Symbol synchronization in convolutionally coded systems
Baumert, L.; McEliece, R.J.; van Tilborg, H.C.A.

Published in:
IEEE Transactions on Information Theory

DOI:
10.1109/TIT.1979.1056044

Published: 01/01/1979

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 31. Aug. 2018
A t-error-correcting code is perfect if the covering radius is t. The code is quasi-perfect if the covering radius is t + 1.

Let β be an element of order n = 2^m - 1. The largest cyclic code whose generator polynomial g(x) ∈ GF(2)[x] has the zeros β, β^2, ..., β^{t-1} but not β^t is defined to be a primitive BCH code of designed distance d and is here denoted by B(d). Note that d must be odd if B(d) exists.

The code B(2) is the Hamming code, which is a one-error-correcting perfect code. Gorenstein, Peterson, and Zierler [1] proved that B(5) is a two-error-correcting quasi-perfect code. They also proved that B(7) is a three-error-correcting code which has covering radius at least five, and thus B(7) is not quasi-perfect. Later Van der Horst and Berger [2], Assmus and Mattson [3], and Helleseth [4] proved that B(7) has covering radius five.

In this correspondence we will prove a conjecture due to Gorenstein, Peterson, and Zierler [1], which says that B(d) is never quasi-perfect when d > 7.

Leont'ev [5] proved that B(d) is not quasi-perfect when 2 < (d - 1)/2 < √n/log n and m > 7.

We will need the following lemmas.

**Lemma 1:** If d = 2^r - 1, r < m, then B(d) exists and has actual minimum distance d.

**Lemma 2:** If d = 2^r - 2^s - 1, where 0 < (r - 1)/2 < s < r < m, then B(d) exists and has actual minimum distance d.

**Theorem 1:** No primitive binary t-error-correcting BCH code is quasi-perfect when t > 2.

Before proving Theorem 1 we prove the following stronger result.

**Theorem 2:** Let ρ_d and τ_d denote the covering radius and actual error correcting ability of B(d), respectively, and let 3 < r < m - 1.

i) If 2^r - 2^{r+1} - 1 < d < 2^r - 2^s - 1 where s is one of the numbers [s], [s - 1], ..., r - 2, then

\[ \rho_d - \tau_d > \frac{2^{r-1} - 3}{2^{r-1} - 1} (\tau_d + 1). \]

ii) If 2^r - 2^{r+1} - 1 < d < 2^r - 2^s - 1, then

\[ \rho_d - \tau_d > \frac{2^{(r+1)/2} - 1}{2^{(r+1)/2} - 1} (\tau_d + 1). \]

**Proof:**

i) Let 2^r - 2^{r+1} - 1 < d < 2^s - 2^r - 1 for some s = [s], [s + 1], ..., r - 2, where 3 < r < m - 1. By Lemma 2, \( B(2^r - 2^{r+1} - 1) \) and \( B(2^r - 2^s - 1) \) exist, and we have

\[ B(2^r - 2^s - 1) \subset B(d) \subset B(2^r - 2^{r+1} - 1). \]

Since \( B(d) \subset B(2^r - 2^{r+1} - 1) \), we can choose \( \alpha \in B(2^r - 2^{r+1} - 1) - B(d) \). Here \( \alpha \) has distance at least \( 2^r - 2^{r+1} - 1 \) from every element in B(d). From the definition of the covering radius it follows that

\[ \rho_d > 2^r - 2^{r+1} - 1. \]

Since \( B(2^r - 2^{r+1} - 1) \subset B(d) \), we get by Lemma 2

\[ \tau_d < 2^{r-1} - 2^{r-1} - 1. \]

Combining (1) and (2) we have

\[ \rho_d - \tau_d > 2^{2r-1} - 1. \]

which combined with (2) gives

\[ \rho_d - \tau_d > (\tau_d + 1)(2^{2r-1} - 1)/(2^{r-1} - 1). \]

This proves i).

