RICE UNIVERSITY

A CODING SYSTEM USING HYBRID TREE-BLOCK CODES

by

Chung-yen Ong

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

Thesis Director's signature:

Paul E. Pfeiffer

Houston, Texas
May, 1965
ABSTRACT

This paper is devoted to introducing an idea about coding and decoding theory. A compromise between tree codes and block codes, a hybrid tree-block code, is introduced. The concepts of discarding all improbable sequences in sequential decoding and decoding a whole block at a time in block decoding are used in this code. The structure of a hybrid tree-block code and its decoding scheme are discussed. The probability of error and the number of decoding operations are considered, and a comparison with block codes and tree codes are also considered.
TABLE OF CONTENTS

Acknowledgements

I. Introduction
   1-1. General Concept of the Transmission of Information
   1-2. Block Coding
   1-3. Convolutional Code and Sequential Decoding

II. Hybrid Tree-Block Codes
   2-1. Definition
   2-2. Structure of the Tree Code Portion
   2-3. Structure of the Block Code Portion

III. Decoding Scheme

IV. Analysis
   4-1. A Bound on the Probability of Error
   4-2. Number of Decoding Operations

V. Comparison
   5-1. Compare with Block Code
   5-2. Compare with Tree Code

References
ACKNOWLEDGEMENTS

I consider myself fortunate to have had the support of so many people while I worked on this thesis. I would like to express my most sincere appreciation to my adviser, Dr. Paul E. Pfeiffer, for his helpful enlightenment and continuous encouragement. I wish to extend my gratitude to Mr. Shu Lin for his valuable suggestions. Many thanks are due to my fellow graduate students, Mr. Wai-hung Ng and Mr. Henry Lyne for their kind help. This work was supported under a grant from the U. S. Army Research Office -- Durham.
INTRODUCTION

1-1. General Concept of the Transmission of Information

In this paper, we are interested in a class of good codes and a suitable decoding scheme for efficient communication. We are concerned only with the transmission through a binary symmetric channel (BSC). A binary symmetric channel is defined by the transition probability diagram as shown in Fig. 1-1. The channel accepts binary symbols (0,1) at its input and produces binary symbols at its output. Each input symbol has a probability $p_0 < 1/2$ of being received incorrectly, and a probability $q_0 = 1 - p_0$ of being received correctly. The transition probability $p_0$ is constant and independent of the value of the symbol being transmitted; the channel is as likely to change a "1" into a "0" as to change a "0" into a "1".

![Binary Symmetric Channel Diagram]

Fig. 1-1 Binary Symmetric Channel

Since the information to be transmitted through a channel can be regarded as consisting of binary digits,
the two parameters that characterize the transmission process are:

(a) The rate of transmission \( R \). Assuming that digits are transmitted over the channel at a given time rate, a measure of the rate of information transmission is
\[
R = \frac{\beta}{\alpha},
\]
where \( \alpha \) is the number of channel digits used to represent \( \beta \) message digits.

(b) The accuracy of reception as indicated by the probability of error per digit \( P(\epsilon) \); that is, the probability that any particular binary digit will be correctly reproduced by the channel decoder.

An important characteristic of the encoding process is the number of binary digits that enter and leave the encoder simultaneously. The digits may enter and leave the encoder one by one, or \( k \) digits may enter the encoder together at one time while the \( k \) preceding digits are taken out of it. For instance, if \( R_t \) is the number of binary digits per second to be transmitted, we may choose to feed the binary digits to the encoder one by one every \( \frac{1}{R_t} \) seconds, or two at a time every \( \frac{2}{R_t} \) seconds, or \( k \) at a time every \( \frac{k}{R_t} \) seconds, where \( k \leq \nu \). (\( \nu \) is the number of binary digits of message). Obviously, \( k\alpha \) encoded channel digits are taken out of the encoder every \( \frac{k}{R_t} \) seconds. We refer to the extreme case of \( k=\nu \) as block encoding. The encoding process is said to be sequential when \( k < \nu \) because of the sequential character of the
dependence of the resulting channel input on its own past. It is clear that if the binary digits enter the encoder \( k \) at a time, each successive block of \( k \) binary digits acts as a single unit in determining the channel input and, therefore, its output. Thus the binary digits must be reconstructed by the decoder in blocks of the same size \( k \). We shall refer to the corresponding decoding operations as block decoding and sequential decoding, respectively.

The Hamming distance between two code words is defined to be the number of positions in which the code words differ.
1-2. **Block Coding**

A block code is a code which maps each block of the symbols of the source alphabet into a fixed sequence of symbols of the code alphabet. These fixed sequences of the code alphabet are called code words. For a binary case, these symbols may be represented by "0" and "1".

