag{46}$$

we can get the outer limits on the regions of p_e and p_s within which the new scheme can be expected to be useful.

We need, from Eqs. (44) through (46),

$$\begin{aligned} &-p_e \log_2 p_e - (1 - p_e) \log_2 (1 - p_e) - p_s \log_2 p_s \\ &- (1 - p_s) \log_2 (1 - p_s) \leqslant 1 \end{aligned} \tag{47}$$

or

$$(p_e)^{p_e} (1 - p_e)^{(1-p_e)} (p_s)^{p_s} (1 - p_s)^{(1-p_s)} \geqslant 0.5 \tag{48}$$

Some sets of parameters p_e and p_s satisfying this relationship with the equality sign are marked with an * in Table 2. Figure 5 shows a plot of old BEP, new BEP, and the ratio of new BEP to old BEP obtained from simulations with the Markov source model included in the program. Two sets of curves are shown: one set has $p_s = 0.1$ and p_e varying, the other has $p_e = 0.1$ and p_s varying. Since an insufficient number of simulations were made, however, the shape of the curve is not very reliable. But it does show an increase of the ratio of new BEP to old BEP as we approach and exceed channel capacity.

E. Comparison of Calculations and Simulations

None of the simulations were made for the very low values of p_e (like 10^{-5} or 10^{-6}) that would be required for the upper bounds of Eqs. (41) and (42) to be approached, since too many blocks of input would be required. Even for the case of $p_s = 0.1$ and $p_e = 0.03$, which represents about the lowest data rate to channel capacity ratio in the simulations, there were 0 bit errors with the new decoder in the 10 blocks of 298 bits used and only 1 bit error with the conventional decoder in the same 10 blocks, corresponding to an old BEP of 0.000335, and an even lower new BEP. Here the bounds of Eqs. (41) and (42) give, respectively,

$$\begin{aligned} p_b(\text{ML}) &\leqslant 0.046 \\ p_b(\text{MAP}) &\leqslant 0.001 \end{aligned} \tag{49}$$

which are both much higher than the actual BEP obtained by simulation. Hence direct verification of the bounds of Eqs. (41) and (42) has not been performed in the simulations, but the bounds are probably very useful in calculating BEPs at rates well below capacity where too many simulations would be needed if computer calculation of BEPs were to be performed.

V. Conclusions

We have shown that a simple modification of the metric used in the Viterbi decoding algorithm achieves MAP decoding for sources with memory. It can cause a noticeable reduction in BEP for sources with strong correlation between bits (i.e., low entropy). Analysis methods to obtain upper bounds on the BEP obtainable with the new metric have been given. The simulations performed for a BSC and a first-order Markov source verify that an improvement occurs, but to obtain the exact ratio of improvement, more simulations are needed. We can conclude, however, that the ratio of new BEP to old BEP for a first-order Markov source and a BSC with very low p_e is at least p_s/q_s and that the extent of improvement reduces as channel capacity is approached and exceeded. Simulations also

indicate that a trellis truncation depth of 20 constraint lengths is probably needed for reliable results with the new metric.

The modification of the metric required for the case of an arbitrary memoryless Gaussian channel is given. The entire analysis given holds for such a Gaussian channel, but the simulations of the decoder have been performed only for the case of the BSC. Except in cases where the first-order Markov source is specified, the analysis holds for any Markov source of arbitrary order. More work is needed to extend the results to more realistic sources like video data sources. Ultimately, the aim is to devise a reasonably simple modification of the Viterbi decoding algorithm that can make use of the correlation between neighboring pixels in video data of scenes to enable more efficient channel use than is possible with the normal Viterbi decoder.

Acknowledgment

I would like to thank Dr. Edward C. Posner of the Jet Propulsion Laboratory, Pasadena, for suggesting the problem and for his valuable comments and encouragement throughout the work.

