H. Kobayashi

Application of Probabilistic Decoding to Digital Magnetic Recording Systems

Abstract: A digital magnetic recording system is viewed in this paper as a linear system that inherently includes a correlative level encoder. This encoder can be regarded as a linear finite-state machine like a convolutional encoder. The maximum likelihood decoding method recently devised by Viterbi to decode convolutional codes is then applied to digital magnetic recording systems. The decoding algorithm and its implementation are discussed in detail.

Expressions for the decoding error probability are obtained and confirmed by computer simulations. It is shown that a significant improvement in the performance with respect to other methods is achievable by the maximum likelihood decoding method. For example, under the Gaussian noise assumption the proposed technique can reduce raw error rates in the 10^{-3} to 10^{-4} range by a factor of 50 to 300. These results indicate that the maximum likelihood decoding method gains as much as 2.5 dB in signal-to-noise ratio over the conventional bit-by-bit detection method.

1. Introduction

In an earlier paper [1] it was shown that a digital magnetic recording channel can be viewed as a partial-response channel. The partial-response signalling or the correlative level coding is a technique recently developed by Lender [2], Kretzmer [3], van Gerwen [4] and by others in data communication systems, in which a controlled amount of intersymbol interference is intentionally introduced to improve the information rate [5]. In a digital magnetic recording system, on the other hand, the differentiation operation inherent in the read-back process generates, in effect, a correlative level coded sequence. Since the representation of a digital magnetic recording channel in terms of its equivalent partial-response channel is essential to the development of the present paper, a brief review of some earlier results [1, 6, 7] is given.

In the ordinary digital magnetic recording system saturation recording is performed, i.e., two stable states of magnetization represent binary data to be stored. Let $\{a_k\}$ represent an information sequence of "0" and "1" to be recorded on the magnetic surface. The magnetization pattern $m(t)$ recorded by the NRZ (Non-Return-to-Zero) method is representable as

$$m(t) = \sum_{k=0}^{\infty} (2a_k - 1)u(t - kT) - 1(-t), \tag{1}$$

where $u(t)$ is a rectangular pulse of duration T seconds:

$$u(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{elsewhere}, \end{cases} \tag{2}$$

and $1(t)$ is a unit step function:

$$1(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0. \end{cases} \tag{3}$$

Here the amplitude of $m(t)$ is normalized by its saturation levels, i.e., $+1$ and -1 represent two saturation levels corresponding to "1" and "0" of the sequence $\{a_k\}$. We assume in Eq. (1) that $m(t) = -1$ for $t < 0$, i.e., the magnetic surface has been magnetized to the -1 saturation level before the arrival of data stream $\{a_k\}$.

In the read-back process the relationship between the output voltage $e(t)$ and magnetization pattern $m(t)$ is given by

$$e(t) = \left[\frac{d}{dt}\, m(t)\right] * h(t), \tag{4}$$

where * means convolution and $h(t)$ represents the magnetic head field distribution characterized by the response due to a unit step function in $m(t)$. Figure 1 illustrates waveforms at various stages in the NRZ recording method.

On substituting Eq. (1) into (4) we obtain

$$\begin{aligned} e(t) &= h(t) * \left[\sum_{k=0}^{\infty} (2a_k - 1)\{\delta(t - kT) - \delta(t - kT - T)\} + \delta(t)\right] \\ &= 2\sum_{k=0}^{\infty} x_k h(t - kT), \end{aligned} \tag{5}$$

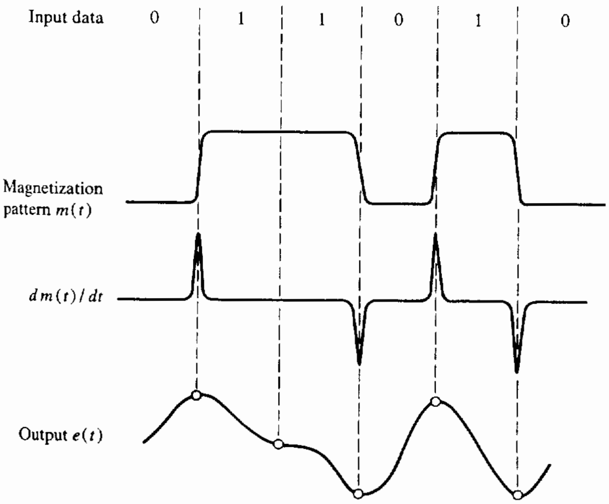


Figure 1 Waveforms at various stages in the NRZ recording system.

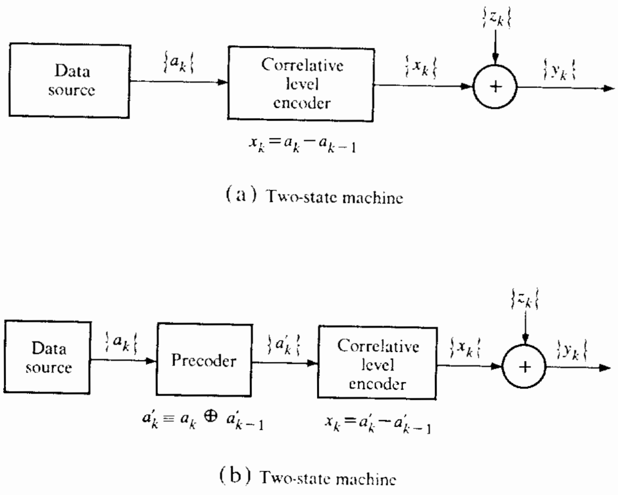


Figure 2 Discrete system representations of (a) the NRZ recording system and (b) the NRZI recording system.

where

$$x_k = \begin{cases} a_k - a_{k-1}, & k \geq 1 \\ a_0, & k = 0. \end{cases} \tag{6}$$

As seen from Eq. (6), the sequence $\{x_k\}$ is a three-level sequence of -1's, 0's and $+1$'s. Unlike the situation in data communication systems, the sequence $\{x_k\}$ per se is not generated nor clearly observed in any part of the recording system. What we actually observe is $e(t)$, a linear function of the sequence $\{x_k\}$ as shown in Eq. (5). In other words, we consider for analytical convenience that the magnetic recording channel contains some imaginary correlative level encoder as a part of the system. If $\{a_k\}$ takes on "1" and "0" equally likely and is independent from bit-to-bit, the sequence $\{x_k\}$ possesses the following statistical properties:

$$\Pr\{x_k = 0\} = \tfrac{1}{2}, \Pr\{x_k = -1\} = \Pr\{x_k = +1\} = \tfrac{1}{4} \tag{7}$$

and

$$\mathrm{E}[x_k x_l] = \begin{cases} \tfrac{1}{2}, & k = l \\ -\tfrac{1}{4}, & |k - l| = 1 \\ 0, & \text{elsewhere.} \end{cases} \tag{8}$$

Equation (8) shows that adjacent digits are highly correlated and hence $\{x_k\}$ is called a correlative level coded sequence. In other words $\{x_k\}$ is a sequence that contains redundancy.