A schematic representation of block encoding and decoding is shown in Fig. 1-2. Here, M is the message space, U is the channel input space, V is the channel output space, W is the discrete space formed by the M subsets. The choice of the code in the channel input space will govern the probability of decoding error. The code which gives the minimum probability of error is called an optimum code.

![Schematic representation of block encoding and decoding](image)

**Fig. 1-2**

To decode, we usually compare the received sequence with every code word and decode the most probable one (i.e. the one which has minimum Hamming distance from the received sequence) as the transmitted sequence. This
procedure is called maximum-likelihood decoding. If the number of code words is very large and the probability of decoding error should be small, then the number of comparisons will be tremendously large.

Block encoding and decoding has the advantage of making the transmission of each block of \( \nu \) binary digits independent of that of all preceding and following blocks, at least in the case of zero memory channels, so there is no decoding error propagation.
1-3. **Convolutional Code and Sequential Decoding**

For purpose of simplicity, we consider a convolutional code constructed by a single generator sequence $g=(g^1, g^2, g^3, \ldots, g^n)$ of $n$ binary digits and assume the transmission rate $R=1/\alpha$, where $n=\nu\alpha$ ($\nu$ is an integer).

We call the digits (represents a message digit) a segment, so that there are $\nu$ segments in the generator sequence $g$. We use this generator sequence $g$ to construct infinite row vectors as follows:

$$R_i^k, R_i^{k+1}, \ldots, \text{where } R_i^k = g^k(i-1) \quad \text{for } 1 \leq k-(i-1) \nu \leq n$$

$$= 0 \quad \text{otherwise}$$

By using these infinite row vectors, we can form a matrix $[G]_\infty$ with infinite dimensions as shown in Fig. 1-3. Use this matrix $[G]_\infty$ as a generator matrix; the row space of $[G]_\infty$ will form an infinite group code. If $m$ is message vector (a row vector), then its corresponding code vector is $u = m[G]_\infty$.

![Fig. 1-3](image-url)
The infinite group code generated by \([G]_\infty\) has an infinite tree structure with nodes spaced \(d\) digits apart and two branches stemming from each node.

The encoding procedure is one in which the transmitter traces a particular path through the tree under the instructions of the information digits. The decoding operation is the process of determining from the received sequence which path in the tree was traced by the encoder. At each node, the decoder makes a decision as to which branch was traced by the encoder; this leads to a decoding scheme by which the decoder tries to decode the information sequence digit by digit. This decoding scheme is called sequential decoding. It was introduced by Wozencraft [2]. Under this decoding scheme, once an error is made, the error has a tendency to propagate indefinitely. The propagation of error will stop if and only if the decoder can decode the next \(\frac{1}{2}\) information digits correctly (where \(\nu\) is the number of segments in the generator sequence).

We define the minimum distance \(D_{\text{min}}\) of a tree code \(S\) as the minimum distance between the upper half and lower half of the tree, i.e.

\[
D_{\text{min}} = \min_{u_1 \in S^\circ, u_j \in S^1} d(u_1, u_j)
\]

where \(S^\circ\) and \(S^1\) are the upper and lower subsets of the tree code and \(d(u_1, u_j)\) is the Hamming distance between \(u_1\) and \(u_j\).
We also define a subtree, with length of $k$ segments, of a tree as a finite tree of $k$ segments long stems from any node of the tree. The enclosed unit in Fig. 1-4 is a subtree of three segments which stems from node (2).

Fig. 1-4
II

HYBRID TREE-BLOCK CODES

2-1. Definition

We define a hybrid tree-block code (we shall abbreviate it HTB code) as a combination of a tree code and a block code obtained by using a special block code as the tail of a tree code, i.e., it is a block code with tree structure. That special block code has such a property that it makes the minimum distance of this HTB code (the minimum distance between any two code words of the HTB code) equal to the minimum distance of the tree code. The form of such a code can be illustrated in Fig. 2-1, where the generator sequence of the tree code is \( g = 111 \ 101 \ 110 \ 001 \), and the generator matrix of the block code is

\[
G = \begin{bmatrix}
1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

We use the same block code at the tail of the upper half and lower half of the tree code. The minimum distance of the tree code portion is 6, and the minimum distance of the HTB code is also 6.
Fig. 2-1 Hybrid tree-block code
2-2. Structure of the Tree Code Portion

To develop the tree code portion, we can use the rules for constructing a general tree code proposed by Shu Lin and Henry Lyne. [7] and [8]

We select the generator sequence of the tree code segment by segment, according to the following four rules:

(1) Maximize the minimum weight path.
(2) Minimize the number of minimum weight paths.
(3) Minimize the difference between the minimum and maximum weight path.
(4) Minimize the number of maximum weight path.

