THE PERFORMANCE ANALYSIS AND DECODING
OF HIGH DIMENSIONAL TRELLIS-CODED MODULATION
FOR SPREAD SPECTRUM COMMUNICATIONS

A Dissertation Presented to
The Faculty of
School of Electrical Engineering and Computer Science
Fritz J. and Dolores H. Russ
College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Changlin Chen
August, 1997
This is dedicated to my wife, my daughter,

my parents,

and the memory of my grand-parents
ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Jeffrey C. Dill for his efforts and encouragements in directing this research project. His optimism shall never be forgotten. Successful completion of this work would not have been possible without his guidance and counseling. His effort of getting us to attend a short course on Turbo Codes at UCLA is also appreciated.

I would also thank the members of the dissertation committee for their helpful insights. These members were: Drs. Joseph Essman, Hollis C. Chen, Jeffrey Giesey, and Sergio Lopez-Permouth. Also I thank Dr. Chen and Dr. Giesey for their references and Dr. Lopez for mathematical discussions.

I would like to express my appreciation to Dr. John A. Tague, my former advisor who is on his two-year sabbatical leave, for his effort and advice in the Automatic Target Recognition Project, and his fifteen months continuous research assistantship. His academic seriousness, his support of my decision of changing advisor and his interesting lectures are remembered.

I would like to thank Mr. Yung-Cheng Lo, my partner in this project, for his help when I changed my advisor and for our countless meetings and discussions about the research. Without this partnership, I would not have been able to make such progress. I am grateful of Dr. Alan R. Lindsey, Rome Laboratory, New York, for
his advice, his help, and his initiative in getting us to go to UCLA. I thank Dr.
Ebrahim Mokari, former Graduate Chairman, and the committee for granting me
the Stocker Initiation Award. I thank the interview with Mr. Nam and Mr. Farrokh
at ioWave, Inc. that inspired me gaining more insight into the HDTCM scheme later.
I also thank those friends I made here, especially Fujihito Hamaguchi, Jian-Jiang
Pan, Chien-hsien Wen, etc.

I thank my family. It is their love that has been strengthening me days and nights,
although I am far away from them. I thank my beloved parents for their unconditional
love, support, sacrifice and anticipation. They are always there when I need them.
I also thank my parents-in-law and my brother for their help, encouragement and
advice. I thank my little daughter, Mae, for her loveliness. She brings us joy and
brightens our lives. I especially thank my wife, Ning. Without her love, patience,
encouragement and support, everything would not have been possible. She is the
person whom I have already made my best discoveries. Finally, I thank the All
Mighty that bless me with such a loving, unite and strong family.

Life is a journey. There is hope where there is love. With hope and hard work,
life is always bright no matter how hard it could be.
# Table of Contents

**DEDICATION** .......................................................... 3

**ACKNOWLEDGEMENTS** .................................................. 4

**LIST OF TABLES** ...................................................... 9

**LIST OF FIGURES** ................................................... 12

**CHAPTER**

| I  | Introduction .................................................................. 1 |
|    | 1.1 History .................................................................. 1 |
|    | 1.2 High Dimensional Trellis-Coded Modulation ................. 2 |
|    | 1.3 Contributions ................................................... 3 |
|    | 1.4 Overview of the Dissertation ................................... 5 |

| II | Background .................................................................... 7 |
|    | 2.1 Conventional Digital Communication Systems ................. 7 |
|    | 2.2 Conventional Coding and Modulation ............................ 8 |
|    | 2.2.1 Modulation and demodulation ................................. 8 |
|    | 2.2.2 Error correcting codes ....................................... 9 |
|    | 2.3 Trellis-Coded Modulation ........................................ 12 |
|    | 2.3.1 TCM vs. conventional coding and modulation ............. 13 |
|    | 2.3.2 Ungerboeck’s mapping by set partitioning ................ 17 |
|    | 2.4 Decoding of TCM: The Viterbi Algorithm .................... 19 |
|    | 2.5 Spread Spectrum Communications ............................... 22 |
|    | 2.6 Bi-orthogonal signaling ........................................ 24 |
III High Dimensional Trellis-Coded Modulation ........................................ 27
  3.1 The Encoding of HDTCM ............................................................... 27
    3.1.1 The representation of state transition .................................... 27
    3.1.2 The construction of the state table ....................................... 31
    3.1.3 The assignment of the transmission symbols ............................... 33
  3.2 The Decoding of HDTCM ............................................................... 37
    3.2.1 Maximum likelihood detection .............................................. 37
    3.2.2 The ML decoding of HDTCM .................................................. 43
    3.2.3 Practical implementation of the ML algorithm ............................ 48

IV Theoretical Performance Analysis of HDTCM ........................................ 50
  4.1 The Properties of HDTCM ............................................................. 50
    4.1.1 Uniformity .............................................................................. 51
    4.1.2 Free Euclidean distance ....................................................... 57
    4.1.3 Minimum Euclidean distance ................................................... 64
    4.1.4 Cyclic properties .................................................................... 76
  4.2 Probability of Codeword Error $P_{ce}$ .......................................... 87
    4.2.1 $P_{ce}$ of two-codeword decoding ........................................... 87
    4.2.2 $P_{ce}$ of multi-codeword decoding .......................................... 89
    4.2.3 Lower bound for probability of codeword error ........................... 91
  4.3 Relationship between $P_{ce}$, $P_{se}$ and $P_{be}$ ............................. 94
    4.3.1 Pairwise probability of symbol error ....................................... 94
    4.3.2 Pairwise probability of bit error ............................................ 96
  4.4 $E_b/N_0$ and $\sigma_n$ ...................................................................... 97
    4.4.1 The relationship between $E_c$, $E_s$ and $E_b$ ............................ 97
    4.4.2 $\sigma_n$ in terms of $E_b/N_0$ .................................................. 98
  4.5 The Upper Bounds and Lower Bounds of $P_{se}$ and $P_{be}$ .................. 99
    4.5.1 Upper bounds ........................................................................... 100
    4.5.2 Lower bounds .......................................................................... 100
  4.6 Asymptotic Upper Bound ................................................................. 103
    4.6.1 Asymptotic performance at high SNR ....................................... 103
    4.6.2 On good code criteria ............................................................ 109
  4.7 Coding Gain of the HDTCM Scheme ................................................ 118
    4.7.1 The comparison of four spread spectrum systems ........................ 118
    4.7.2 The coding gain of HDTCM ...................................................... 124
  4.8 Trade-offs ....................................................................................... 135
    4.8.1 Trellis structure — with or without parallel transitions .............. 135
    4.8.2 The selection of $B$ .................................................................. 136
4.8.3 Orthogonal signal constellation ...................... 137