Let $e(t)$ be passed into a linear filter $f(t)$, the output of which is denoted by $r(t)$:

$$r(t) = e(t) * f(t). \tag{9}$$

If the total response function $g(t) = h(t) * f(t)$ satisfies the condition

$$g(kT) = \delta_{k,0}, \qquad k = 0, \pm 1, \pm 2, \cdots, \tag{10}$$

then the sampled value of the filter output is

$$r(kT) = 2x_k. \tag{11}$$

Equation (10) is satisfied if the filter $f(t)$ includes an equalizer so that the effect of intersymbol interference is removed. However, the sampled voltage cannot be exactly equal to $2x_k$ because of the noise and the residue of intersymbol interference. Therefore, what we actually observe at a sampling instant is represented by the following random variable y_k:

$$y_k = x_k + z_k, \tag{12}$$

where z_k represents the total disturbance.

From Eqs. (6) and (12) we obtain the block diagram of Fig. 2(a), which is a linear discrete system representation of a magnetic recording system. Here $\{a_k\}$ is a sequence of "1" and "0", $\{x_k\}$ is a sequence of -1's, 0's and $+1$'s, whereas $\{y_k\}$ is a random sequence that may take on any real number. In earlier papers [1, 6, 7] we described a decision scheme that quantizes $\{y_k\}$ into a three-level sequence $\{q_k\}$. The data sequence $\{a_k\}$ can be estimated on the basis of this "hard" decision output $\{q_k\}$ by solving Eq. (6). However, an erroneous decision in $\{q_k\}$ would result in the propagation of error in decoding the data sequence $\{a_k\}$. To avoid such error propagation, $\{a_k\}$ is transformed into another binary sequence $\{a'_k\}$ by the following relation before being passed into a correlative level encoder:

$$a'_k = a_k \oplus a'_{k-1} \bmod 2 \tag{13a}$$

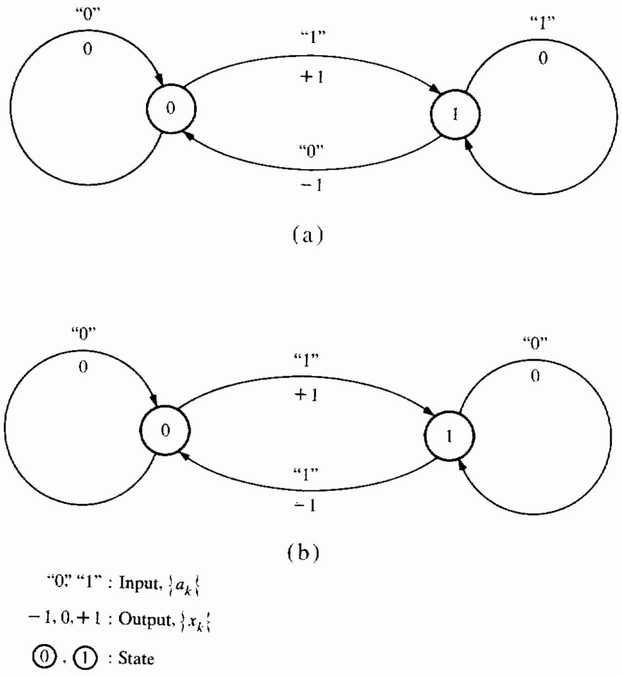
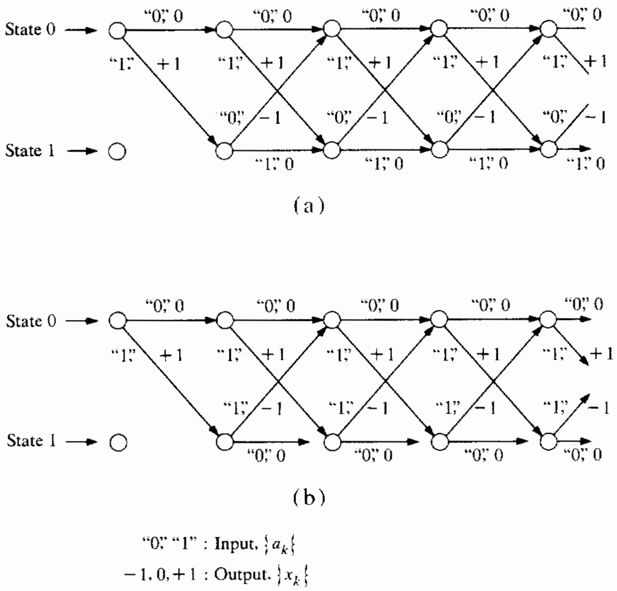


Figure 3 State transition diagrams of (a) the NRZ recording system and (b) the NRZI recording system.

Figure 4 Trellis picture representations of the state transition in (a) the NRZ recording system and (b) the NRZI recording system.



and $a'_{-1} = 0$. (13b)

This transformation is usually called precoding in data communication systems [2]. It has been shown [1] that the so-called NRZI (Non-Return-to-Zero-Inverse) recording method is equivalent to a precoding operation followed by the NRZ recording method. Figure 2(b) is an equivalent discrete system representation of the NRZI system.

An algebraic method of error detection proposed in References 1, 6 and 7 makes full use of the inherent redundancy of the three-level sequence $\{x_k\}$. This algebraic approach has been further extended to the case in which the receiver makes a "soft" decision, i.e., the number of quantization levels is increased from three to five or seven including ambiguity levels [6, 7].

The present paper describes a completely different approach to decoding the magnetic recording output. This decoding method is a very simple scheme to realize the maximum likelihood decoding (MLD) rule and is a probabilistic decoding scheme rather than the algebraic one discussed earlier. It will be shown that a significant improvement in the performance is obtainable by the proposed decoding scheme.

2. Maximum likelihood decoding

In 1967 Viterbi [8] devised a new nonsequential decoding algorithm for convolutional codes. Forney [9] showed that this algorithm is in fact the maximum likelihood decoding rule. Omura [10] discussed the algorithm in a state-space context and showed its equivalence to the dynamic programming.

A correlative level encoder can be viewed as a simple type of linear finite-state machine over the real number field as opposed to a Galois field over which a convolutional encoder is defined [11]. Now that we know the equivalence between a magnetic recording channel and a correlative level encoder, it is not difficult to show that the Viterbi decoding rule is applicable to our problem.

We define s_k, the state of the imaginary correlative level encoder by the latest encoder input, i.e., $s_k = a_k$ in the NRZ recording system and $s_k = a'_k$ in the NRZI recording system. A precoder defined by Eq. (13) is also a two-state machine; hence we can combine the precoder and correlative encoder in the state representation of the NRZI system. Figures 3(a) and (b) show the state transition diagrams of the NRZ and NRZI recording systems, respectively, where 1 and 0 in small circles represent two possible states. Each time the machine receives a new bit, "1" or "0", a state transition takes place depending on the input and the current state. Numbers −1, 0 or +1 attached to arrows represent the encoder output $\{x_k\}$. Although the diagram of Fig. 3 completely describes our system, the description in terms of the trellis picture introduced by Forney [9] will provide a better understanding of the decoding rule to be discussed.

The trellis picture of Fig. 4 shows the transition of the encoder state as a function of time t. Here the input "1" or "0" and the corresponding output −1, 0 or +1 are attached to each branch connecting two states. Starting from $s_0 = 0$ the encoder follows a particular path according to the input sequence $\{a_k\}$. Let us consider an input sequence of length L, $[a_1a_2 \cdots a_L]$. If each bit a_k can take on "1" or "0" with no restriction, there are 2^L different sequences. These result in 2^L different paths on the trellis of Fig. 4. An optimum decoder will be the one that chooses, on the basis of the observation sequence $\{y_k\}$, the most likely single path out of the 2^L possible candidates.