### References


### Symbol Synchronization in Convolutionally Coded Systems

LEONARD D. BAUMERT, ROBERT J. MCELIECE, MEMBER, IEEE, AND HENK C. A. VAN TILBORG

**Abstract—** Alternate symbol inversion is sometimes applied to the output of convolutional encoders to guarantee sufficient richness of symbol transition for the receiver symbol synchronizer. A bound is given for the length of the transition-free symbol stream in such systems, and those convolutional codes are characterized in which arbitrarily long transition free runs occur.

#### I. INTRODUCTION

Many digital communication systems derive symbol synchronization from the transitions in the received symbol stream. In such systems usually long sequences of all zeros or all ones can cause temporary loss of synchronization and thus data loss. To avoid this problem, alternate symbols of the data stream are inverted; presumably a long alternating string is less likely than a long constant string.

Suppose the symbol stream is the alternately inverted output of a convolutional encoder. How long a constant string occurs?

Manuscript received May 8, 1978, revised August 28, 1978. This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract NAS7-100, sponsored by the National Aeronautics and Space Administration.

L. D. Baumert is with the Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91103.

R. J. McEliece is with the Mathematics Department, University of Illinois, Urbana, IL 61801.

H. C. A. van Tilborg is with the Mathematics Department, Technical University Eindhoven, Eindhoven, The Netherlands.
then? That is, how long a run of alternating symbols \( \cdots 010101 \cdots \) occurs in some codeword of a convolutional code? As we shall see, arbitrarily long alternating runs do occur in some codes; we characterize these codes in Section II. In Section III, for codes which do not have arbitrarily long alternating runs, we give upper bounds for the length of the longest run. In Section IV we consider examples which illustrate the use of these results and indicate how good the various upper bounds can be expected to be.

The reader is assumed to be familiar with the theory of convolutional codes and encoders as it appears, say, in Forney [1]. Thus terms like “overall constraint length,” “minimal encoder,” “dual code and dual encoder,” etc., are assumed known and used without definition. However, we remind the reader that the convolutional encoders of concern operate on binary sequences of the form \( x = (x_0, x_1, \ldots) \) which, theoretically at least, extend to infinity in both directions. The index refers to discrete time intervals. In practice each sequence “starts” at some finite time; i.e., there is an index \( s \) such that \( t < s \) implies \( x_t = 0 \). The codewords produced by the encoders are of the same type. Using the delay operator \( D \), it is sometimes convenient to write \( x = x_0D + x_1D^2 + \cdots \). We also use certain algebraic properties of these formal power series, e.g., \( D^k + D^{k+1} + \cdots = D^k(1 + D) \).

II. CONVOLUTIONAL CODES WITH AN INFINITE RUN OF ALTERNATING SYMBOLS

**Theorem 1:** Let \( C \) be an \((n, k)\) convolutional code over \( GF(2) \) with generator matrix \( G \). Then \( C \) contains a codeword with an infinite run of alternating symbols if and only if there exists a linear combination \( u = [u_1, \ldots, u_n] \) of the rows \( g_i \) of \( G \) such that

\[
[u_1, \ldots, u_n] \equiv [0, 1, \cdots, 0, 1] \text{ or } [1, 0, \cdots, 1, 0] \text{ modulo } 1 + D, \ n \text{ even }
\]

\[
[u_1, \ldots, u_n] \equiv [1, 1, \cdots, 1, 0] \text{ or } [D, 1, \cdots, 1, D] \text{ modulo } 1 + D^2, \ n \text{ odd }
\]

**Proof:** (Sufficiency): When \( n \) is even, consider the codeword produced by the inputs \( u_1/(1 + D) \) applied to the rows \( g_i \), where \( v = \Sigma a_j g_j \). Note that this same codeword is produced by applying \( 1 + D \) to each row of the equivalent encoder whose rows are \( a_j \). Thus after an initial transient the output will be \( v_1(1), \ldots, v_n(1) \) and since \( v(D) \equiv 0 \) modulo \( 1 + D \) the result follows. For \( n \) odd note that \( v(D) \equiv D^2 \) modulo \( 1 + D^2 \) means that the sum of its even coefficients is 0 and the sum of its odd coefficients is 1. whereas the situation is reversed for \( v(D) \equiv 1 \) modulo \( 1 + D^2 \). Thus after an initial transient the input sequences \( a_1/(1 + D^2) \) will produce an infinite run of alternating symbols.