V Efficient Decoding Algorithms ............................ 140

5.1 Reliability of Hard-decoded Symbol .................... 140
5.2 Reliability of Hard-Decoded Multi-Symbol Sub-Blocks ..... 144
5.3 Efficient Decoding by Using Reliability Information ...... 148
5.4 Trellis Paths of Multi-Symbol Sub-Blocks ................. 152
5.5 Efficient Decoding Algorithms ............................ 155
   5.5.1 Algorithm A: The use of a codebook ................. 156
   5.5.2 Improvement of Algorithm A using multi-symbol sub-blocks 160
   5.5.3 Algorithm B: The use of trellis structure ............ 162

VI Conclusions ............................................. 166

6.1 Conclusions ............................................. 166
6.2 Future Research ........................................ 169

BIBLIOGRAPHY ............................................ 171

APPENDIX

A The Performance of Bi-Orthogonal Signaling ............... 174

ABSTRACT
# List of Tables

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>67</td>
</tr>
<tr>
<td>3</td>
<td>70</td>
</tr>
<tr>
<td>4</td>
<td>71</td>
</tr>
<tr>
<td>5</td>
<td>71</td>
</tr>
<tr>
<td>6</td>
<td>71</td>
</tr>
<tr>
<td>7</td>
<td>73</td>
</tr>
<tr>
<td>8</td>
<td>74</td>
</tr>
<tr>
<td>9</td>
<td>75</td>
</tr>
<tr>
<td>10</td>
<td>76</td>
</tr>
<tr>
<td>11</td>
<td>77</td>
</tr>
<tr>
<td>12</td>
<td>77</td>
</tr>
<tr>
<td>13</td>
<td>79</td>
</tr>
<tr>
<td>14</td>
<td>80</td>
</tr>
</tbody>
</table>

1. The state table of the trellis shown in Figure 16.
2. The weight distribution of Euclidean distances for $n = 2$ with various $N, D$ and $B$.
3. The relationship between $d_{\text{free}}^2$ and $d_{\text{min}}^2$ when $(N, n, D) = (8, 2, 3)$.
4. The relationship between $d_{\text{free}}^2$ and $d_{\text{min}}^2$ when $(N, n, D) = (16, 2, 4)$.
5. The relationship between $d_{\text{free}}^2$ and $d_{\text{min}}^2$ when $(N, n, D) = (32, 2, 5)$.
6. The relationship between $d_{\text{free}}^2$ and $d_{\text{min}}^2$ when $(N, n, D) = (64, 2, 6)$.
7. The weight distribution of Euclidean distances for $n = 2$ when $B \leq D$.
8. The weight distribution of Euclidean distances for $n = 4$ with various $N, D$ and $B$.
9. The relationship between $d_{\text{free}}^2$ and $d_{\text{min}}^2$ when $(N, n, D, B) = (32, 4, 2, 3)$.
10. Categorization of the codebook for $(8, 2, 3, 1)$ scheme.
11. Categorization of the codebook for $(8, 2, 3, 2)$ scheme.
12. Categorization of the codebook for $(8, 2, 3, 3)$ scheme.
13. Categorization of the codebook for $(8, 2, 3, 4)$ scheme.
14. Categorization of the codebook for $(8, 2, 3, 5)$ scheme.
Patterns of source data, trellis paths and codewords when $N = 8$, $n = 2$ and $D = 3$ and their corresponding groups whose elements are the indices of source blocks. ......................... 81

The upper bounds, lower bounds, and asymptotic upper bound of $P_{ee}$, $P_{se}$ and $P_{be}$. .................................................. 111

The coding gains and processing gains of four spread spectrum systems. 124

The asymptotic coding gains of HDTCM spread spectrum systems over MPSK-, MFSK-modulated uncoded conventional spread spectrum systems. .............................. 129

The asymptotic coding gain of HDTCM spread spectrum systems over BPSK or BFSK uncoded conventional spread spectrum systems when $n = 2$ and $B \geq D + 2$ and the processing gains of HDTCM systems. 130

The processing gains of HDTCM systems, BPSK- and BFSK-modulated systems when $n = 2$. ........................................... 132

The processing gains of HDTCM systems, QPSK- and QFSK-modulated systems when $n = 4$. ........................................... 133

The asymptotic coding gain of HDTCM spread spectrum systems over QPSK- or QFSK-modulated uncoded conventional spread spectrum systems when $n = 4$ and $B \geq D + 1$. .............................. 133

Re-arrangement of received sequence. ................................. 140

The error probability of hard symbol decision at various $E_b/N_0$ for (8,2,3,12) scheme. ........................................... 142

The error probability of hard symbol decision at various $E_b/N_0$ for (8,2,3,24) scheme. ........................................... 145

The error probability of hard-decoded two-symbol sub-block at various $E_b/N_0$ for (8,2,3,12) scheme. .............................. 147

The error probability of hard-decoded two-symbol sub-block at various $E_b/N_0$ for (8,2,3,24) scheme. .............................. 148
28 The probability of hard-decoded three-symbol sub-block error at various $E_b/N_0$ for (8,2,3,24) scheme.
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simplest digital communications structure.</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Transmitter block diagram.</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Receiver block diagram.</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>A general block encoder.</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Rate 1/2 convolutional code with $K=2$.</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>Trellis diagram of rate 1/2 convolutional code with $K=2$.</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>A general model for TCM.</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>Three digital communication schemes: (a) uncoded transmission with BPSK; (b) BPSK with a rate 1/2 encoder and bandwidth expansion; (c) QPSK with a rate 1/2 encoder and no bandwidth expansion.</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Two examples of MPSK set: (a) BPSK, (b) QPSK.</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>Set partitioning of a QPSK constellation.</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>Signal assignment for a four-state TCM.</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>Illustration of the mechanism of Viterbi algorithm.</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>The DS spread spectrum systems.</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>The FH spread spectrum systems.</td>
<td>24</td>
</tr>
</tbody>
</table>
The matched filter for $N$-ary decision problem.

A trellis diagram and its disintegration according to inputs.

By fixing input, the state transitions can be separated into two permutation groups with different cycles.

The partitioning of transmission symbols.

An example of the partitioning of transmission symbols when $N=8, n=2$.

The maximum likelihood criterion.

The maximum inner product criterion.

The minimum Euclidean distance criterion.

The development of the optimum criteria.

The HDTCM block encoding.

The encoding of linear algebraic codes.

The construction of codebook.

The representation of bi-orthogonal signaling.

An imagined model of HDTCM encoder.