In the present section we assume that the noise $\{z_k\}$ is independent from digit to digit. This means that the channel has no memory besides the one-digit memory introduced by the correlative level encoder (i.e., one digit memory due to the inherent differentiation of the reading head). A more general case in which the disturbance $\{z_k\}$ is correlated is considered in a later section.

Under the present assumption, for a given output sequence $[y_1y_2 \cdots y_L]$, the likelihood function of a path $[s_0; a_1a_2 \cdots a_L]$ is given by the product of the likelihood function of L transitions:

$$p(y_1y_2 \cdots y_L \mid s_0; a_1a_2 \cdots a_L) = \prod_{k=1}^{L} p(y_k \mid s_{k-1}; a_k), \tag{14}$$

where

$$s_{k-1} = a_{k-1} \qquad \text{(NRZ system)} \tag{15a}$$

$$s_{k-1} = a'_{k-1} = a_{k-1} \oplus s_{k-2} \qquad \text{(NRZI system).} \tag{15b}$$

On taking the logarithm we obtain

$$l(y_1y_2 \cdots y_L \mid s_0; a_1a_2 \cdots a_L) = \sum_{k=1}^{L} l(y_k \mid s_{k-1}; a_k) \tag{16}$$

where

$$l(\cdot \mid \cdot) = \ln p(\cdot \mid \cdot). \tag{17}$$

Equation (16) means that the log-likelihood function of a given path is representable as the sum of the log-likelihood function of all branches.

In the rest of the present and following sections we limit ourselves to the NRZ recording method. All these results can be interpreted for the NRZI system by simply referring to the trellis picture of Fig. 4(b). Consider in Fig. 4(a) two paths λ_1 and λ_2 defined by

$$\lambda_1 = [s_0 = 0; a_1 = 0, a_2 = 0, a_3a_4 \cdots a_L] \tag{18}$$

$$\lambda_2 = [s_0 = 0; a_1 = 1, a_2 = 0, a_3a_4 \cdots a_L], \tag{19}$$

in which $a_3a_4 \cdots a_L$ are arbitrary but are common to two paths λ_1 and λ_2. Clearly λ_1 and λ_2 diverge at $t = 1$, remerge at $t = 2$ and remain together beyond that. Then for any output sequence $\mathbf{y} = [y_1y_2 \cdots y_L]$ the difference of the log-likelihood functions of two paths λ_1 and λ_2 is simply given, due to the relation (16), by

$$l(\mathbf{y} \mid \lambda_1) - l(\mathbf{y} \mid \lambda_2) = l(y_1y_2 \mid 0; 00) - l(y_1y_2 \mid 0; 10). \tag{20}$$

Therefore, if $l(y_1y_2 \mid 0; 00) > l(y_1y_2 \mid 0; 10)$, then path λ_2 can never be the most likely path, hence we might as well discard path λ_2 at $t = 2$.

The MLD algorithm proceeds as follows: Starting from the known initial state s_0, the decoder considers two paths emanating from s_0 and computes log-likelihood functions $l(y_1 \mid s_0; 0)$ and $l(y_1 \mid s_0; 1)$. We define the metric of the nodes $s_1 = 0$ and $s_1 = 1$ by

$$m_1(0) = l(y_1 \mid s_0; 0) \tag{21a}$$

and

$$m_1(1) = l(y_1 \mid s_0; 1). \tag{21b}$$

Then at $t = 2$, the decoder compares the log-likelihood functions of two different paths leading to $s_2 = 0$, i.e., $m_1(0) + l(y_2 \mid 0; 0)$ and $m_1(1) + l(y_2 \mid 1; 0)$. Let the path with a larger likelihood function be called the "survivor" [8], since this path possesses the possibility of being a portion of the maximum likelihood path and hence should be preserved. In like manner the decoder compares two paths ending at $s_2 = 1$. To each of the nodes, $s_2 = 0$ and $s_2 = 1$, the decoder assigns the metric which is the log-likelihood function of the survivor:

$$m_2(0) = \max\,\{m_1(0) + l(y_2 \mid 0; 0),\, m_1(1) + l(y_2 \mid 1; 0)\} \tag{22}$$

and

$$m_2(1) = \max\,\{m_1(0) + l(y_2 \mid 0; 1),\, m_1(1) + l(y_2 \mid 1; 1)\}. \tag{23}$$

At time $t = k$, in general, the decoder compares the log-likelihood functions of two different paths leading to the node $s_k = i$, i.e., $m_{k-1}(0) + l(y_k \mid 0; i)$ and $m_{k-1}(1) + l(y_k \mid 1; i)$ and discards the less likely path, where $i = 0, 1$. The metric of the node $s_k = i$, $i = 0, 1$ is the entire log-likelihood function of the survived path and is given by

$$m_k(i) = \max\,\{m_{k-1}(0) + l(y_k \mid 0; i),\; m_{k-1}(1) + l(y_k \mid 1; i)\}, \qquad i = 0, 1. \tag{24}$$

As is already clear from the above argument, for a given time $t = k$ two different paths have survived; one ending at node $s_k = 0$ and the other ending at node $s_k = 1$. Thus it seems as though the decoder had always to store the two most-likely sequences from $t = 1$ up to $t = k$. If that were the case, the decoder would need a huge memory capacity. Fortunately such difficulty is resolved by the following observation. Each time an event occurs such that

$$m_k(i) = m_{k-1}(j) + l(y_k \mid s_{k-1} = j; a_k = i) \tag{25}$$

holds for both $i = 0$ and 1, then the most likely path up to $t = k - 1$ is uniquely determined independently of the succeeding digits. That is, the survivor ending at node