We first accept, at any stage, the generator segments which give maximum minimum distance. We then select from these the segments which give minimum number of minimum weight paths. We then select from these the segments which give minimum difference between the minimum and maximum weight path and we then select from these the segment which gives the minimum number of maximum weight paths as the generator segment.
2-3. Structure of the Block Code Portion

In this section, we are going to describe a way to construct the block code portion which will make the minimum distance of the HTB code equals to the minimum distance of the tree code portion. In other words, the distance between any two code words of the HTB code is at least equal to the minimum distance of the tree code portion. For instance, in the example of Fig. 2-1 for the tree code portion, the minimum distance between the sequences (0) and (1), (2) and (3), (4) and (5), (6) and (7), (8) and (9), (10) and (11), (12) and (13), (14) and (15) are 3; among the sequences (0) (1) and (2) (3), (4) (5) and (6) (7), (8) (9) and (10) (11), (12) (13) and (14) (15) are 4; among sequences (0) to (3) and (4) to (7), (8) to (11) and (12) to (15), are 5; among sequences (0) to (7) and (8) to (15) is 6. Therefore, we need to construct a block code with minimum distance 3 between those code words which have minimum distance 3 in the corresponding tree code portion, with minimum distance 2 among those code words which have minimum distance 4 in the corresponding tree code portion, with minimum distance 1 among those code words which have minimum distance 5 in the corresponding tree code portion, and minimum distance 0 among those codes which have minimum distance 6 in the corresponding tree code portion. Therefore, we can use the same block code at the tail of the upper half and lower half of the tree code.
We define the order of pair of a block code as shown in Fig. 2-2 and define the distance of the pair as the minimum distance between the upper half and lower half code words of the pair, assume the same order of pairs have the same minimum distance, and use a symbol $D_k$ to represent it, where $k$ is the order number of the pair.

As shown in Fig. 2-2 $D_1=4$, $D_2=3$, $D_3=2$, and $D_4=1$.

![Diagram showing a tree code with different order pairs and their distances]

**Theorem**: Suppose a tree code of length $V$ segments is generated by the matrix $[G]$ having row vectors $g_1, g_2, g_3, \ldots, g_V$. If the message sequence matrix is $[m_1, m_{v-1}, \ldots, m_1]$, the vector $g_1$ is the $2^{l-1}$th path of the tree code which has paths numbered from 0 to $2^v-1$. Then a block code generated by a generator matrix $[H]$ with row vectors $h_1, h_2, h_3, \ldots, h_V$
will have the distance property $D_k \geq d(y-k+1)$, where
$$h_j = g_1 \oplus g_2 \oplus g_3 \oplus \cdots \oplus g_{y-j+1}$$
for $1 \leq j \leq y$ and $d(1)$ is the minimum distance of the tree code of $i$ segments long.

Proof: We use $[n]$ to represent the n-th code word of the block code, which has $2^y$ code words, from
[0] to $[2^y-1]$, then $h_j = [2^{j-1}]$.

By the structure of the code, the first $2^{j-1}$ code words do not use the $j$-th generator sequence $h_j$. The next $2^{j-1}$ code words are obtained by adding, in succession, the generator sequence $h_j$ to the first $2^{j-1}$ code words. If $[n]$ is a generator sequence (i.e. if it is $h_j$ for some $j$), then $[n-x]$, where $0 < x \leq n-1$, is a code word above it in the list. Thus $[n] \oplus [n-x]$ is the code word $[2n-x]$.

That is, $[2^{j-1}] \oplus [2^{j-1}-x] = [2^{j-x}]$.

For the $k$-th order pair, the distance is the minimum distance between the following two sets of code words:

$S_u = \{ [2^k(m-1)], [2^k(m-1)+1], \ldots, [2^k(m-1)+2^{k-1}-1] \}$

$S_l = \{ [2^k-1(2m-1)], [2^k-1(2m-1)+1], \ldots, [2^k-1(2m-1)+2^{k-1}-1] \}$

where $1 \leq m \leq 2^{y-k}$ (Note that $2^k(m-1) = (2m-2)2^{k-1}$)

The members of $S_u$ are the code words in the upper half of the $k$-th order pairs and the members of $S_l$ are the code words in the lower half. For each value of $m$, we have a pair code words. Among the code words in $S_l$ only $[2^k-1(2m-1)]$ can be a generator sequence, and this is only when $m=1$.