(A) The “natural” labeling of constellation for $N=8, n=2$; (B) an isometric labeling of (A).

(a) The transmission symbol table; (b) the transmission symbol table of the isometric scheme.

The trellis diagram when $N=2, n=2, D=2$.

Trellis diagram when $N=2, n=2, D=2, B=4$. 
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>Trellis diagram when $N = 2, n = 2, D = 2, B = 5$.</td>
</tr>
<tr>
<td>34</td>
<td>Trellis diagram when $N = 2, n = 2, D = 2, B = 7$.</td>
</tr>
<tr>
<td>35</td>
<td>(a) A trellis with parallel transition; (b) its minimum-length error event.</td>
</tr>
<tr>
<td>36</td>
<td>(a) A trellis without parallel transition; (b) its minimum-length error event.</td>
</tr>
<tr>
<td>37</td>
<td>The weight distribution of Euclidean distance when $N = 8, n = 2, D = 3$.</td>
</tr>
<tr>
<td>38</td>
<td>The weight distribution of Euclidean distance when $N = 16, n = 2, D = 4$.</td>
</tr>
<tr>
<td>39</td>
<td>The transmission symbol assignment of an arbitrary state when $n = 4$.</td>
</tr>
<tr>
<td>40</td>
<td>A general encoder model.</td>
</tr>
<tr>
<td>41</td>
<td>A simple illustration of the relationship between source, trellis path and codeword.</td>
</tr>
<tr>
<td>42</td>
<td>The generation of $i$th group.</td>
</tr>
<tr>
<td>43</td>
<td>The grouping of (8,2,3,5) scheme.</td>
</tr>
<tr>
<td>44</td>
<td>One-to-one correspondence between information matrix and codeword matrix.</td>
</tr>
<tr>
<td>45</td>
<td>The upper and lower bounds of $P_{ce}$ of (8,2,3,5) scheme.</td>
</tr>
<tr>
<td>46</td>
<td>The upper and lower bounds of $P_{be}$ of (8,2,3,5) scheme.</td>
</tr>
<tr>
<td>47</td>
<td>The union and asymptotic upper bounds of $P_{ce}$ of (8,2,3,5) scheme.</td>
</tr>
<tr>
<td>48</td>
<td>The union and asymptotic upper bounds of $P_{be}$ of (8,2,3,5) scheme.</td>
</tr>
<tr>
<td>49</td>
<td>The union bound, lower bound and asymptotic upper bound of $P_{be}$ of (256,2,8,10) scheme.</td>
</tr>
<tr>
<td>50</td>
<td>The union bound, lower bound and asymptotic upper bound of $P_{be}$ of (256,2,8,11) scheme.</td>
</tr>
</tbody>
</table>
The union bound, lower bound and asymptotic upper bound of \( P_{be} \) of (256,2,8,12) scheme.

The union bound, lower bound and asymptotic upper bound of \( P_{be} \) of (256,2,8,13) scheme.

The percentage contribution to the union bound of \( P_{be} \) due to each distinct Euclidean distance of (16,2,4,7) scheme.

The percentage contribution to the union bound of \( P_{be} \) due to each distinct Euclidean distance of (16,2,4,8) scheme.

The percentage contribution to the union bound of \( P_{be} \) due to each distinct Euclidean distance of (16,2,4,9) scheme.

The percentage contribution to the union bound of \( P_{be} \) due to each distinct Euclidean distance of (16,2,4,10) scheme.

The simulation model for HDTCM schemes.

The simulation result of \( P_{be} \) of (16,2,4,B) schemes with \( B = 6, 7, 8, 9, 10 \) respectively.

The comparison of system’s performance: (A) an uncoded conventional spread spectrum communication system; (B) a coded conventional spread spectrum communication system; (C) a combined coding and modulation — TCM scheme; (D) a high dimensional trellis-coded modulation spread spectrum communication system.

The definition of coding gain.

(a) An uncoded MPSK spread spectrum system; (b) an uncoded MFSK spread spectrum system; (c) an HDTCM spread spectrum system.

The asymptotic coding gain \( \gamma_{asym} \) of HDTCM when \( n = 2 \).

\( \gamma_{asym_{BPSK}} \) vs. \( \sigma^2_{min} \).

The asymptotic coding gain \( \gamma_{asym} \) of HDTCM when \( n = 4 \).
65 The error probability of symbol hard decision at various reliability when $B = 12$. .......... 143

66 The error probability of symbol hard decision at various reliability when $B = 24$. .......... 146

67 The error probability of two-symbol sub-block hard decision at various reliability when $B = 12$. .......... 147

68 The error probability of two-symbol sub-block hard decision at various reliability when $B = 24$. .......... 149

69 The error probability of three-symbol sub-block hard decision at various reliability when $B = 24$. .......... 150

70 A sub-block of length two. ................. 153

71 A sub-block of length four. .................. 153

72 A sub-block of length five. .................. 154

73 A suboptimum decoding algorithm that reduces the size of codeword candidates by using codebook — Algorithm A. ............ 157

74 The illustration of template index matrices: POSITIVE and NEGATIVE. ....................... 158

75 The error performance of Algorithm A. ................. 159

76 The segmentation of trellis paths by two anchors: (a) both are end blocks; (b) both are in the middle; (c) one of them is the middle and the other is an end block. ................. 163

77 The segmentation of a trellis path by two anchor sub-blocks. ........ 164

78 Probability of first kind word error ................. 177

79 Probability of second kind word error ................. 177
CHAPTER I

Introduction

1.1 History

Coding and modulation are two indispensable parts in modern digital communications systems. In conventional digital communications systems, coding and modulation are two separate, sequential processes. The coding gain is achieved through bandwidth expansion. The trade-off between bandwidth and power is a main concern in the design of a digital communications system. Because of bandwidth expansion, coding has not been popular for the bandwidth-limited channels, such as telephone channels, until the introduction of trellis-coded modulation (TCM).

In 1982, Ungerboeck proposed a TCM scheme combining coding and modulation as one single process so that coding gain can be achieved without bandwidth expansion [2, 3]. It combines a multilevel/phase modulation signaling set with a state-oriented trellis coding scheme, which is usually a convolutional code. TCM achieves coding gain at the price of decoder complexity. TCM has found wide applications in band-limited channels, such as telephone channels [4, 5, 6]. The success of V.34+ standard for transmission rate up to 33.6 Kbps over dial and 2-wire leased lines is just one example. The transmission speed is pushed close to the Shannon limit of
the telephone lines.