(Necessity): When \( n \) is even an infinite run of alternating symbols results from the juxtaposition of \( n \)-tuples of the form \( 10 \cdots 10, 01 \cdots 01 \). For definiteness, assume the former occurs. Then, if a codeword of \( C \) contains such an infinite run, there exists a codeword \( u \) such that

\[
u = h + \frac{D^2}{1 + D^2} [1, 0, \cdots, 1, 0]
\]

Here \( h \) is an \( n \)-tuple of polynomials of degree \( < s \) which describes the initial segment of \( u \). Let \( v(D) = (1 + D^2)u(D) \). Obviously, \( v(D) \) is polynomial and \( v(D) \equiv [0, 1, \cdots, 0, 1] \) modulo \( 1 + D \).

Similarly, for \( n \) odd, \( C \) contains

\[
w = h' + \frac{D^2}{1 + D^2} [1, D, \cdots, D, 1, 1]
\]

Define \( v(D) \) as \( (1 + D^2)w(D) \). It follows as above that

\[
v(D) \equiv D^2 [1, D, \cdots, D, 1] \text{ modulo } 1 + D^2
\]

and the proof is complete.

If a basic encoder \( G \) is known for \( C \) then only \( 2^k \) (respectively, \( 4^k \)) linear combinations \( v = \Sigma a_j g_j \) need be tried, for then the \( a_i \) can be restricted to 0,1 (respectively, 0,1,\( D, 1+D \)) when \( n \) is even (respectively, \( n \) is odd). Even more efficiently, a row reduction could be used to determine whether or not the required vector was in the row space of \( G \) modulo \( 1 + D \) (or \( 1 + D^2 \)).

The case \( k = 1 \) is particularly important. Here, basic just means that the \( n \) polynomials making up the single generator \( g_1 \) have no common polynomial divisor and the test amounts to reducing \( g_1 \) modulo \( 1 + D \) or \( 1 + D^2 \).

It is also possible to test for the presence of an infinite alternating run in terms of the dual code (see Corollary to Theorem 2 below)

**Theorem 2:** Suppose an \((n, n - 1)\) convolutional code \( C \) over \( GF(2) \) is given and \( f = [f_1, \ldots, f_n] \) generates the dual code, where \( \gcd(f_1, \ldots, f_n) = 1 \). Then there is an infinite run of alternating symbols in some codeword of \( C \) if and only if

\[
(n \text{ even}) \quad 1 + D \beta \equiv 0 \text{ modulo } 1 + D^2 \text{ for } \alpha = 0 \text{ or } \alpha = 1
\]

\[
(n \text{ odd}) \quad 1 + D^2 \beta \equiv 0 \text{ modulo } 1 + D^2
\]

**Proof:** Since \( f_1, f_2, \ldots, f_n \) all codewords of the dual code are linear combinations of shifts of

\[
\cdots 0f_{10}f_{10} \cdots f_{11}f_{11}f_{11}f_{41}f_{41}f_{41}
\]

where \( d = \max(\deg(f_i)) \). Thus it is sufficient to check the inner products of this codeword of \( C \) with an infinite alternating run.

n even

\[
\cdots 01 0 1 0 1 0 \cdots
\]

\[
\alpha = 1
\]

\[
f_{10}f_{10}f_{10}f_{10}f_{11}f_{11}f_{11}f_{11}f_{41}f_{41}f_{41}
\]

\[
\alpha = 0
\]

n odd

\[
\cdots 01 0 1 0 1 0 \cdots
\]

\[
\text{(coefficient of } D)
\]

\[
f_{10}f_{10}f_{10}f_{10}f_{11}f_{11}f_{11}f_{11}f_{41}f_{41}f_{41}
\]

\[
\text{(constant)}
\]

In both cases the necessity of the above conditions is immediate. (For \( n \) odd the coefficients referred to are \( a, b \) from \( \Sigma_{j+1} + D \beta_1 \beta_2 \equiv aD + b \text{ modulo } 1 + D^2 \)).