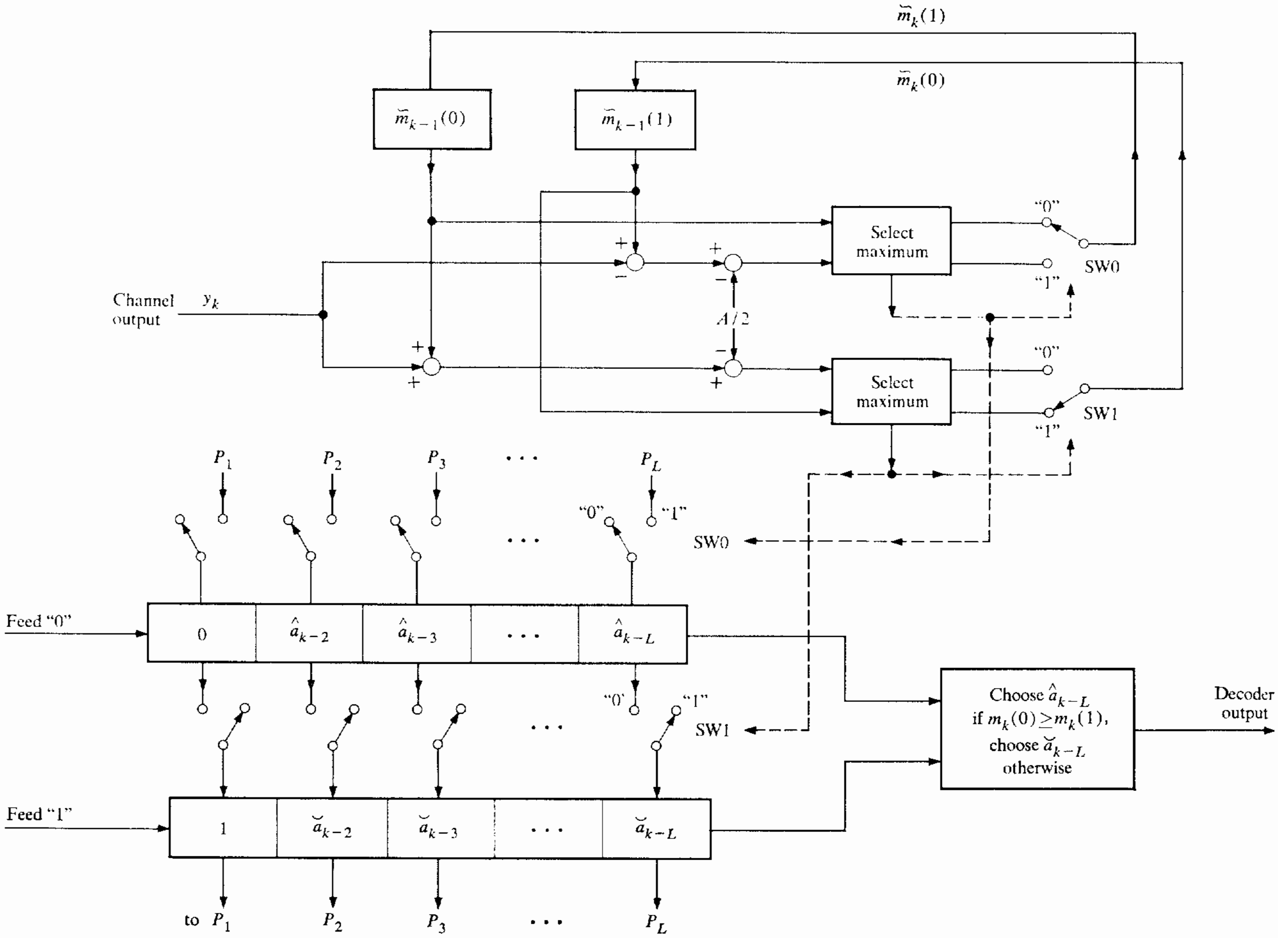


Figure 5 An implementation example of the maximum likelihood decoder for the NRZ system.

$s_{k-1} = j$ must be a portion (for $0 \leq t \leq k-1$) of the maximum likelihood solution. Therefore, the decoder can send out this portion of the sequence as the final output and this part need no longer be stored in the decoder.

Now we are in a position to derive a practical scheme for implementing the maximum likelihood decoder. We denote the actual signal levels in the channel by $+A$, 0 and $-A$ instead of -1, 0 and $+1$. Let us assume here that $\{z_k\}$ of Eq. (12) is a Gaussian random variable with zero mean and variance σ^2. The log-likelihood functions of Eqs. (24) are now given by

$$l(y_k \mid 0; 0) = l(y_k \mid 1; 1) = -y_k^2/2\sigma^2 - \ln\sqrt{2\pi\sigma}, \quad (26a)$$

$$l(y_k \mid 0; 1) = -(y_k - A)^2/2\sigma^2 - \ln\sqrt{2\pi\sigma}, \quad (26b)$$

$$l(y_k \mid 1; 0) = -(y_k + A)^2/2\sigma^2 - \ln\sqrt{2\pi\sigma}. \quad (26c)$$

Notice that the terms $-y_k^2/2\sigma^2 - \ln\sqrt{2\pi\sigma}$ are common to all the log-likelihood functions and hence can be deleted. Furthermore by dividing all terms by a constant $A/2\sigma^2$, we have a simplified version of the maximum likelihood decoding rule:

$$\tilde{m}_k(0) = \max\{\tilde{m}_{k-1}(0), \tilde{m}_{k-1}(1) - y_k - A/2\} \quad (27a)$$

and

$$\tilde{m}_k(1) = \max\{\tilde{m}_{k-1}(0) + y_k - A/2, \tilde{m}_{k-1}(1)\}, \quad (27b)$$

where $\tilde{m}_k(j)$ represents the modified metric of state j at time $t = k$, where $j = 0, 1$. Note that σ^2, the variance of the noise, does not appear in Eq. (27); i.e., the decoder structure is independent of the signal-to-noise ratio. Figure 5 shows the maximum likelihood decoder diagrammatically based on the rule of Eq. (27).

Let $[\cdots \hat{a}_{k-L} \cdots \hat{a}_{k-3}\hat{a}_{k-2}, 0]$ be the survivor ending at state 0 at time $t = k-1$. Similarly, let $[\cdots \breve{a}_{k-L} \cdots \breve{a}_{k-3}\breve{a}_{k-2}, 1]$ be the survivor path ending at state 1 at time $t = k-1$. These two sequences are stored in the shift registers, Storage 0 and 1, respectively. If the size L of the storages is sufficiently large, the digits $\hat{a}_{k-L}$ and $\breve{a}_{k-L}$ almost always agree: the agreement of $\hat{a}_{k-L}$ and $\breve{a}_{k-L}$ is assured if and only if an event defined by Eq. (25) has occurred at least once during the interval between $t = k - L$ and $t = k$. If it has not, we say that a buffer overflow has taken place and the decoder will send out $\hat{a}_{k-L}$ if $m_k(0) \geq m_k(1)$ and $\breve{a}_{k-L}$ otherwise. The problem of buffer overflows is considered in a later section.

For a given channel output y_k a new pair of surviving paths is determined according to Eq. (27) along with the new values of metrics $\tilde{m}_k(0)$ and $\tilde{m}_k(1)$. The switch SW 0 is to be connected to the position "0" if

$$\tilde{m}_k(0) = \max\{\tilde{m}_{k-1}(0),\ \tilde{m}_{k-1}(1) - y_k - A/2\}$$
$$= \tilde{m}_{k-1}(0) \qquad (28)$$

and to "1" otherwise. Similarly the switch SW 1 is to be connected to "1" if

$$\tilde{m}_k(1) = \max\{\tilde{m}_{k-1}(0) + y_k - A/2,\ \tilde{m}_{k-1}(1)\}$$
$$= \tilde{m}_{k-1}(1) \qquad (29)$$

and to "0" otherwise.

If SW 1 is set on the position "0", the information content of Storage 0 is written into Storage 1. Likewise if SW 0 is connected to "1", the sequence of Storage 1 is copied into Storage 0. Then the sequences in Storage 0 and Storage 1 are to be shifted to the right by one unit, sending out the rightmost digit $\hat{a}_{k-L}$ as the decoder output, and at the same time "0" and "1" are fed into the leftmost registers of Storages 0 and 1, respectively. Then the decoder waits for the arrival of the next channel output y_{k+1}.