The Viterbi algorithm is used for the asymptotically optimum decoding of TCM [7, 8, 9, 10]. It requires less computation than that of the brute-force maximum likelihood approach. However, its computational complexity and storage requirements are proportional to the length of the transmitted sequence because of its mechanism. In practical applications, decisions are forced to be made before the entire sequence has been observed to avoid delay and reduce memory requirement [11]. The price for this is optimality.

Maximum-likelihood (ML) and computationally efficient soft-decision decoding for linear block code have been studied extensively. The abundance of decoding schemes, such as [12, 13, 14, 15, 16] reflects the difficulty in reducing the decoding complexity. These algorithms aim at reducing the decoding complexity by exploiting the structure and the order of the reliability information of the code to reduce the search space while making their performance as close to the maximum likelihood performance as possible.

### 1.2 High Dimensional Trellis-Coded Modulation

In a spread spectrum system, the transmission bandwidth employed is much greater than the minimum bandwidth required to transmit the information [17]. This gives spread spectrum communications systems many advantages over conventional communication systems. Because of bandwidth expansion, a spread spectrum communications system can be regarded as a power-limited system, rather than a bandwidth-limited system. In current spread spectrum communications, coding and modulation
are still two separate, sequential processes. Although TCM has found great success in bandwidth-limited channels, modifications are needed to implement TCM in spread spectrum communications. Efforts of merging two processes as an entity have been seen [18, 19]. However, the signal constellation used is still bandwidth-efficient. The bandwidth expansion is still accomplished by PN spreading. Moreover, PN spreading is rather independent of modulation.

A new trellis coded-modulation scheme, high dimensional trellis-coded modulation (HDTCM), that allows direct bandwidth expansion has been developed for spread spectrum communications [20, 21, 22]. The proposed scheme integrates a state-permuted block code and an expanded high dimensional signal constellation. This dissertation chooses bi-orthogonal signaling set as the high dimensional signal constellation. The state-permuted trellis structure is characterized by a state table and a state constraint, i.e., the starting state of a legal trellis path must be the same as the ending state. By “block codes” here we do not mean linear algebraic codes, such as Reed Solomon codes. The assignment of channel symbols during encoding is characterized by a transmission symbol table. Lo’s Ph.D. dissertation [21] has concentrated on the design of the state table and the transmission symbol table.

The goal of this dissertation is to theoretically analyze the performance of this proposed HDTCM scheme and to investigate its decoding algorithms [23].

1.3 Contributions

Although HDTCM is a new kind of trellis-coded modulation, it is quite different from current TCM in nature. One of the main contributions of this research is
identifying those differences. Because of these differences, the performance analysis and decoding algorithm can not be readily borrowed from that of current TCM. By starting from scratch, a self-established and systematic performance analysis method is developed and thus the advantages of this HDTCM scheme are able to be unveiled. An important insight to the scheme is that HDTCM integrates not only coding and modulation, but also PN spreading as one single process in spread spectrum communications. Failing to see that, we would have trouble in analyzing the system’s coding gain and distinguishing it from current TCM.

Other main contributions include the examination of HDTCM properties and the theoretical performance analysis of HDTCM system, which are the backbones of this dissertation. The dissertation proves the uniformity of the HDTCM scheme, analyzes the Euclidean distance and also studies the cyclic properties. In the performance analysis, theoretical expressions of error probabilities are given. Optimum selection of four design parameters is discussed. Also, coding gain is analyzed thus the advantages of HDTCM scheme are unveiled. Moreover, the result of the theoretical analyses provides feedback for the search for an optimum transmission symbol table. It can also serve as a benchmark for future research.

Finally, it is pointed out that the Viterbi algorithm can no longer be implemented in the decoding of HDTCM. An optimum maximum likelihood decoder is thus developed. Efficient maximum likelihood soft decision decoding algorithms are also investigated.
1.4 Overview of the Dissertation

The rest of the dissertation is organized as follows:

Chapter 2  Necessary background knowledge is provided to better understand this dissertation. First, a general picture of conventional coding and modulation is depicted. Then current trellis-coded modulation scheme is detailed, such as its difference from conventional coding and modulation, Ungerboeck's mapping by set partitioning method, and the working mechanism of Viterbi algorithm. Three kinds of spread spectrum communications systems are also briefly reviewed. Finally, the bi-orthogonal signaling and its $N$-ary detection algorithm are described.

Chapter 3  The encoding of HDTCM is reviewed, including the design of a state-permuted state table and a transmission symbol table. Next, an optimum maximum likelihood sequence decoder for HDTCM scheme is presented. The practical implementation of the algorithm based on the characteristics of channel symbols is considered thereafter.

Chapter 4  This chapter is devoted to the performance analysis of HDTCM scheme. The performance analysis can be separated into three parts: the investigation of the properties of the HDTCM scheme, the derivation of error probabilities, and the coding gain. First, it is shown that the HDTCM scheme with bi-orthogonal signal constellation is uniform. Then the weight distribution of the minimum Euclidean distance is elaborated upon. The cyclic properties of HDTCM is also discussed. Next, the theoretical performance of the HDTCM
scheme is analyzed by giving analytical expressions for lower bounds and upper bounds of the error probabilities. The calculation of probability of bit error $P_{be}$ is presented by first calculating probability of codeword error $P_{ce}$ and then establishing the relationship between $P_{be}$ and $P_{ce}$. Next, the asymptotic performance of the scheme at high signal-to-noise ratios is discussed. Finally, the asymptotic coding gain of the HDTCM scheme is given.

Chapter 5 Reliability of hard-decoded symbols/sub-blocks is discussed. The concept of an anchor is then introduced. Finally, two algorithms aiming at reducing the number of codeword candidates are presented.

Chapter 6 The work of this dissertation is summarized and conclusions are drawn. Further research in some specific topics is also recommended.

Appendix The performance of bi-orthogonal signaling is analyzed.
CHAPTER II

Background

2.1 Conventional Digital Communication Systems

The simplest digital communications structure consists of a transmitter, a channel and a receiver (Figure 1).

![Simplest digital communications structure](image)

Figure 1: Simplest digital communications structure.

The subdivision of transmitter and receiver is shown in Figure 2 and Figure 3 respectively. The receiver is to recover the source symbols from noise-corrupted received signal by inverting the operation of the transmitter. The channels can be categorized as bandwidth-limited channels or power-limited channels [17].

![Transmitter block diagram](image)

Figure 2: Transmitter block diagram.
The role of source encoder is to randomize the source so as to provide the maximum amount of information by stripping off as much redundancy as possible, while the goal of channel encoder is to introduce an error correction capability, or a controlled redundancy, into the source encoder output to combat the channel noise [24]. The source coding is beyond the scope of this dissertation. It is assumed that the output of the source encoder is inputed to the channel encoder.