Let $0 \leq S \leq 2^{k-1}-1$

$0 \leq T \leq 2^{k-1}-1$
Then, the code words in the upper half and lower half of the \( k \)th order pair can be represented by \([2^{k(m-1)}+T]\) and \([2^{k-1}(2m-1)+S]\) respectively.

(1) In the case \( m=l \) and \( S=0 \)

\[
\left|[2^{k-1}(2m-1)+S] \oplus [2^{k}(m-1)+T]\right| = \left|[2^{k-1}] \oplus [T]\right| \\
= |h_k \oplus [T]| \\
= |s_1 \oplus s_2 \oplus \ldots \oplus s_{\nu-k+1} \oplus [T]| \\
\]

(A) If \( T=0 \), then

\[
\left|[2^{k-1}] \oplus [T]\right| = \left|s_1 \oplus s_2 \oplus \ldots \oplus s_{\nu-k+1}\right| \geq d(\nu-k+1) \\
\]

(B) If \( 0<T<2^{k-1}-1 \), then \([T]\) must be the sum of some \( h_j's \), where \( j<k \); that means \([T]\) is the sum of some \( s_i's \) where \( i>\nu-k+1 \).

Thus \( \left|[2^{k-1}] \oplus [T]\right| \geq d(\nu-k+2) \geq d(\nu-k+1) \). Therefore,

if \( m=1 \) and \( S=0 \),

\[
\left|[2^{k-1}(2m-1)+S] \oplus [2^{k}(m-1)+T]\right| \geq d(\nu-k+1) \\
\]

(2) In the case \( m=1 \) and \( S \neq 0 \)

\[
\left|[2^{k-1}(2m-1)+S] \oplus [2^{k}(m-1)+T]\right| = \left|[2^{k-1}+S] \oplus [T]\right| = \left|[2^{k-1}] \oplus [S] \oplus [T]\right| \\
= |s_1 \oplus s_2 \oplus \ldots \oplus s_{\nu-k+1}| \geq d(\nu-k+1) \\
\]

(A) If \( S=T \), then

\[
\left|[2^{k-1}] \oplus [S] \oplus [T]\right| = \left|[2^{k-1}]\right| = |h_k| = |s_1 \oplus s_2 \oplus \ldots \oplus s_{\nu-k+1}| \geq d(\nu-k+1) \\
\]

(B) If \( S \neq T \), then \([S] \oplus [T]\) must be a code word above \([2^{k-1}]\) in the list and it is the sum of some \( h_j's \), where \( j<k \); that means it is the sum of some \( s_i's \), where \( i>\nu-k+1 \).
Thus \( |[2^{k-1}] \oplus [S] \oplus [T]| \geq d(y-k+2) \).

Therefore, if \( m=1 \) and \( S \neq 0 \)

\[
|[2^{k-1}(2m-1)+S] \oplus [2^k(m-1)+T]| \geq d(y-k+1)
\]

(3) In the case \( m=2 \)

Since the code words in the second pair \( (m=2) \) are obtained by adding, in succession, the generator sequence \( h_{k+1} \) to the first \( 2^k \) code words, it must have the same distance property as the first pair \( (m=1) \).

(4) In the case \( m=3 \) or \( m=4 \)

Since the code words in these two pairs are obtained by adding, in succession, the generator sequence \( h_{k+2} \) to the first \( 2^{k+1} \) code words, so they must have the same distance property as \( m=1 \) and \( m=2 \).

Similarly, we can show that for \( 1 \leq m \leq 2^{\nu-k} \)

\[
|[2^{k-1}(2m-1)+S] \oplus [2^k(m-1)+T]| \geq d(y-k+1).
\]

Therefore, \( D_k \geq d(y-k+1) \)

QED

According to this theorem, we can construct the block code portion by using the generator matrix of a tree code with the required distance property.

Suppose we want to construct a block code at the tail of a tree code of length \( y = 5 \) (where \( y \) is the number of segments of the tree code) with the following distance property: \( d(1)=3, d(2)=4, d(3)=5, d(4)=6 \) and \( d(5)=7 \), where \( d(k) \) is the minimum distance of a tree code of \( k \)
segments long. In order to satisfy the required minimum distance among all code words of the desired HTB code ($d_{\text{min}}=7$), we construct another tree code with the following distance property: $d(1)=2$, $d(2)=3$ and $d(3)=4$. Develop each sequence into two by adding one with "0", another one with "1" at the tail, such as shown in Fig. 2-3. Give each sequence a number, from (0) to (15). The generator sequences of this tree code are sequences (1), (2), (4), and (8). We use

$$\begin{pmatrix}
(1) \oplus (2) \oplus (4) \oplus (8) \\
(1) \oplus (2) \\
(1) \\
(1)
\end{pmatrix} =
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 
\end{pmatrix}$$

as a generator matrix to generate a block code. Put this block code at the upper half and lower half of the tail of the tree code, then we complete the structure of the HTB code.