On the other hand, the above conditions obviously guarantee the existence of a codeword \( \cdots \cdot 0101 \cdots \cdot 1010 \cdot \cdot \cdot \cdot \· \) extending infinitely in both directions. However, only codewords “starting” at some finite time are of concern, and it remains to be shown that such a codeword is in the code. But this is trivial; it amounts to using the same input sequences truncated to start at some time \( t_0 \), thus by time \( t_0 + \delta \), where \( \delta \) is the overall constraint length, the encoders shift registers will be set exactly as they were when generating the doubly infinite sequence. Thus from \( t_0 + \delta \) on the output will be an infinite alternating run.

Suppose an \((n, k)\) convolutional code \( C \) over \( GF(2) \) with generator matrix \( F \) for its dual code is given. Suppose \( F \) is a basic encoder, i.e., the \( \gcd \) of its \((n-k)\) subdeterminants is 1, then, if \( [f_1, \ldots, f_n] \) is any row of \( F \) it follows that \( [f_1, \ldots, f_n] = 1 \).

Let \( C_i \) \((i = 1, \ldots, n-k)\) be the \((n, n-1)\) convolutional code dual to the \( i \)-th row of \( F \). Clearly, \( C = \bigcap_{i=1}^{n-k} C_i \)

and the maximum run of alternating symbols in any codeword of \( C \) has length \( L = L(C) \leq \min_i L(C_i) \).

**Corollary:** When \( n \) is odd, an \((n, k)\) convolutional code \( C \) over \( GF(2) \) contains a codeword with an infinite run of alternating
symbols if and only if every row of a basic generator matrix \( F \) for \( C \) satisfies the congruences of Theorem 2. When \( n \) is even it is further necessary that this be true for the same value of \( \alpha (0 \) or 1).

**Note:** Suppose \( n \) is even and \( L(C_i) = L(C_j) = \infty \) with \( \alpha \neq 1 \) for \( C_i \) and \( \alpha \neq 0 \) for \( C_j \). Add row \( j \) to row \( i \) in \( F \); this gives an equivalent basic encoder which has \( L(C_i) < \infty \).

### III. Bounds for Finite Runs of Alternating Symbols

If no codeword contains an infinite run of alternating symbols the question arises as to the maximum length \( L \) of such a finite run. It is easy to give a bound for \( L \) in terms of the generators for the dual code. From this bound it is possible to derive another bound (in general, weaker) which has the advantage that it can be applied directly without knowledge of the dual (see the Corollary to Theorem 3, below). In Section IV these bounds are applied to some specific examples.

Suppose \([f_1, \ldots, f_s]\) is a generator matrix for an \((n, 1)\) convolutional code \( C \) over \( GF(2) \) with \( d = max (deg f_i) \). Then

\[
\begin{align*}
&f_0 f_1 f_2 \cdots f_{s-2} f_{s-1} f_s f_{s+1} f_{s+2} \cdots f_{n-1} f_n \\
is its associated bit pattern. Let \( s \) be the number of symbols occurring between the first and last nonzero symbols \( f_j \) inclusively. If \((f_1, \ldots, f_s) = 1, s \) is the minimum length of any nonzero codeword of \( C \) and
\end{align*}
\]

\[
n(d-1) + 2 < s < n(d+1).
\]

**Theorem 3:** Let \( C \) be an \((n, n)\) convolutional code over \( GF(2) \) with generator matrix for its dual code given by \([f_1, \ldots, f_s]\), where \((f_1, \ldots, f_s) = 1\). Suppose no codeword of \( C \) contains an infinite run of alternating symbols. Then the maximum run of alternating symbols in any codeword of \( C \) has length \( L = s + n - 2 \), when \( n \) is even or when \( n \) is odd and

\[
h(D) = \Sigma f_i D^i \mod (1 + D^2).
\]

Combining this with the limits given above for \( s \) yields

\[
n(d+1) < L < n(d+3) - 2, \quad n \text{ odd},
\]

\[
h(D) \equiv 1 + D \mod (1 + D^2).
\]