References

1. Mark, J. W., "Adaptive Trellis Encoding of Discrete-Time Sources with a Distortion Measure," *IEEE Trans. Comm. Technol.*, Vol. COM-25, pp. 408-417, 1977.

2. Viterbi, A. J., and Omura, J. K., *Principles of Digital Communication and Coding*, McGraw-Hill, NY, 1979.

3. Heller, J. A., and Jacobs, I. M., "Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. Comm. Technol.*, Vol. COM-19, pp. 835-848, 1971.

4. Forney, G. D., Jr., "The Viterbi Algorithm," *Proc. IEEE*, Vol. 61, pp. 268-278, 1973.

5. Neuhoff, D. L., "The Viterbi Algorithm as an Aid in Text Recognition," *IEEE Trans. Info. Theory*, Vol. IT-21, pp. 222-226, 1975.

6. Rosenberg, A. E., and Schmidt C. E., "Automatic Recognition of Spoken Spelled Names for Obtaining Directory Listings," *BSTJ*, Vol. 58, pp. 1797-1824, 1979.

Table 1. Simulations described in Subsection IV-C to estimate effect of trellis truncation (all runs here have $p_s = 0.1$, $p_e = 0.4$)

Program	Truncation length, units	Number of 298-bit blocks	Conventional BEP	New BEP	Remarks
01	10K	1	0.4866	0.5336	Programs 01 through 04 show that 20K truncation length is needed.
02	10K	1	0.4832	0.5537	
03	20K	1	0.4899	0.4698	
04	30K	1	0.4899	0.4698	Same sets of random numbers used for the source and channel in 04 as in 03; otherwise different sets used for different blocks of data for both source and channel simulations.
05	30K	5	0.5080	0.4170	
06	40K	1	0.4765	0.2416	
07	40K	1	0.5101	0.4060	
08	40K	5	0.5000	0.3698	

Table 2. Simulations described in Subsection IV-C (all with truncation length 40K)

Program	p_s	p_e	Number of 298-bit blocks	Conventional BEP	New BEP	Ratio $\frac{\text{New BEP}}{\text{Conventional BEP}}$	Remarks
1	0.1	0.4	5	0.5000	0.3698	0.7396	All runs have different sets of random numbers for the source and different sets for the channel.
2	0.1	0.3	3	0.4508	0.2573	0.5708	
3	0.1	0.2	3	0.3367	0.09396	0.2791	
4*	0.1	0.12	10	0.1117	0.02886	0.2583	
5	0.1	0.08	12	0.02041	0.005593	0.2740	
6	0.1	0.05	15	0.006714	0.0004474	0.0667	
7	0.1	0.03	10	0.000335	0	0	
8*	0.12	0.1	10	0.06146	0.013194	0.2147	
9	0.05	0.1	5	0.07651	0.01342	0.1754	
10	0.03	0.1	10	0.05570	0.002013	0.03614	
11*	0.2	0.04	10	0.003691	0.001678	0.4546	

*Means p_s, p_e for that program define operation at channel capacity.