![Fig. 2-3](image-url)
With this procedure, if we know the form of some tree codes with maximum minimum distance, then we can use one of these tree codes as the tree code portion and use another one to construct a block code as the block code portion of the HTB code. The data in Table 2-1 were found by Henry Lyne[8].

| \( \nu \) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| \( d(\nu) \) | 4 | 6 | 8 | 9 | 10 | 11 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

Table 2-1

By using the data in Table 2-1 and the procedure as described above, the needed length of block code portion \( (n_b \text{ digits}) \) for different value of \( \nu \) can be obtained. An example for \( \nu = 7, d = 3 \), the needed value of \( n_b \) is obtained as follows:

From Table 2-1, for \( d = 3 \), we find \( d(7) = 9 \), \( d(6) = 8 \), \( d(5) = 7 \), \( d(4) = 6 \), \( d(3) = 5 \), \( d(2) = 4 \) and \( d(1) = 3 \). At first, we construct a tree code with distance property: \( d(1) = 3 \), \( d(2) = 4 \), \( d(3) = 5 \) and \( d(4) = 6 \). From Table 2-1, this is corresponding to a tree code with \( \nu = 4, d = 3 \). Then, develop each sequence of this tree code into four sequences by adding three digits at the tail which give the distance between the first and second sequence, third and fourth sequence equal to 1, between the first two and last two sequences equal to 2. Therefore, the needed length of \( n_b \) is 15. The values of \( n_b \) for different \( \nu \) and the corresponding transmission rate \( (R) \) of the HTB code are found as shown in Table 2-2 and Table 2-3. Here \( n_t \) is the number of digits of the tree code portion, and \( p(\xi) \) is the upper bound.
For $\alpha = 3$, $p_0 = 0.01$, $q_0 = 0.99$

<table>
<thead>
<tr>
<th>$V$</th>
<th>$d(V)$</th>
<th>$n_b$</th>
<th>$n_t$</th>
<th>$n_b + n_t$</th>
<th>$R = \frac{V}{n_b + n_t}$</th>
<th>$P(\epsilon)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0.3333</td>
<td>2.980 x 10^-4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>0.2857</td>
<td>2.031 x 10^-4</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
<td>9</td>
<td>11</td>
<td>0.2727</td>
<td>1.553 x 10^-4</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>0.2352</td>
<td>6.121 x 10^-4</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>8</td>
<td>15</td>
<td>23</td>
<td>0.2174</td>
<td>7.605 x 10^-5</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>12</td>
<td>18</td>
<td>30</td>
<td>0.2000</td>
<td>2.225 x 10^-4</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>15</td>
<td>21</td>
<td>36</td>
<td>0.1944</td>
<td>2.911 x 10^-5</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>15</td>
<td>24</td>
<td>39</td>
<td>0.2051</td>
<td>4.336 x 10^-5</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>19</td>
<td>27</td>
<td>46</td>
<td>0.1957</td>
<td>9.743 x 10^-5</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>23</td>
<td>30</td>
<td>53</td>
<td>0.1887</td>
<td>1.534 x 10^-5</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>27</td>
<td>33</td>
<td>60</td>
<td>0.1833</td>
<td>3.152 x 10^-5</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>27</td>
<td>36</td>
<td>63</td>
<td>0.1905</td>
<td>4.170 x 10^-5</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>31</td>
<td>39</td>
<td>70</td>
<td>0.1857</td>
<td>6.908 x 10^-6</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>35</td>
<td>42</td>
<td>77</td>
<td>0.1818</td>
<td>1.304 x 10^-5</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>39</td>
<td>45</td>
<td>84</td>
<td>0.1785</td>
<td>2.218 x 10^-6</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>39</td>
<td>48</td>
<td>87</td>
<td>0.1839</td>
<td>2.896 x 10^-6</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>43</td>
<td>51</td>
<td>94</td>
<td>0.1809</td>
<td>5.185 x 10^-6</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>43</td>
<td>54</td>
<td>97</td>
<td>0.1856</td>
<td>6.555 x 10^-6</td>
</tr>
</tbody>
</table>