**Proof:** Suppose \( n \) is even. Then, from Theorem 2, \( \Sigma f_i D^i \equiv f_{s+1} = 1 \mod (1 + D) \). If there were an alternating run of length \( > s + n - 1 \) it would have \( s \) consecutive symbols which would have inner product zero with the bit pattern of \( f \). This contradicts \( \Sigma f_i D^i \equiv 1 \), so \( L < s + n - 2 \). On the other hand, consider an alternating run of length \( s + n \). Change the first and last of these symbols; the inner products will be correct provided \( \alpha = 1 \) or \( \alpha = 0 \) in the \( i \)th row of \( F \), it follows that

\[
L(C) < L(C_i) < \left\{ \begin{array}{ll}
(n(d+2) - 2), & n \text{ even} \\
(n(d+3) - 2), & n \text{ odd}.
\end{array} \right.
\]

**Corollary:** Suppose an \((n, k)\) convolutional code \( C \) over \( GF(2) \) is given with basic generator matrix \( G \). Let \( \mu \) be the maximum degree of any element in the \( i \)th row of \( F \) for \( i = 1, \ldots, L \). If \( \mu = I \), the maximum degree of any element in the dual encoder and so the bound given above is valid here also.

### IV. Examples

Consider the \((3, 2)\) code \( C \) generated by the encoder \( G \):

\[
\begin{bmatrix}
D^3 + D & D^3 + 1 & D^4 + D^2 + D + 1 \\
D^2 & D^3 + D + 1 & D^3 + D^2 + 1
\end{bmatrix}
\]

Also,

\[
\equiv \begin{bmatrix}
0 & D + 1 & D + 1 \\
1 & 1 & D
\end{bmatrix} \mod (1 + D^2).
\]

Note that the sum of its rows is congruent to \([1, D, 1]\) modulo \( 1 + D^2 \) and thus, by Theorem 1, \( C \) contains a codeword with an infinite run of alternating symbols.

As a second example, consider the \((4, 1)\) code \( C \) with generator \( F \) of its dual code given by

\[
\begin{bmatrix}
D & D^2 + D + 1 & D + 1 & D^2 + D + 1 \\
D^2 + D + 1 & D^3 + 1 & D^3 & D^2 + 1 \\
D^2 & D^2 + D + 1 & D^2 & D^3 + 1
\end{bmatrix}
\]

Thus each row of \( F \) satisfies the congruences of Theorem 2 for some value of \( \alpha \). But row 1 satisfies the congruence only for \( \alpha = 0 \) and row 3 only for \( \alpha = 1 \). Thus \( C \) does not contain a codeword with an infinite run of alternating symbols. In fact the sum of rows 1 and 3 of \( F \) has degree \( d = 3 \) it follows that the maximum run of alternating symbols in any codeword of \( C \) is bounded above by \( n(d+2) - 2 = 18 \). If we compute \( s \) here we get \( s = 14 \); so \( L < s + n - 2 = 16 \) is a little sharper. A basic generator for \( C \) is \([1 + D^2 + D^4 + D^5 + D^6 + D^7 + D^8, D^3 + D^4]

**Note:** Suppose \( n \) is even and \( L(C_i) = L(C_j) = \infty \) with \( \alpha \neq 1 \) for \( C_i \) and \( \alpha \neq 0 \) for \( C_j \). Add row \( j \) to row \( i \) in \( F \); this gives an equivalent basic encoder which has \( L(C_i) < \infty \).

So \( L < s + 2n - 2 \). As above, a finite codeword of \( C \) can be constructed containing an alternating run of length \( L = s + 2n - 2 \). It is merely necessary that positions \( n, \ldots, n + s - 1 \) of this run have inner product zero with the bit pattern of the \( f \).

Recall from the previous section the codes \( C_i \) \((n, n - 1)\) convolutional codes dual to the rows of \( F \), where \( F \) was a basic generator matrix for \( C \) and the obvious property

\[
C_{i} = \bigcap_{i=1}^{n-k} C_i
\]

from which it follows that the maximum run of alternating symbols in any codeword of \( C \) has length \( L = L(C) < \min L(C_i) \).