2.2 Conventional Coding and Modulation

Usually, coding and modulation were considered as two separate, sequential processes of a digital communication system. The modulator and demodulator are devised to convert an analog waveform channel to a discrete channel and the encoder and decoder are designed to correct errors which occurred in the discrete channel.

2.2.1 Modulation and demodulation

The function of modulator is to match the encoder output to the transmission channel. It accepts binary or $M$-ary encoded symbols and produces waveforms appropriate to the physical transmission medium, which is always analog.
The modulation technique is fixed, but change in the method of demodulation are feasible. The method can be categorized into hard-decision and soft-decision. A hard-decision demodulator makes a definite decision for each received symbol, that is, 0 or 1 for binary transmission or one of $0, 1, \ldots, M - 1$ for $M$-ary transmission. However, all real transmission channels are analog channels that deliver waveforms that vary continuously over some range. The quantization during demodulation causes a loss of information. In order to preserve information, a soft-decision demodulator is used, in which the quantization level $Q > 2$ for binary transmission, and $Q > M$ for $M$-ary transmission. If $Q \to \infty$, no quantization is performed.

If the decoder operates on the hard decisions made by the demodulator in a discrete memoryless channel, such decoding is called hard-decision decoding. On the other hand, if a decoder operates on the soft decisions made by the demodulator over a Gaussian channel, the decoding is called soft-decision decoding [17].

2.2.2 Error correcting codes

Error correcting codes may be viewed as mappings from the space of discrete-alphabet input sequences, called messages, to the space of discrete-alphabet output sequences, called codewords. They fall into two broad categories: block codes and trellis codes [11].

Block codes

Block codes operate on a fixed-length block of source messages. The source message stream is divided into blocks, and the encoder/decoder operates independently on
each block. The encoder transforms each $k$-bit data block into a larger block of $n$ bits. The $r = n - k$ redundant bits carry no new information. The code may be viewed as a dictionary of codewords addressed by input messages. The receiver then operates on the $n$-bit received block and attempts to determine the original $k$-bit source block. The ratio $k/n$ is called the code rate. A general block encoder is shown in Figure 4.

![Figure 4: A general block encoder.](image)

**Trellis codes**

The trellis encoder, on the other hand, can be viewed as mapping an arbitrarily long input message sequence to an arbitrarily long code stream without block structure. The encoder is essentially a finite-state machine, which defines its output code symbol at a certain time to depend on the state of the encoder, as well as on current inputs. The encoder produces output bits at a rate of $n$ for each $k$ input bits. The code rate is still $k/n$. The encoder can be characterized by a state diagram, a tree diagram, or a trellis diagram [17]. The trellis diagram, which is a regular, directed finite-state graph reminiscent of a garden trellis, is the most popular.
Convolutional codes [17, 25, 26, 27, 28] are linear trellis codes [11, 29]. A convolutional code is described by three integers, \( n, k, K \), where the ratio \( k/n \) has the same code rate significance that it has for block codes; however, \( n \) does not define a block or codeword length as it does for block coded. The integer \( K \) represents the number of \( k \)-tuple stages in the encoding shift registers and is known as the constraint length. An important characteristic of convolutional codes, different from block codes, is that the encoder has memory — the \( n \)-tuple emitted by the convolutional encoder is not only a function of an input \( k \)-tuple, but is also a function of the previous \( K - 1 \) input \( k \)-tuples.

A rate 1/2 convolutional code generator with \( K = 2 \) is shown in Figure 5. Its trellis diagram is shown in Figure 6 for an input sequence of 101100. A solid line is used to specify “0” input, and a dotted line to specify “1” input. The output sequence is 11 01 00 10 10 11 and the trellis path is shown in a thick solid line.

![Rate 1/2 convolutional code with \( K=2 \).](image)

The use of error control coding improves the performance of a digital communication system at the expense of bandwidth expansion by an amount equal to the reciprocal of the code rate [25, 26]. Higher improvement in performance is achieved
Figure 6: Trellis diagram of rate 1/2 convolutional code with $K=2$.

by lowering the code rate to increase the redundancy in the code at the cost of decoding complexity and bandwidth expansion. For power-limited channels, like deep space communications channels, one may trade the bandwidth expansion for transmitted power in order to achieve a desired system performance. However, because of bandwidth expansion, coding has not been popular for the bandwidth-limited channels, such as telephone channels, until the introduction of *trellis-coded modulation* (TCM) [2, 3, 4, 5, 6].

### 2.3 Trellis-Coded Modulation

TCM, a technique that combines coding and modulation as one entity was developed to achieve coding gain without the sacrifice of bandwidth expansion. It provides a greatly increased latitude for overall optimization of the design of coding and modulation. One of the most prominent applications so far is in modem design. A TCM scheme with an asymptotic coding gain of 4 dB was adopted by CCITT for use in 9600 bits/s two-wire full-duplex voiceband modem in 1984 [7, 30]. It is still part of
the most recent ITU-TSS\textsuperscript{1} V.34+ standard for transmission rate up to 33.6 Kbps over dial and 2-wire leased line\textsuperscript{2}. Today, when bandwidth is more precious than ever, TCM is playing a greater role in increasing the system’s throughput.

2.3.1 TCM vs. conventional coding and modulation

The innovative aspect of TCM is the concept that coding and modulation should not be treated as separate entities, but rather, as a single unique operation [29, 31]. The received signal, instead of being first demodulated and then decoded, is processed by a receiver that combines demodulation and decoding in a single step. To avoid bandwidth expansion, one may integrate a code, say a convolutional code, with an expanded bandwidth-efficient signal set, such as MPSK or QAM, and exploit the redundancy resulting from such expansion. A general model of trellis-coded modulation can be seen in Figure 7.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure7}
\caption{A general model for TCM.}
\end{figure}

\textsuperscript{1}International Telecommunication Union – Telecommunications Standards Sector, formerly CCITT.

\textsuperscript{2}For reference, please check the ITU homepage: www.itu.ch
Assume that the signal $x_n$ transmitted at discrete time $n$ depend not only on the current source symbol $a_n$, but also on a finite number of previous source symbols:

$$x_n = f(a_n, a_{n-1}, \ldots, a_{n-L}). \quad (2.1)$$

By defining

$$\sigma_n = (a_{n-1}, \ldots, a_{n-L}) \quad (2.2)$$

as the state of the encoder at time $n$, then the finite state machine can be summarized as

$$x_n = f(a_n, \sigma_n) \quad (2.3)$$

$$\sigma_{n+1} = g(a_n, \sigma_n) \quad (2.4)$$

where the function $f(\cdot, \cdot)$ shows the mapping of channel symbols and $g(\cdot, \cdot)$ shows the evolution of modulator states. The current channel symbol $x_n$ and next state $\sigma_{n+1}$ depend on the current input $a_n$ and current state $\sigma_n$.