3. Performance analysis and simulation

In the present section we present analytical results on the performance of the MLD algorithm and the confirmation of these results by computer simulations. Before we start the performance analysis, an important remark should be given. The MLD algorithm discussed in the previous section is an optimum decoding rule only when 1) all possible 2^N paths (where N is the total length of "message" sequences) are a priori equally likely and when 2) the criterion for optimality is minimization of message error probability. Clearly condition 1) is satisfied in most cases, since we usually assume that the information sequence $\{a_k\}$ takes on "1" and "0" equally likely and independently. The condition 2, however, is not always true. Instead, information bit error rate rather than the message error probability will be a more appropriate performance measure in many cases. Under such a criterion the MLD rule given above is not necessarily the optimum decoding rule. In other words, the most likely coded message sequence $[\hat{x}_1\hat{x}_2 \cdots \hat{x}_N]$ determined by the MLD does not necessarily correspond to the decoded information sequence $[\hat{a}_1\hat{a}_2 \cdots \hat{a}_N]$ that contains, on the average, the minimum number of erroneous bits. Of course, if $[\hat{x}_1\hat{x}_2 \cdots \hat{x}_N]$ is error-free, so is $[\hat{a}_1\hat{a}_2 \cdots \hat{a}_N]$.

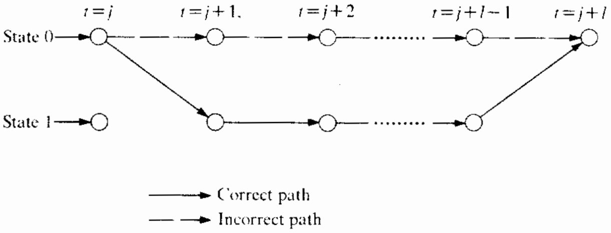


Figure 6 A correct path and its adversary path.

We see from the trellis picture of Fig. 4 that in the maximum likelihood decoding algorithm an error occurs when and only when the decoded path diverges from the correct path at some time $t = j$. They remerge at some time later, say at $t = j + l$, $l \geq 2$. Now we make an important observation which will simplify our analysis. If a correct path changes its state from $s_{k-1} = 0$ to $s_k = 1$, it generates $+A$ as the channel sequence x_k. Then under a high SNR (signal-to-noise ratio) condition any path which includes a transition from state $s_{k-1} = 1$ to $s_k = 0$ rarely remains as a survivor, since such a transition represents the opposite extreme level, $-A$, in the signal level. In other words, it is very unlikely under a high SNR that the correct path and the decoded path cross each other in the trellis picture. So we may neglect such rare cases and yet be able to compute the decoding error probability with satisfactory accuracy.

Suppose that at time $t = j$ the decoder is located at the correct state s_j and we assume without loss of generality that $s_j = 0$. In Fig. 6 the solid line represents the correct path and the broken line shows the incorrect path which remerges for the first time at $t = j + l$, $l \geq 2$. Note that the state of the correct path remains unchanged (so does the state of the incorrect path) between $t = j + 1$ and $t = j + l - 1$. Figure 6 shows the case in which $s_{j+1} = s_{j+2} = \cdots = s_{j+l-1} = 1$. If the state of the correct path ever changed at $t = k(j + 1 \leq k \leq j + l - 1)$, then remerging must have taken place at $t = k$ or before because of the remark just made. In other words, if the correct path changes the state at $t = k$, there exists no unmerged "adversary" beyond $t = k$. It is also important to notice that for a given correct path there exists only one adversary path that diverges at $t = j$ and remerges for the first time at $t = j + l$.

Consider the true path of Fig. 6. The adversary path diverges from a given correct path at $t = j$ and remerges at $t = j + l$. Take the likelihood ratio of that adversary with respect to the correct path:

$$\Lambda(\mathbf{y})$$

$$= \frac{p(y_{j+1} \cdots y_{j+l} | s_j = 0; a_{j+1} = \cdots = a_{j+l-1} = 0, a_{j+l} = 0)}{p(y_{j+1} \cdots y_{j+l} | s_j = 0; a_{j+1} = \cdots = a_{j+l-1} = 1, a_{j+l} = 0)}$$

$$= \frac{p(y_{j+1} | s_j = 0;\ a_{j+1} = 0)}{p(y_{j+1} | s_j = 0;\ a_{j+1} = 1)} \frac{p(y_{j+l} | s_{j+l-1} = 0;\ a_{j+l} = 0)}{p(y_{j+l} | s_{j+l-1} = 1,\ a_{j+l} = 0)}. \tag{30}$$

The last expression is obtained using the fact that $x_k = 0$ for $j + 2 \leq k \leq j + l - 1$ in both the correct and adversary paths; hence these branches do not contribute to the likelihood ratio function. We define random variable w_l which is proportional to the log-likelihood ratio function:

$$w_l = \frac{\sigma^2}{A} \ln \Lambda(\mathbf{y}). \tag{31}$$

On substituting Eqs. (17) and (26) into (31), we obtain

$$w_l = -y_{j+1} + y_{j+l} + A. \tag{32}$$

Since y_{j+1} and y_{j+l} are Gaussian random variables with means A and $-A$, respectively, and with variance σ^2, w_l is also a Gaussian random variable with the mean $-A$ and variance $2\sigma^2$. Note that Eq. (32) could have been obtained directly from Eq. (27). Although the likelihood function of (30) was defined for the particular choice of a correct path represented by Fig. 6, it can be shown that the random variable w_l defined by Eq. (31) holds the same distribution when we interchange the correct path and adversary, or when we let these two paths end at state $s_{j+l} = 1$ rather than at state $s_{j+l} = 0$.

As is clear from the decoding rule discussed in the previous section, a decoding error occurs at $t = j + l$ when the random variable w_l exceeds zero. The probability that w_l exceeds zero, with the conditions that the particular correct path is selected at the information source and that the correct path has survived at the decoder up to $t = j + l$, is given by

$$P_e(l) = \int_0^\infty \phi\left(\frac{w_l + A}{\sqrt{2}\,\sigma}\right) \frac{dw_l}{\sqrt{2}\,\sigma}$$

$$= 1 - \Phi(d) = \Phi(-d), \tag{33}$$

where $\phi(\cdot)$ is the unit normal density distribution function, $\Phi(\cdot)$ is the complementary error function defined by

$$\Phi(x) = \int_{-\infty}^{x} \phi(t)\, dt \tag{34}$$

and d^2 is SNR in the channel:

$$d^2 = \mathrm{E}[x_k^2]/\sigma^2 = A^2/2\sigma^2. \tag{35}$$

This error event causes decoding error in $(l - 1)$ successive information bits $[a_{j+1}a_{j+2} \cdots a_{j+l-1}]$ in the NRZ recording system. In the NRZI recording system the two digits a_{j+1} and a_{j+l} are erroneously decoded and $l - 2$ digits between them are decoded correctly which is due to the precoding operation. The expected number of decoding error bits due to all possible incorrect paths which diverge at $t = j$ is

$$P_{e\,\mathrm{MLD}} = \sum_{l=2}^{\infty} (l - 1) 2^{-(l-2)} P_e(l) = 4\Phi(-d) \tag{36}$$

for the NRZ system and the corresponding expression for the NRZI system is

$$P_{e\,\mathrm{MLD}} = \sum_{l=2}^{\infty} (2) 2^{-(l-2)} P_e(l) = 4\Phi(-d), \tag{37}$$

which is the same as (36). In Eqs. (36) and (37) the weighting factor $2^{-(l-2)}$ which is included gives the probability that the correct path maintains the same state at least during the interval from $t = j + 1$ through $t = j + l - 1$.