Table 2-2
For $\alpha = 4$, $p_0 = 0.01$, $q_0 = 0.99$

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>$d(\gamma)$</th>
<th>$n_b$</th>
<th>$n_t$</th>
<th>$n_b + n_t$</th>
<th>$R = \frac{\gamma}{n_b + n_t}$</th>
<th>$P(\varepsilon)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.2500</td>
<td>$5.920 \times 10^{-4}$</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>0.2000</td>
<td>$1.138 \times 10^{-5}$</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>6</td>
<td>12</td>
<td>18</td>
<td>0.1667</td>
<td>$2.735 \times 10^{-6}$</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>9</td>
<td>16</td>
<td>25</td>
<td>0.1600</td>
<td>$4.496 \times 10^{-6}$</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>9</td>
<td>20</td>
<td>29</td>
<td>0.1724</td>
<td>$9.720 \times 10^{-6}$</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>15</td>
<td>24</td>
<td>39</td>
<td>0.1538</td>
<td>$2.458 \times 10^{-6}$</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>21</td>
<td>28</td>
<td>49</td>
<td>0.1439</td>
<td>$5.945 \times 10^{-7}$</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>24</td>
<td>32</td>
<td>56</td>
<td>0.1429</td>
<td>$1.510 \times 10^{-7}$</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>27</td>
<td>36</td>
<td>63</td>
<td>0.1429</td>
<td>$2.374 \times 10^{-7}$</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>31</td>
<td>40</td>
<td>71</td>
<td>0.1408</td>
<td>$6.074 \times 10^{-7}$</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>34</td>
<td>44</td>
<td>78</td>
<td>0.1410</td>
<td>$9.790 \times 10^{-8}$</td>
</tr>
<tr>
<td>12</td>
<td>18</td>
<td>38</td>
<td>48</td>
<td>86</td>
<td>0.1395</td>
<td>$2.298 \times 10^{-7}$</td>
</tr>
<tr>
<td>13</td>
<td>19</td>
<td>42</td>
<td>52</td>
<td>94</td>
<td>0.1383</td>
<td>$4.210 \times 10^{-8}$</td>
</tr>
</tbody>
</table>

Table 2-3
of probability of error for transition probability \( p_0 = 0.01 \) by formula

\[
P(\epsilon) \leq \sum_{k=t+1}^{n} \binom{n}{k} (p_0)^k (q_0)^{n-k}
\]

where \( t \) is an integer such that \( \frac{d(\nu)}{2} > t \geq \frac{d(\nu)}{2} - 1 \).

The encoding scheme of an HTB code is the same as that of a tree code except for adding a tail.

The HTB code is a group code. The tree portion is determined by the \( \nu \) message digits. The block portion is determined only by the last \( \nu - 1 \) message digits. This is equivalent to saying that the block code tail is the same for any given position in the upper half of the code and the corresponding position in the lower half of the code.

If \( [G_t] \) is the generator matrix for the tree portion (with \( \nu \) rows) and \( [H_b] \) is the generator matrix for the block portion in either half, then the matrix

\[
\begin{pmatrix}
G_t & H_b \\
0 & 0 & \cdots & 0
\end{pmatrix}
\]

is the generator matrix for the entire code. The resulting code must therefore be a group code. For example, in Fig. 2-1, the generator matrix for the tree portion is

\[
\begin{bmatrix}
000 & 000 & 000 & 111 \\
000 & 000 & 111 & 101 \\
000 & 111 & 101 & 110 \\
111 & 101 & 110 & 001
\end{bmatrix}
\]

the generator matrix for the block portion is

\[
\begin{bmatrix}
1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

and the generator matrix for the entire code is

\[
\begin{bmatrix}
000 & 000 & 000 & 111 & 1101 \\
000 & 000 & 111 & 101 & 00111 \\
000 & 111 & 101 & 110 & 00001 \\
111 & 101 & 110 & 001 & 00000
\end{bmatrix}
\]
We define a b-unit as a subtree which is b segments long. It is desirable to choose b so that the probability of channel noise exceeds $T(b)$, where $d(b) > T(b) > \frac{d(b)}{2}$, in a b-channel digits is reasonably small and yet the number of paths in the b-unit is not too large. For instance, when $\alpha = 3$, $b = 7$ is a good choice. In this case, the probability of channel noise exceeding $T(7)$ is $5 \times 10^{-5}$ and the number of $b$ th nodes is $2^7 = 128$ for binary symmetric channel with transition probability $p_o = 0.02$. (4)