In conventional coding applications, redundancy means that the set of allowable codewords is smaller than the size of possible codewords. Thus the code symbols do not carry as much information per symbol as they might without coding, and we speak of the transmission as being redundant. The code rate is decreased and the bandwidth is expanded. However, in TCM, coding is used to increase the Euclidean distances between allowed symbols rather than perform error correction. Redundancy is due to the increased signal alphabet in the modulation constellation to accommodate the added code bits and the inter-dependency among the signals. These two, along with set partitioning which is described in the sequel, are three basic ideas underlying trellis-coded modulation. The difference between TCM and conventional error-control coding can be summarized in three ways [32, pp. 3-12]:
1. Coding is used to make transmission errors less likely, rather than to detect/correct errors,

2. Encoders and modulators are jointly designed to achieve larger distance in the Euclidean signal space,

3. Redundancy for coding is provided by signal set expansion, thus avoiding the code rate loss or bandwidth expansion typical for traditional error-control coding.

Consider a simple situation where it is desired to transmit a data stream through a channel one information bit every $T$ seconds. One possible solution is to use an uncoded BPSK with throughput of $1/T$ bits/s/Hz, as shown in Figure 8(a).

If a rate 1/2 convolutional code and BPSK modulation are used, the bandwidth will be doubled with respect to the uncoded scheme, which is shown in Figure 8(b). Figure 8(c) shows another solution. We may consider a rate 1/2 convolutional code combined with QPSK, which has throughput of $2/T$ bits/s/Hz. This coded QPSK scheme yields the same throughput as uncoded BPSK, that is, $1/T$ bits/s/Hz. Each signal still carries one information bit and the reduction of the signal duration is avoided. Both BPSK and QPSK schemes require the same bandwidth, hence no bandwidth expansion is incurred. If the coded scheme performs superior to the uncoded one at the same SNR, we can claim that such an improvement is achieved without sacrificing either data rate or bandwidth expansion.

Although the expansion of a signaling set provides the redundancy required for coding, it shrinks the distance between the signal points if the average signal energy is kept constant (Figure 9). The reduction in the distance between signal points...
Figure 8: Three digital communication schemes: (a) uncoded transmission with BPSK; (b) BPSK with a rate 1/2 encoder and bandwidth expansion; (c) QPSK with a rate 1/2 encoder and no bandwidth expansion.

Figure 9: Two examples of MPSK set: (a) BPSK, (b) QPSK.
increases the probability of error which is compensated with coding, if the coding scheme is to provide a benefit. The coding gain is achieved with the price of neither bandwidth expansion nor power increment. However, it is at the expense of increased complexity of decoder.

2.3.2 Ungerboeck’s mapping by set partitioning

Because of the limitations in symbol transitions introduced by trellis coding, error performance is no longer determined by closest neighbors in the signaling set. Rather, minimum Euclidean distances between members of allowable symbol sequences, or the free Euclidean distance, will determine error performance. Proper coding will ensure that the free Euclidean distance will be greater than the minimum Euclidean distance in the uncoded signal constellation.

Let $d(a_n, b_n)$ denote the Euclidean distance between two symbols $a_n$ and $b_n$ at time $n$. The coder is then designed to maximize the free Euclidean distance

$$d_{\text{free}} = \min \left[ \sum_n d^2(a_n, b_n) \right]^{1/2}, \quad \forall \{a_n\} \neq \{b_n\}. \tag{2.5}$$

where $\{a_n\}, \{b_n\}$ are two sequences of symbols. This optimization is ensured by a method called mapping by set partitioning [3, 7]. The method first successively partitions an $M$-ary constellation into 2, 4, 8, ... subsets with size $M/2, M/4, M/8, \ldots$, having progressively larger minimum distances $d_{\text{min}}^1, d_{\text{min}}^2, d_{\text{min}}^3, \ldots$. The signal assignment rule is:

- Members of the same partition are assigned to parallel transitions;

- Members of the next larger partition are assigned to adjacent transitions;
• All the signals are used equally often.

Figure 10 shows the set partitioning of a QPSK constellation appeared in Figure 8 (c). The QPSK constellation \{0, 1, 2, 3\} is divided into two BPSK constellations, \{0, 2\} and \{1, 3\}. The minimum distance in BPSK constellation is \(\sqrt{2}\) times larger than that of QPSK constellation.

![Figure 10: Set partitioning of a QPSK constellation.](image)

Assume the encoder has four states. Each state has two outgoing and incoming branches. There are eight outgoing or incoming branches in each state transition. Each branch is assigned a signal from the provided signaling set. Because there are no parallel transitions in this case, the assignment is shown in Figure 11. The free Euclidean distance is

\[
d_{\text{free}} = \sqrt{d^2(0, 2) + d^2(0, 3) + d^2(0, 2)} = \sqrt{10},
\]

which is \(\sqrt{10}/2\) times larger than the \(d_{\text{min}}\) of the uncoded scheme. The coding gain is

\[
10 \log_{10} \frac{d_{\text{free}}^2}{4} \approx 4\text{dB}.
\]
2.4 Decoding of TCM: The Viterbi Algorithm

The use of hard-decision demodulation prior to decoding in a coded scheme causes an irreversible loss of information which translates into a loss of SNR. Soft decision is used instead to make full use of all information. Soft decoding can provide about 2 dB of additional coding gain when compared to hard-decision decoding.

The direct implementation of the maximum likelihood decoder for convolutional codes requires examining all code sequences or code paths through the trellis and choosing the one with largest likelihood. Since the number of codeword grows exponentially with sequence length, such a brute force implementation would be fairly complex because of the long input sequence. There are several decoding algorithms for convolutional codes, such as sequential decoding and feedback decoding [17, 11], which are suboptimal.

A practical implementation of the maximum likelihood decoding for convolutional codes is the Viterbi algorithm [8]. The Viterbi algorithm was originally proposed as an asymptotically optimum decoding algorithm for convolutional codes and was later
shown by Omura [9] to be a dynamic programming solution to the problem of finding the maximum likelihood solution.

If the maximum likelihood criterion is applied in soft-decision decoding on the AWGN channel, the decision rule of an optimum sequence decoder for convolutional codes will depend on free Euclidean distance. A maximum likelihood path is the one with minimum path metric if path metric is Euclidean distance.