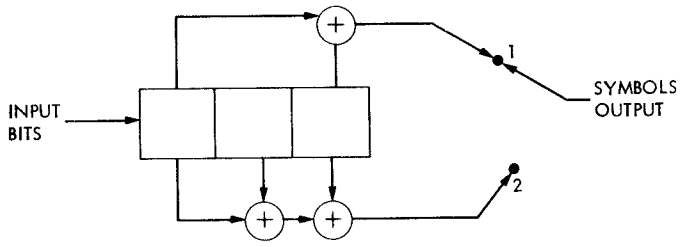

Fig. 1. A $K = 3$ encoder

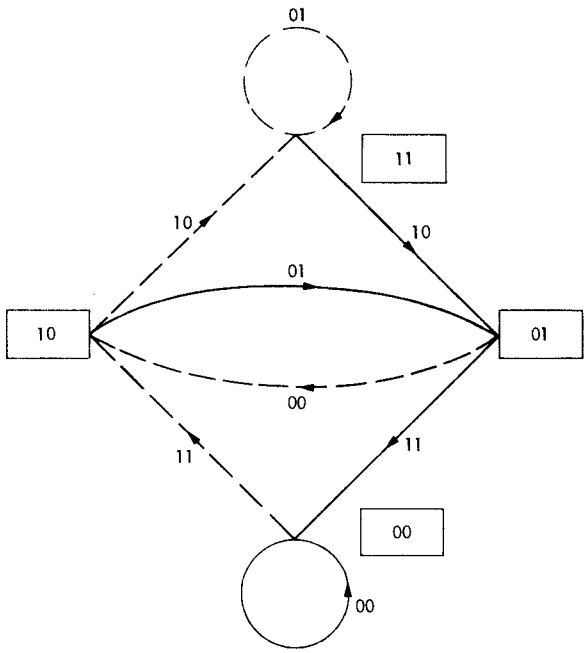

Fig. 2. State diagram for encoder of Fig. 1

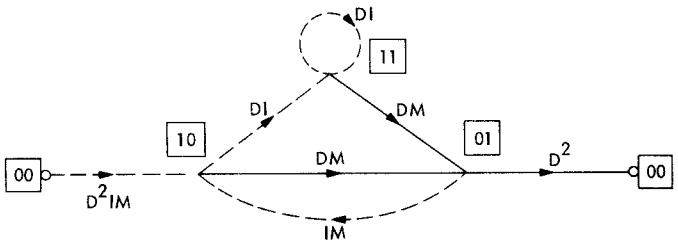

Fig. 3. State diagram labeled with distance, bit errors, and branch probability ratios relative to all-zeros input