At first, we use branch-by-branch search method to find a path in the first b-unit (first b segments in the tree). That means we compare the received sequence with the tree branch-by-branch; at each node we select the nearest branch which is the “more probable” one. After having found a path in the b-unit, we then compare this path with the corresponding portion of $X$. If their distance is less than or equal to $T(b)$, then we shift in one segment and shift out the first segment. If their distance is greater than $T(b)$, then we must search the whole b-unit by using $T(b)$ as a discard function to discard those sequences which have distance from the corresponding portion of $Y$ exceeds $T(b)$ and select the first sequence in the b-unit which has distance from the corresponding portion of $Y$ less than or equal to $T(b)$ as the tentatively decoded path. The decoding procedure continues like this until the last segment.
of the tree code portion. Every time when we have searched \( kb \) segments, where \( 1 \leq k \leq \sqrt{b} \), we use threshold \( T(kb) \) to test its distance from the corresponding portion of \( \mathcal{V} \), i.e. we use Multiple-threshold test. The detail of this decoding scheme is shown in reference (4). After having decoded to the last \( b \)-unit of the tree code portion, we compare all sequences in the last \( b \)-unit (including the block code portion) with \( \mathcal{V} \), decode the first sequence which has distance with \( \mathcal{V} \) less than or equal to \( T(\nu) \) as the transmitted sequence. If there is no one, we extend this \( b \)-unit one segment longer and compare all sequences in this \((b+1)\)-unit with \( \mathcal{V} \). If we still can not decode one sequence in this \((b+1)\)-unit, then we extend this \((b+1)\)-unit one more segment longer and compare all sequences in this \((b+2)\)-unit with \( \mathcal{V} \), etc. If we still can not decode one sequence after compare all \( 2^\nu \) sequences with \( \mathcal{V} \), then we decode the most probable one as the transmitted sequence. Refer to Fig. 2-5.

![Diagram of decoding process](image-url)
IV

ANALYSIS

4-1. A Bound on the Probability of Error

We define the minimum distance of an HTB code as the minimum distance between any pair of its code words and use the symbol $d_m$ to represent it. According to the structure of HTB code as described in part II, $d_m$ is equal to the minimum distance of the tree code portion, $d(V)$, so the minimum distance of HTB code depends on the minimum distance of the tree code.

Let $\mathcal{E}$ be the event of a decoding error, $d_0$ be the integer such that

$$\frac{d_m}{2} - 1 \leq d_0 \leq \frac{d_m}{2}$$

(i.e., this is the error-correcting capability of the code)

$T$ be the event that the weight of the error pattern less than or equal to $d_0$.

Then, $P(\mathcal{E}) = P(\mathcal{E} | T)P(T) + P(\mathcal{E} | T\mathcal{C})P(T\mathcal{C})$

Since $P(\mathcal{E} | T) = 0$

Thus, $P(\mathcal{E}) = P(\mathcal{E} | T\mathcal{C})P(T\mathcal{C})$

$$\leq P(T\mathcal{C})$$

$$\leq \sum_{k=d_0+1}^{n} \binom{n}{k} p_0^k q_0^{n-k}$$

Where $p_0$ is the transition probability

$q_0 = 1 - p_0$

$n$ is the number of digits in the code word.
Therefore, the probability of error of a HTB code will depend upon its minimum distance. For a given number of code words, if we want to increase the minimum distance of the HTB code, we should increase $\alpha$, but this will decrease the transmission rate. This can be seen if we investigate Table 2-2 and Table 2-3.
4-2. Number of Decoding Operations

The problem of the number of decoding operations is very complicated, so it is very difficult to calculate it exactly. We shall consider only a few special cases in this paper.

We define one decoding operation to be the generation of one segment of the decoding sequence in the decoder plus applying the threshold test to the appropriate path.

The maximum number of operations in the first $b$ segments is the number of branches in such a unit. It is given by

$$2+2^2+2^3+\ldots+2^b=2^b+1-2=2^b+1=N_b$$

Let $N$ be the average number of decoding operations for decoding one code word.

(1) If $|E(\alpha)|<\frac{d}{2}$ and $|E(n_t+n_b)|<\frac{d}{2}$

Where $|E(\alpha)|$ and $|E(n_t+n_b)|$ are the weight of the noise pattern of $\alpha$ and $n_t+n_b$ digits long. Then, the number of decoding operations for decoding one code word is $\nu+1$

(2) If $|E(kb)|<T(kb)$ for $1\leq k \leq \frac{\nu}{b}$

and $|E(n_t+n_b)|<\frac{d}{2}$

Where $|E(kb)|$ is the weight of the noise pattern of $kb$ segments long.