It should be remarked here that the expression given by Eq. (35) is not exactly the decoding error probability. To be precise we should have defined $P_e(l)$ as the probability of the event that the correct path is beaten by the adversary path diverging at $t = j$ *for the first time at* $t = j + l$. It can be shown after some manipulation that the precise expression for $P_e(l)$ is given by

$$P_e(l) = \int_{-\infty}^{\infty} \Phi^{l-2}(u) \Phi(-u) \phi(u - \sqrt{2}\, d)\, du. \tag{38}$$

It is not hard to show that Eq. (38) is reduced to Eq. (33) for $d \gg 1$. For the numerical evaluation we use the approximate solution (36) or (37), since the union bound provides a satisfactory approximation under a high SNR condition. Recall that we already adopted some approximating assumption concerning adversary paths at the very beginning of the present section.

An impressive, but not totally unexpected, result is the fact that $P_e(l)$ of Eq. (33) is exactly equal to the probability of error in a binary antipodal signalling system with the same signal-to-noise ratio, i.e., d^2. Equation (36) shows that the decoding error rate for the NRZ system is asymptotically (i.e., as SNR goes to infinity) four times the error rate in an optimum binary system with the same SNR. Curve A of Fig. 7 is a plot of Eq. (36) or (37), where the horizontal ordinate is SNR in dB, i.e., 20 $\log_{10} d$.

The performance represented by Curve A is now compared with that of the conventional bit-by-bit detection method for the NRZI system in which precoding eliminates the propagation of errors. If the thresholds of the detector are set at $-A/2$ and $A/2$, the bit error rate is given by

$$P_{e\,\mathrm{BIT}} = \tfrac{3}{2}\Phi(-d/2). \tag{39}$$

If the variance σ^2 of the noise is known in advance, $A/2$ and $-A/2$ are not the optimum thresholds, since

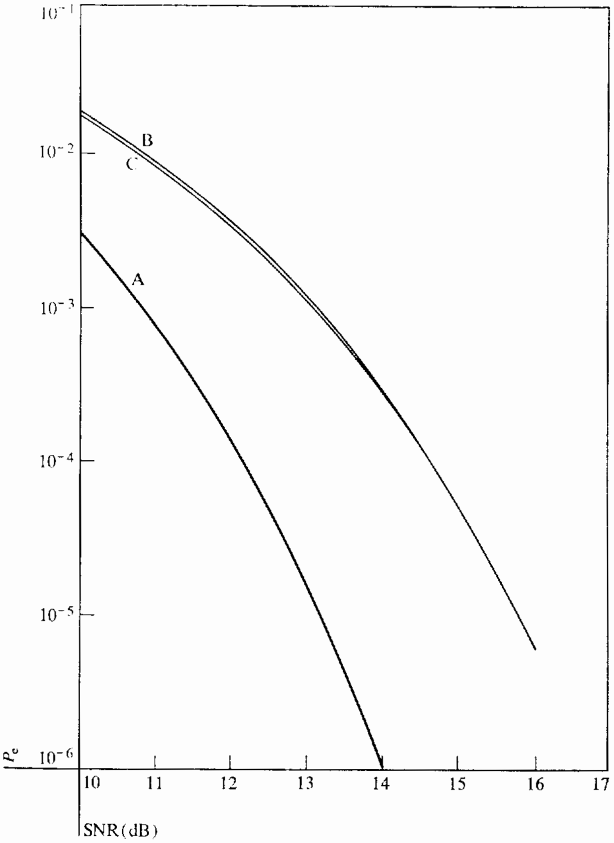


Figure 7 Probability of error vs SNR: (A) the maximum likelihood decoding. (B) the bit-by-bit detection with thresholds at $-A/2$ and $A/2$. (C) the bit-by-bit detection with the optimum thresholds.

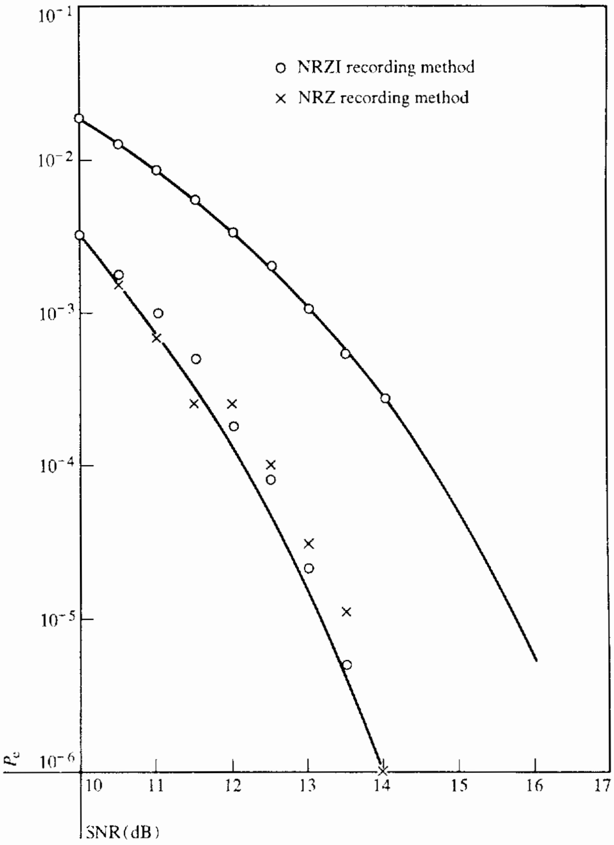


Figure 8 Simulation results of the maximum likelihood decoding method and the bit-by-bit detection method.

the channel symbol x_k takes on A, 0, and $-A$ with probabilities, $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{1}{4}$, respectively. The optimum thresholds are t_{opt} and $-t_{\text{opt}}$, where t_{opt} is the solution of the equation

$$\tfrac{1}{2}\phi(t_{\text{opt}}/\sigma) = \tfrac{1}{4}\phi[(t_{\text{opt}} - A)/\sigma], \tag{40}$$

which yields

$$t_{\text{opt}} = \frac{A}{2} + \frac{\sigma^2}{A} \ln 2. \tag{41}$$

Thus the bit error rate with these optimum threshold values is given by

$$P'_{e\,\text{BIT}} = \Phi\left(-\frac{d}{\sqrt{2}} - \frac{\sqrt{2}}{d} \ln 2\right) + \frac{1}{2}\,\Phi\left(-\frac{d}{\sqrt{2}} + \frac{\sqrt{2}}{d} \ln 2\right). \tag{42}$$

Curves B and C in Fig. 7 are plots of Eqs. (39) and (41), respectively. Their difference is fairly small. We can see, however, a substantial difference between Curves A and C. For example, at SNR = 13 dB $P'_{e\,\text{BIT}} = 1.1 \times 10^{-3}$ whereas $P_{e\,\text{MLD}} = 1.8 \times 10^{-5}$, i.e., improvement by a factor of 70. The performance improvement is even higher for a higher SNR: the decrease in the error probability by a factor of several hundred is possible beyond SNR = 14 dB. In terms of SNR, the maximum likelihood decoding method gains as much as 2.5 dB in the range of raw error rate 10^{-3} to 10^{-4}.

Now we shall report the computer simulation results and confirm the analytical results obtained above. The discrete channel models of Figs. 2(a) and (b) are assumed. The data sequence $\{a_k\}$ was generated through the random number generator program, and the noise sequence $\{z_k\}$ was generated by transforming the random variable with a uniform distribution through a polynomial approximation formula of the mapping $\Phi^{-1}(\cdot)$ [12]. It will be worth mentioning here that most existing subroutine programs under the name "Gaussian Random Generator" are not appropriate to this type of simulation, in which a high accuracy is required at the tail of the probability density distribution.