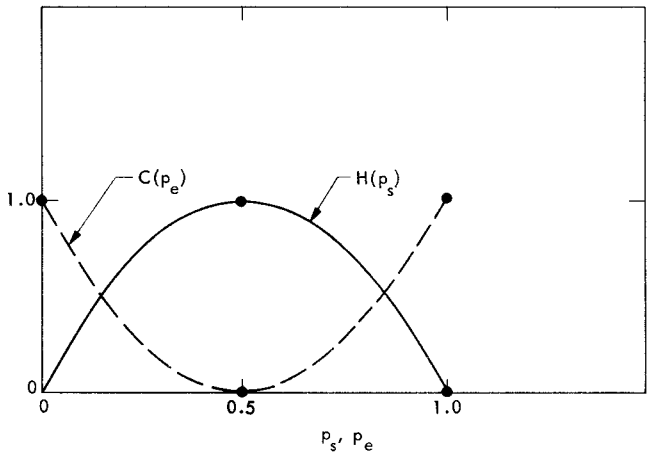

Fig. 4. Source entropy vs p_s and channel capacity vs p_e

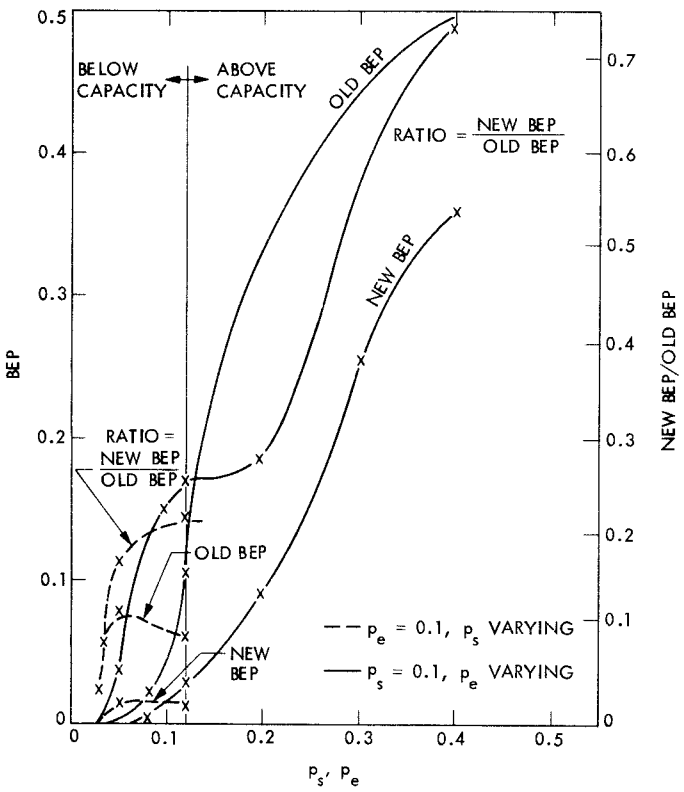

Fig. 5. BEP with new and old metrics and their ratio vs p_e with p_s constant and vs p_s with p_e constant (from Table 2)

Appendix

Simplifying the Extended Bhattacharya Bound for Memoryless Channel

We show here the derivation of Eq. (20) from Eq. (17) for a memoryless channel. The method used is similar to that in §2.9 of Ref. 2. We have, from Eq. (17), since $p(\bar{y}|\bar{x}_m)$ and $p(\bar{y}|\bar{x}_{m'})$ can be factored for a memoryless channel,

$$p_E(m \to m') \leqslant \sum_{\bar{y}} \sqrt{\frac{p_{m'}}{p_m} p(\bar{y}|\bar{x}_{m'}) p(\bar{y}|\bar{x}_{m'})} \tag{17}$$

$$= \sqrt{\frac{p_{m'}}{p_m}} \prod_{j=1}^{nB} \sum_{y} \sqrt{p(y|u_{mj}) p(y|u_{m'j})} \tag{A1-1}$$

where u_{mj}, $u_{m'j}$ are the input bits corresponding to the code subvectors $\bar{x}_{mj}$, $\bar{x}_{m'j}$; the sum in Eq. (17) runs over all possible $\bar{y}$ vectors consisting of B n-dimensional subvectors, and the sum over y in Eq. (A1-1) runs over the two possibilities 0 and 1 for each of these nB components.

We have

$$p_E(m \to m') \leqslant \sqrt{\frac{p_{m'}}{p_m}} \prod_{j=1}^{nB} \sum_{y} \sqrt{p(y|u_{mj}) p(y|u_{m'j})} \tag{A1-l}$$

$$= \sqrt{\frac{p_{m'}}{p_m}} \prod_{j:u_{m'j}=u_{mj}} \sum_{y} \sqrt{p(y|u_{mj}) p(y|u_{m'j})}$$

$$\cdot \prod_{j:u_{m'j}=\bar{u}_{mj}} \sum_{y} \sqrt{p(y|u_{mj}) p(y|u_{m'j})}$$

(here, $\bar{u}_{mj}$ = logical complement of u_{mj}). This bound can be written as

$$p_E(m \to m') \leqslant \sqrt{\frac{p_{m'}}{p_m}} \prod_{j:u_{m'j}=u_{mj}} \sum_{y} \sqrt{p(y|u_{mj}) p(y|u_{m'j})} \tag{A1-2}$$

since each sum in the first product in the previous equation is equal to 1.

Thus, if the messages m, m' differ in $w_{mm'}$ bits, we get

$$p_E(m \to m') \leqslant \sqrt{\frac{p_{m'}}{p_m}} \left[\sum_{y} \sqrt{p(y|1) p(y|0)} \right]^{w_{mm'}}$$

$$\triangleq \sqrt{\frac{p_{m'}}{p_m}} \left[\sum_{y} \sqrt{p_1(y) p_0(y)} \right]^{w_{mm'}}$$

$$\triangleq \sqrt{\frac{p_{m'}}{p_m}} Z^{w_{mm'}} \tag{20}$$