Suppose \( L(C_i) \) is finite for at least one value of \( i \). Then, if \( d \) is the maximum degree of any element in the \( i \)th row of \( F \), it follows that

\[
L(C) < L(C_i) \begin{cases}
(n(d+2) - 2), & n \text{ even} \\
(n(d+3) - 2), & n \text{ odd}.
\end{cases}
\]

**Proof:** Under these conditions \( C \) has a generator matrix \( F \) (a so-called minimal encoder for \( C \)) all of whose entries are of degree \( \mu < \mu \). Thus the result follows immediately except when \( n \) is even and \( L(C_i) = \infty \) for \( i = 1, \ldots, n-k \). Here if \( L \) is finite, a finite bound for it can be determined by replacing row \( i \) of \( F \) in turn by the sum of row \( i \) and row \( j, \) for \( j = 1, \ldots, n-k \) \((j \neq i)\). Of course in general all this work will not be required but the point is that such transformations do not increase the maximum degree of the elements of the dual encoder and so the bound given above is valid here also.
$+ D^5 + D^9$, $D + D^2 + D^3 + D^6 + D^7 + D^8 + D^9$
to $\mu = 9$ and the Corollary to Theorem 3 gives only the weaker
bound $n(\mu + 2) - 2 = 42$.

In the example above, the Corollary to Theorem 3 was a little
disappointing in that it gave a bound of 42 whereas more careful
examination yielded $L < 16$ (even 16 may be too high, for a
cursory examination of the bit pattern associated with the basic
generator for $C$ given above indicates that 13 may be the
answer). When $k = n - 1$ it is clear from Theorem 3 that encoders
do exist for which the bound given by the Corollary is tight. In
general there are minimal encoders whose codes have no infinite
alternating run but do possess codewords with finite alternating
runs of length $n_k + k + 1$ which compares reasonably well with the
bounds given by the Corollary. For example, consider the
$(n,k)$ convolutional encoder

$$G = \begin{bmatrix} I & 0' \\ 0 & 0 \end{bmatrix}$$

where $I$ is an identity matrix of order $k - 1$ and $0'$ is a $k - 1$ by
$n - k + 1$ matrix of zeros.

Here $p = p(D) = 1 + D + D^3$ and, for $n$ even, $q = q(D) = 1 + D^2 + D^5$ ($\mu \geq 3$) while for $n$ odd $q(D) = 1 + D^3 + D^8$ ($\mu > 4$). $G$ is
obviously basic and minimal. Further Theorem 1 guarantees that
no codeword generated by $G$ contains an infinite run of alternating
symbols. That $G$ generates a codeword with a run of alternating
symbols of length $n_k + k + 1$ can be confirmed by selecting
the inputs $x^{(0)}, \ldots, x^{(k)}$ properly. For example, let $n = 8$,
$k = 4$, and $\mu = 3$, then the bit pattern associated with the bottom
row of $G$ is

00011111 00010101 00001001 00011111.

So if $x^{(4)} = 1 + D^2 + D^3 (+ 101101 \cdots)$ and $x^{(5)} = 1 + D^2 + D^3 + 1 + D^4$ with $x^{(0)} = x^{(3)} = 0$ the codeword generated by $G$ is

00011111 01010101 01010101 01010101 01011111.

which, starting with its 8th symbol, has an alternating run of length 79 = $8 + 5$. Obviously $x^{(1)}$, $\ldots$, $x^{(k-1)}$ can always be
adjusted to fill in the first $k - 1$ symbols of each block of $n$
symbols in the proper fashion. So the input $x^{(k)}$ is the critical
one. For $n$ even, $k$ even, and $\mu$, odd, $x^{(k)} = 1 + D^2 + D^4 + D^5 + D^6 + D^7 + D^8$. Similar formulas exist for the other cases—
when $n$ is odd these vary with $\mu$ modulo 4.