The simulation results are plotted in Fig. 8, where decoding error rates for the NRZ and NRZI recording methods are marked by × and ○, respectively. The size N of data is 10^5 for SNR = 10 to 11.5 dB, and 10^6 for

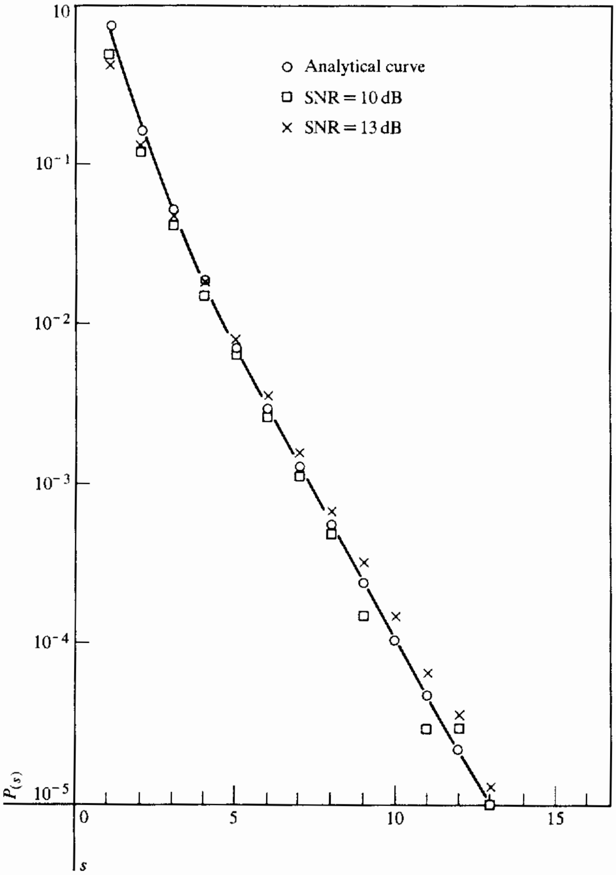


Figure 9 Plot of Eq. (43) and simulation results for SNR = 10 dB and 13 dB.

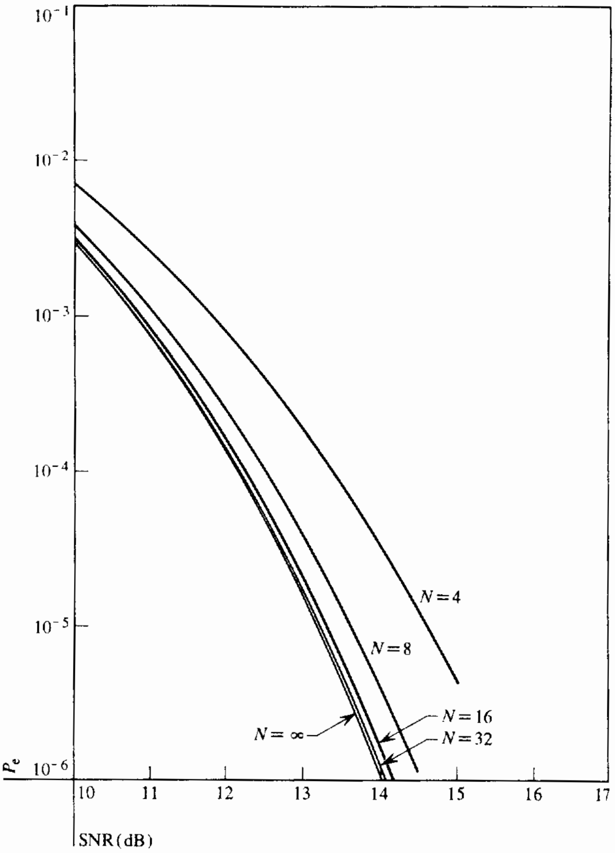


Figure 10 Probability of decoding error vs SNR when the maximum likelihood decoder is preceded by a quantizer with a uniform spacing A/N, where $N = 4$, 8, 16 and 32.

SNR $\geq$ 12 dB. Although the simulation data size N should be even higher for a more reliable measurement we may safely conclude that the analytical curve and the experimental result agree satisfactorily. The NRZI method gives a slightly better performance than the NRZ method. This difference can be explained if we go back to Eq. (38) and realize that the term $\Phi^{l-2}(u)(< 1)$ in Eq. (38) becomes non-negligible for a large l and hence the error rate of the NRZI system is less than that of the NRZ system.

In the simulation performed above the buffer length $L = 25$ was chosen to avoid a possible overflow. As was mentioned earlier a buffer overflow occurs when and only when events defined by Eq. (25) are separated by more than L time units. Let k and k' ($k' > k$) be two consecutive times at which Eq. (25) holds. Then the distribution of the separation $s = k' - k$ is given by

$$P(s) = 2^{-s}\left[\frac{2}{s} - \frac{1}{s+1}\right] \approx \frac{1}{s}\, 2^{-s}. \tag{43}$$

The derivation of Eq. (43) will be found elsewhere [13]. Figure 9 shows the simulation results for SNR = 10 dB and SNR = 13 dB along with the analytical result of (43). A satisfactory agreement is observed here also, and for $L \geq 13$ the probability of buffer overflow is less than 10^{-5} per digit.

Thus far we have assumed that the decoder input y_k is a sampled but unquantized value (or, equivalently, quantized into infinite number of levels). In an actual implementation, presumably in a digital circuit, the channel output y_k must be quantized into a finite number of levels before entering the decoder. Let us assume a typical analog-to-digital converter, i.e., a uniformly spaced quantizer with spacing A/N, where A is the signal level spacing as was defined in the previous section. Figure 10 shows the variation of decoding error rate for different quantization spacings and we may conclude that $N = 16$ achieves almost the same performance as the infinite quantization levels (less than 0.1 dB loss in SNR). Thus if we quantize uniformly between $-2A$ and $2A$, a 6-bit analog-to-digital converter is sufficient to perform the MLD satisfactorily, and the computation of metrics $\tilde{m}_k(0)$ and $\tilde{m}_k(1)$ is performed simply by addition and subtraction of integers.

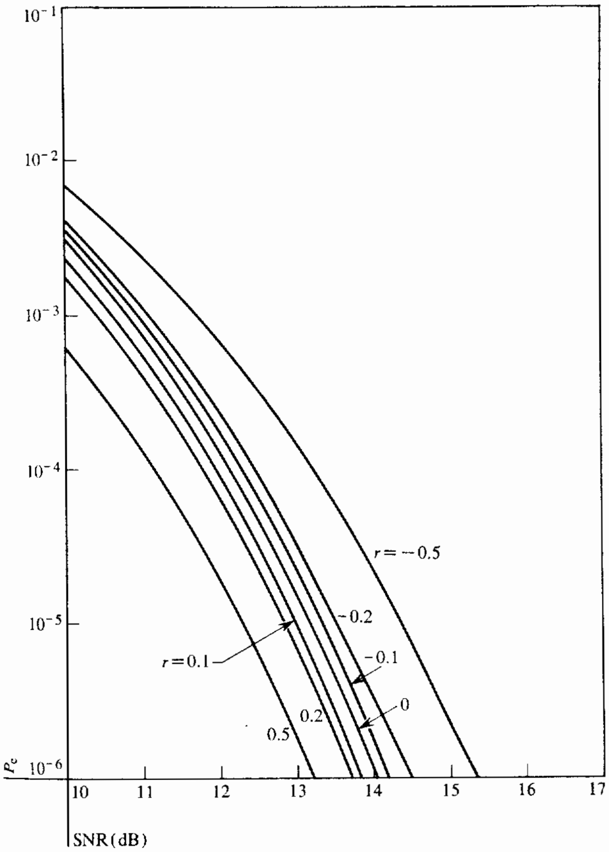


Figure 11 Effects of correlated noise on the performance of the maximum likelihood decoder. r = the correlation coefficient of the noise.