The Viterbi algorithm starts from an agreed-on initial state and attempts to compute the maximum likelihood function for each possible path remerging at a specific state (node). In each state, it compares the path metrics of different paths arriving at the same state. Only the path with the smallest metric, the survivor, is preserved and is stored at each node level together with its path metric.

The Viterbi algorithm can be illustrated by a simple example. Figure 12 shows how the algorithm iteratively finds the path with the minimum path metric through a two-state trellis with its branch metrics already calculated. The decisions are made at those states where two paths merge together. Only survivors are shown at each state transition.

The computational complexity and storage requirements of the Viterbi algorithm are proportional to the length of the transmitted sequence. In fact, the algorithm does not reach a conclusion on the entire symbol sequence until the end of transmission. For most practical applications the convolutional codes are very long or even infinite. Long delay would happen and very large memory would be required for a very long symbol sequence. In practice, in order to solve the delay and the memory problems, decisions are forced to be made before the entire sequence has been observed. The survivor path is truncated to certain length $\sigma_d$. $\sigma_d$ is determined by the constraint
Figure 12: Illustration of the mechanism of Viterbi algorithm.
length $K$. In this case, the decoder at any level $l$ retains only the most recent $\sigma_d$ branches back in the trellis. For each new node level, a decision is made on the nodes up to level $l - \sigma_d$ at level $l$. It has been shown [33] that if the truncation depth $\sigma_d$ is large enough, the degradation in error probability due to these premature decisions becomes irrelevant. The trade-off is between algorithm optimality and the selection of $\sigma_d$. Further discussion of implementation issues of the Viterbi algorithm can be found in [7, 11].

2.5 Spread Spectrum Communications

Although in band-limited channels we want to control the spectrum of the transmitted signals so as to minimize the bandwidth required for transmission, in spread spectrum system transmission bandwidth is employed which is much greater than the minimum bandwidth required to transmit the information [17]. This gives spread spectrum systems many advantages [17, 34] over conventional communication systems, such as:

- Low probability of interception/detection (LPI/LPD);
- Code-division multiple access: multiple signals occupying the same RF bandwidth are allowed to be transmitted simultaneously without interfering with one another;
- High processing gain to resist jamming;
- Effective elimination of multi-path interference, countering inter-symbol interference.
The bandwidth spreading in spread spectrum is accomplished by means of a PN code, which is independent of the data, and a synchronized reception with the PN code at the receiver is used for de-spreading and subsequent data recovery. The most common types of spread spectrum systems are direct sequence (DS) systems and frequency hoping (FH) systems which are shown in Figure 13 and Figure 14 respectively. In DS systems, the PN code is used in conjunction with PSK to shift the phase of the PSK signal pseudo-randomly at a rate that is an integer multiple of the bit rate. When the PN code is used in conjunction with FSK in FH systems, the code is used to select the carrier frequency of the transmitted signal pseudo-randomly.

![DS spread spectrum systems diagram](image)

Figure 13: The DS spread spectrum systems.

A transform domain / cyclic code shift keying (TD/CCSK) method was proposed [35] to combat fading in HF skywave spread spectrum communications. CCSK is a form of spread spectrum in which the signal is intentionally spread in a manner that is similar to DS, but is an $M$-ary orthogonal or near-orthogonal signaling scheme. Such a method can be integrated into the proposed high dimensional trellis-
coded modulation scheme as a post-processing unit right before transmission. The detail is out of the scope of this dissertation.

2.6 Bi-orthogonal signaling

A bi-orthogonal signaling set [36, 37] can be constructed from a set of $N$ orthogonal signals, $\{s_i(t), i = 1, 2, \ldots, N\}$, by augmenting it with the negative of each signal, that is

\[
\left\{ s_1(t), \ldots, s_N(t), -s_1(t), \ldots, -s_N(t) \right\}.
\]
If a set \( \{s_k(t) | k = 1, 2, \ldots, 2N \} \) is used to denote the above signal set, then \( s_k(t) = -s_{k-N}(t) \) if \( k = N + 1, \ldots, 2N \). The correlation coefficient between \( s_i(t) \) and \( s_j(t) \) is

\[
\rho_{ij} = E[s_i(t) s_j^*(t)] = \begin{cases} 
1, & i = j, \\
0, & i \neq j, \ |i - j| \neq N, \\
-1, & i \neq j, \ |i - j| = N.
\end{cases}
\]

where \( i, j = 1, 2, \ldots, 2N \) and * denotes the complex conjugate. The size of the bi-orthogonal signaling set is \( 2N \).

![Figure 15: The matched filter for N-ary decision problem.](image)

The matched filter detection of \( 2N \) bi-orthogonal signals can base on \( N \) orthogonal signals, \( s_1(t), \ldots, s_N(t) \), as shown in Figure 15. The decision is then based upon which matched filter’s output has the largest absolute value and the sign of the output with largest absolute value determines whether \( s_i(t) \) or \( -s_i(t) \) was most likely transmitted [17, 36]. The modulator’s output is quantized into one of \( 2N \) symbols.
However, the decision mentioned is an $N$-ary hard decision. This will cause loss of information. In the proposed scheme, we pass the outputs of $N$ matched filters directly to the decoder for soft-decision decoding. The performance analysis of bi-orthogonal signaling is presented in Appendix A.
CHAPTER III

High Dimensional Trellis-Coded Modulation

In this chapter, the introduction of the proposed trellis-coded modulation scheme, high dimensional trellis-coded modulation (HDTM), which is intended for spread spectrum communications is presented. First, the encoder design is reviewed. Then a maximum likelihood decoder is developed.

3.1 The Encoding of HDTM

3.1.1 The representation of state transition

As we know, a finite-state machine is characterized by Equation (2.3) and Equation (2.4), which describe the mapping of channel symbols and the evolution of finite states, respectively. Graphically, a finite-state machine can be represented by a state diagram, or a one-stage trellis diagram as shown in Figure 16(a). In stead of using equations, a finite-state machine can be characterized by a state table and a transmission symbol table. The state table can be obtained from the connections between states at different time slots. The transmission symbol table can be directly constructed from the channel symbol assignment appeared to the left of the trellis in Figure 16(a). For the given trellis the state table is listed in Table 1.
Figure 16: A trellis diagram and its disintegration according to inputs.

Table 1: The state table of the trellis shown in Figure 16.
For the given trellis, the size of the input alphabet is two, so its state table can be disintegrated into two parts according to its inputs. One for 0 input, the other for 1 input as shown in Figure 16(b). The state transitions are $1 \rightarrow 1; 2 \rightarrow 3, 3 \rightarrow 4, 4 \rightarrow 5, 5 \rightarrow 6, 6 \rightarrow 7, 7 \rightarrow 8, 8 \rightarrow 2$ if the input symbol is zero. If the source is $n$-ary, there are $n$ outgoing and incoming branches for every state. Thus its state table can be separated into $n$ sets in a similar manner.