As final examples consider the NASA Planetary Standard
encoders of rates $1/2$ and $1/3$. Here $G = [g_1, g_2]$ or $[g_1, g_2, g_3]$ with $g_1 = 1 + D + D^2 + D^3 + D^6$, $g_2 = 1 + D + D^2 + D^3 + D^6$, $g_3$

= $1 + D + D^2 + D^3 + D^6$. These both are basic minimal encoders
which do not possess infinite alternating runs in any codeword
as Theorem 1 easily shows. (Note that $[g_1, g_2, g_3]$ and $[g_2, g_3, g_1]$
do possess such runs, thus if infinite alternating runs are to be
avoided the outputs in $[g_1, g_2, g_3]$ must be interleaved properly).
For the rate $1/2$ code the Corollary of Theorem 3 yields $L < 29$
for a code of $n = 16$, so the significance of his result is that the signal and noise are
independent and that no pseudo-independence of the sort consid-
ered by Widrow [4] is possible.

Furthermore, the derived expression for the output variance is
intractable except for simple input densities, causing some researchers to derive approximate
formulas for some common densities. Rce [3] derives an
approximation for the input interval endpoints assuming that the
widths of these intervals are small, i.e., the number of output
levels is large. Wood [4] derives a result which states, in effect,
that the variance of the output of a minimum mean-square error
quantizer should be less than the input variance. He also states
that the significance of his result is that the signal and noise are
dependent and that no pseudo-independence of the sort consid-
ered by Widrow [4] is possible.

However, Wood's derivation assumes the input density to be
times differentiable and that the quantizer input intervals be
very small in order to truncate various Taylor series expansions.
Furthermore, the derived expression for the output variance is
dependent upon the input interval lengths and the input probabil-
ity density function evaluated at the midpoints of these interv-
als.

In this note we derive a generalization of Wood's results that
eliminates a number of his approximations and generalizes the
results to apply to more than just Max quantizers.

A Note on Optimal Quantization

JAMES A. BUCKLEW AND NEAL C. GALLAGHER, JR.,
MEMBER, IEEE

Abstract—For a general class of optimal quantizers the variance of the
output is less than that of the input. Also the mean value is preserved by
the quantizing operation.

I. INTRODUCTION

J. Max [1] is generally credited with being the first to consider
the problem of designing a quantizer to minimize a distortion
measure given that the input statistics are known. Max derives
necessary conditions for minimizing the mean square quantiza-
tion error. These results are summarized in the following equa-
tions:

$$\gamma = \int_{x_{j-1}}^{x_j} f(x) dx / P(x_{j-1} < x < x_j)$$

$$\gamma^2 \gamma^2 + \gamma^2 = \gamma$$

where $f(x)$ is the probability density of the variable to be
quantized and $P(x_{j-1} < x < x_j)$ is the probability that $x$ lies in the
interval $(x_{j-1}, x_j)$. The $y_j$ are output levels and the $x_j$ are the
break points where an input value between $x_{j-1}$ and $x_j$ is
quantized to $y_j$. Flesher [2] later gave a sufficient condition for
Max's equations to be the optimal set.

Typically, the above equations are intractable except for simple
input densities, causing some researchers to derive approximate
formulas for some common densities. Roe [3] derives an
approximation for the input interval endpoints assuming that the
widths of these intervals are small, i.e., the number of output
levels is large. Wood [4] derives a result which states, in effect,
that the variance of the output of a minimum mean-square error
quantizer should be less than the input variance. He also states
that the significance of his result is that the signal and noise are
dependent and that no pseudo-independence of the sort consid-
ered by Widrow [4] is possible.

Wood's derivation assumes the input density to be fiv-
times differentiable and that the quantizer input intervals be
very small in order to truncate various Taylor series expansions.
Furthermore, the derived expression for the output variance is
dependent upon the input interval lengths and the input proba-

bility density function evaluated at the midpoints of these interv-
als.

In this note we derive a generalization of Wood's results that
eliminates a number of his approximations and generalizes the
results to apply to more than just Max quantizers.

Manuscript received May 5, 1978; revised September 5, 1978. This work
was supported in part by the National Science Foundation under Grant
ENG-7682426 and in part the Air Force Office of Scientific Research, Air
Force Systems Command, USAF under Grant AFOSR-78-3605.

The authors are with the School of Engineering, Purdue University, West
Lafayette, IN 47907.

Acknowledgment

The authors wish to thank M. K. Simon and J. G. Smith for
bringing this problem to their attention and for suggesting
several possible approaches.

REFERENCES

correction: same journal, May 1971, page 360.)