Then, the number of decoding operations for decoding one code word $\leq (\nu-b+1)N_b+2^b \leq (\nu-b+\frac{3}{2})2^{b+1}$
(3) For general case,

Let \( \nu = k_i \cdot (ib) + a_i = kb \)

\[
N < \sum_{\frac{\alpha}{2} < t \leq \frac{\alpha}{2}} + 1 \left( \frac{\nu}{b} \right) C_t p_0 q_0^{d-t} \left\{ \left( (k_i-1)ib + a_i \right) 2^{ib+1} \right\}
\]

\[
+ \sum_{r=0}^{\frac{\nu}{b}} C_r p_0 q_0^{r} \left\{ \left( (k_2-1)2b + 1 + a_2 \right) 2^{2b+1} \right\}
\]

\[
+ \sum_{r=0}^{\frac{\nu}{b}} C_r p_0 q_0^{r} \left\{ \left( 2^{2b} + 2^{3b} + \ldots \right) \right\}
\]

\[
+ \ldots + \left( 2^{\nu+1} + 2^\nu \right) \right\}
\]

\[
\sum_{\frac{\alpha}{2} < t \leq \frac{\alpha}{2}} + 1 \left( \frac{\nu}{b} \right) C_t p_0 q_0^{d-t} \left\{ \sum_{i=1}^{\frac{\nu}{b}} \left( (k_i-1)ib + a_i \right) 2^{ib+1} \right\}
\]

\[
+ \sum_{r=0}^{\frac{\nu}{b}} C_r p_0 q_0^{r} T(\nu) - r \sum_{i=1}^{\frac{\nu}{b}} i 2^{ib} \right\}
\]

\[
\sum_{\frac{\alpha}{2} < t \leq \frac{\alpha}{2}} + 1 \left( \frac{\nu}{b} \right) C_t p_0 q_0^{d-t} \sum_{i=1}^{\frac{\nu}{b}} \left( (k_i-1)(ib)+a_i \right) 2^{ib+1} \right\}
\]

\[
+ i 2^{ib} \sum_{r=0}^{\frac{\nu}{b}} C_r p_0 q_0^{r} T(\nu) - r \right\}
COMPARISON

5-1. Compare with Block Code

Because of the tree structure of the tree code portion, we can use the concept of sequential decoding to discard the "improbable" sequences as described in part III. A few of the sequences will be retained at the end of the tree code portion. Therefore, we need only to compare the retained sequences with the received sequence instead of every code word as maximum likelihood decoding for a block code. Thus we can save a great many decoding operations as compared with block code. Since the problem of calculating the number of decoding operations is too complicated, we shall not compare them exactly.

Since we add the tree code portion to the block, so for the same message digits and error-correcting capability, we need a longer code as compared with the block code. If they have the same message digits and same code length, the probability of error of an HTB code will be greater than that of a block code.

A comparison of HTB codes with some known optimum block codes are shown in Table 5-1.
Where $k$ is the number of message digits, $t$ is the maximum number of error digits can be corrected.

* Refer to Table 2-2 and Table 2-3.

** See reference [5] Table 5.2, Table 9.1 and Table 9.2.
5-2. Compare with Tree Code

An HTB code has the property of a block code that the decoding of each code word is independent of that of all preceding and following code words. Therefore, it has no error propagation as that in sequential decoding for a tree code.

Since we decode the whole block instead of only one digit at a time as in the sequential decoding, so it can be decoded faster than the sequential decoding.

Since we need to add the block code portion which does not carry any message, so the transmission rate of a HTB code is less than that of a tree code. As described in part II, the block code portion is constructed by another tree code, but the length of this tree code will always less than the first tree code, so the transmission of a HTB code will not be less than one half of that of the tree code.

From section 4-2, we can see that for the case $|E(\alpha)| < \frac{d}{2}$ and $|E(n_t+n_b)| < \frac{d_m}{2}$, the number of decoding operations of an HTB code ($N=\nu+1$) is much less than that of a block code ($N=2^\nu$), especially for $\nu$ very large. For the case $|E(kb)| < T(kb)$ and $|E(n_t+n_b)| < \frac{d_m}{2}$, the number of decoding operations of an HTB code ($N \leq (\nu-b+2)2^b$). Therefore, the HTB code is especially useful in low noise communication.
REFERENCES

4. Paul E. Pfeiffer and Shu Lin, "A Sequential Decoding System Utilizing Distance Properties of Convolution Tree Codes".