4. Remarks and conclusions

Throughout our discussion it has been assumed that the noise $\{z_k\}$ is an independent Gaussian random variable and that the channel is memoryless. We include here a few remarks concerning the behavior of the maximum likelihood decoder under the circumstances where this idealized assumption does not hold. The effect of correlatedness of the noise sequence $\{z_k\}$ can be observed from Eq. (32); that is, if the noises z_{j+1} and z_{j+l} are highly correlated and have a positive correlation, then the noises effectively cancel each other and the error event will occur less frequently. On the other hand if two noises z_{j-1} and z_{j+l} are correlated negatively, the result will be the reverse. Figure 11 is a plot of the curve P_e vs SNR when the noise is correlated according to the relation

$$E[z_k z_{k+l}] = r^l \sigma^2, \qquad (44)$$

$$\text{where} \qquad |r| \le 1. \qquad (45)$$

As expected from the above discussion a better performance is obtained for $r > 0$ than for the uncorrelated case, i.e., $r = 0$.

If the probability density function of the noise $p_z(\cdot)$ is not Gaussian, the simple structure of Fig. 5 is no longer the maximum likelihood decoder. However Eq. (32), which is the key to this superb decoding algorithm, still holds and the performance can be computed exactly in the same way as for a Gaussian case simply by replacing Eq. (33) with

$$P_e(l) = \int_0^\infty p_z^{(2)}(w_l + A)\,dw_l, \qquad (46)$$

where $p_z^{(2)}(\cdot)$ is the autocorrelation of the function $p_z(\cdot)$:

$$p_z^{(2)}(w) = \int_{-\infty}^\infty p_z(x)p_z(x + w)\,dx. \qquad (47)$$

The intersymbol interference, which is the major obstacle to high-density recording reduces in effect the margin of signal level separation against the random noise. It is then clear that the MLD performs better than the bit detection method in the presence of intersymbol interference also. The probability of decoding error in the presence of the intersymbol interference will be accordingly increased. An upper bound for the decoding error rate is given by

$$P_{e\,\mathrm{MLD}} \le 4\Phi[-d(1 - D)], \qquad (48)$$

where D ($0 \le D < 1$) is the distortion coefficient due to intersymbol interference given by

$$D = \frac{1}{g_0}\sum_{k\ne 0}|g_k|, \qquad (49)$$

where $g_k = g(kT)$ is the sampled response function of the channel [see Eq. (10)].

We have shown in this paper the following results.

1) An analogy between convolutional coding and correlative level coding (or the partial-response signalling) is clarified, and a linear finite machine description of the magnetic recording system has been derived.
2) The maximum likelihood decoding algorithm has been applied to the NRZ and NRZI recording system. The decoding rule and its practical implementation (Fig. 6) are discussed in detail.
3) Asymptotic expressions for the decoding error probability [Eqs. (36) and (37)] have been obtained. The superb performance of the maximum likelihood decoder has been shown and confirmed by computer simulations. The maximum likelihood decoding method gains approximately 2.5 dB in SNR compared with the bit detection method and the error rate is reduced by a factor of 50 to 300 in the raw error rates in the 10^{-3} to 10^{-4} range. The improvement factor further increases for a higher SNR.
4) Important problems associated with the maximum likelihood decoding algorithm are discussed. These include

the problem of buffer overflow, the number of quantization levels required, the effects of correlatedness and non-Gaussianness of the noise on the decoder performance, and the degradation in the presence of intersymbol interference.

Although our discussion has been limited to the NRZ and NRZI recording methods, the MLD algorithm is immediately applicable to the Interleaved NRZI, a high density recording scheme recently proposed [1]. Extensions to other types of recording systems such as MFM (modified frequency modulation) and the double-frequency modulation are rather straightforward. Under current investigation is an extension of the MLD algorithm to concatenated or hybrid schemes, i.e., correlative level coding plus some other coding such as the run-length limited coding, burst error correcting codes, etc.

Acknowledgment

It is a pleasure to record here a debt of gratitude to A. J. Viterbi and J. K. Omura for the opportunity of discussing their work on the maximum likelihood decoding algorithm when the author was visiting the Department of System Science, University of California at Los Angeles. The author also wishes to thank D. T. Tang, IBM Research, for the various constructive comments he provided.

References

1. H. Kobayashi and D. T. Tang, "Application of Partial-response Channel Coding to Magnetic Recording Systems," *IBM J. Res. Develop.* **14,** 368 (1970).
2. A. Lender, "Correlative Level Coding for Binary Data Transmission," *IEEE Spectrum* **3,** No. 2, 104 (1966).
3. E. R. Kretzmer, "Generalization of a Technique for Binary Data Transmission," *IEEE Trans. Comm. Tech.* **COM-14,** 67 (1966).
4. P. J. van Gerwen, "On the Generation and Application of Pseudo-ternary Codes in Pulse Transmission," *Phillips Res. Repts.* **20,** 469 (1965).
5. R. W. Lucky, J. Salz and E. J. Weldon, Jr., *Principles of Data Communication,* McGraw-Hill Book Co., New York, 1968.
6. H. Kobayashi and D. T. Tang, "On Decoding and Error Control of Correlative Level Coding," presented at 1970 International Symposium on Information Theory, Noordwijk, the Netherlands, June 15–19. Abstracts, pp. 43–44.
7. H. Kobayashi and D. T. Tang, "On Decoding and Error Control for Correlative Level Coding Systems," submitted to *IEEE Trans. Comm. Tech.;* also to appear as IBM Research Report, IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y.
8. A. J. Viterbi, "Error Bounds for Convolutional Codes and Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Info. Theory* **IT-13,** 260 (1967).
9. G. D. Forney, Jr., "Final Report on a Coding System Design for Advanced Solar Mission," NASA Contract NAS2-3637, Codex Corp., Watertown, Mass., December 1967.
10. J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE Trans. Info. Theory* **IT-15,** 177 (1969).
11. A. J. Viterbi, "Convolutional Codes, Linear Finite-State Machines and Maximum Likelihood Decoding," to be published.
12. M. Abramowitz and I. R. Stegun (Eds.), *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables,* U. S. Department of Commerce, National Bureau of Standards, Applied Mathematics Series 55; 1964, p. 933.
13. H. Kobayashi, "Correlative Level Coding and the Maximum Likelihood Decoding," IBM Research Report RC 2999, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, August, 1970.

Received March 30, 1970

Part of the work reported in this paper was done during the author's stay at the Department of System Science, University of California at Los Angeles from September 1969 to March 1970. The author is now located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.