However, the above expression is very cumbersome. One short cut is to use permutation notation. By fixing the input, the state transition can be separated into two types with different cycles as shown in Figure 17. For 0 input, state 1 always goes back to itself, forming a single-cycle permutation group, that is $\{1\}$. The other states form another ordered permutation group, i.e., $\{2, 3, 4, 5, 6, 7, 8\}$ with a cycle of seven. Similarly, for 1 input, the single-cycle permutation group is $\{2\}$, while the ordered seven-cycle permutation group is $\{1, 5, 8, 3, 7, 6, 4\}$.

Figure 17: By fixing input, the state transitions can be separated into two permutation groups with different cycles.

Like TCM, the high dimensional trellis-coded modulation scheme is also a finite state machine. The high dimensional trellis-coded modulation scheme integrates a blockwise trellis code and an expanded high dimensional signaling set. The blockwise trellis code has a state-permuted structure which is achieved by the designing of a
new kind of state table that satisfies a state constraint. The mapping of channel symbols are accomplished by the transmission symbol table.

The HDTCM scheme can be characterized by the following four parameters:

\[ N = \text{Dimensionality of transmitted signal space}, \]
\[ n = \text{Source alphabet size}, \]
\[ D = \text{Trellis depth}, \]
\[ B = \text{Input source symbol sequence length}, \]

or block length, or codeword length.

Simply, the high dimensional trellis-coded modulation scheme can be described by \((N, n, D, B)\). The number of states of HDTCM trellis is determined by \(S = n^D\).

Unlike current TCM, the source input is processed block by block independently rather than continuously in the HDTCM scheme. Also, no agreed-on initial state is assumed. A state-permuted structure is introduced such that a legal trellis path is the one satisfying a state constraint, i.e., the starting state of the trellis path is equal to the ending state of the path. Such trellis is tail-biting or circular. For all \(S\) possible states, each can be a starting state for certain legal trellis path. One-to-one mapping between an information source block and a legal trellis path can be assured by a properly designed state table. Such a state table is constructed by using Zech’s logarithm table [21, 22]. Then the mapping of channel symbols is accomplished by matching the source block to the legal trellis path. This mapping is also one-to-one correspondent. The following section is a concise review of the design of a new state table.
3.1.2 The construction of the state table

Assume \( u_j \) is an \( n \)-ary source input symbol, \( \alpha_{j-1} \) is a previous state and \( \alpha_j \) is a current state. There are total \( S \) states. Similar to TCM, the state transition in HDTCM can be characterized by the function \( \alpha_j = f(u_j, \alpha_{j-1}) \), where \( u_j \in \{u_0, u_1, \ldots, u_{n-1}\} \) and \( \alpha_j, \alpha_{j-1} \in \{\alpha_0, \alpha_1, \ldots, \alpha_{S-1}\} \). It describes a relationship between the current state \( \alpha_j \) and previous state \( \alpha_{j-1} \) and current input symbol \( u_j \).

Similar to the method shown in Figure 17, we can use a combination of disjoint ordered permutation groups to express a set of state-transitions \( \{\sigma_j\} \) by fixing input symbol \( u_j \). For instance, if \( \sigma_0 = (\alpha_0)(\alpha_1 \alpha_2 \alpha_3) \), this \((1,3)\)-type permutation contains one single-cycle (the transition from state 0 to state 0) and one three-cycle (the transition from state 1 to state 2, then state 3, and then back to state 1 itself) as \( u_0 \) is transmitted.

Based on the above expression of the state-transition set, a new trellis structure can be built by setting up \((1,S-1)\)-type permutation. Consider a finite field \( GF(q) \), a primitive element \( \alpha \) is defined if
\[
\alpha^{q-1} = (\alpha \times \alpha \times \ldots \alpha) = 1
\]
for \( \alpha \in GF(q) \). The elements in \( GF(q) \) can be represented in terms of a power of a primitive element. Here we assume \( \alpha \) is the primitive generator and the order of the finite field is \( S \).

As we know, multiplication in \( GF(q) \) is simply the addition of exponents, but addition is usually cumbersome to obtain. The Zech’s logarithm transforms the

---

\(^1\)The notations here conform to the usage in Galois Field \( GF(q) \), in which the numbering starts from 0 to \( q - 1 \). However, we number transition states from 1 to \( S \) and sources from 1 to \( n \) elsewhere without specific declaration. This is to follow the usage in [21].
addition of field elements into multiplication. The mechanism of Zech’s logarithm is briefly addressed below.

For $GF(q)$ field with primitive generator $\alpha$, we tabulate the integer $j = z(i)$ such that

$$1 + \alpha^j = \alpha^{z(i)} \quad (3.2)$$

where $0 \leq i \leq q - 2$. By tabulating the mapping between $i$ and $j$, a Zech’s logarithm table is obtained. Then for the addition between any two elements $\alpha^a$ and $\alpha^b$, we have

$$\alpha^a + \alpha^b = \alpha^a (1 + \alpha^{(b-a) \mod (q-1)}) = \alpha^a \cdot \alpha^{z(b-a)} = \alpha^{a+z(b-a)} \quad (3.3)$$

and $z(b-a)$ can be obtained from the Zech’s logarithm table. Further detail of the Zech’s logarithm is referred to [21].

A “natural permutation” of ordered $(1,S-1)$-type is given as follows

$$(0)(\alpha \ \alpha^2 \ \alpha^3 \ \ldots \ \alpha^{S-3} \ \alpha^{S-2} \ \alpha^{S-1})$$

where $\alpha^{S-1} = 1$. Such $(1,S-1)$-type permutation achieves the maximum permutation cycle of $S - 1$. It can be expressed algebraically as $(\infty)(1 \ 2 \ \ldots \ S - 3 \ S - 2 \ S - 1)$. A good family of $(1,S-1)$-type permutation can be constructed from the “natural permutation”, denoted by $\sigma_\infty$. The rest of family members can be obtained by adding $\alpha^i$ to $\sigma_\infty$, where $1 \leq i \leq S - 1$, i.e.,

$$\sigma_i = (0 + \alpha^i)(\alpha + \alpha^i \ \alpha^2 + \alpha^i \ \ldots \ \alpha^{S-2} + \alpha^i \ \alpha^{S-1} + \alpha^i) \quad (3.4)$$

By applying the Zech’s logarithm table, the above equation can be easily re-written algebraically as

$$\sigma_i = (i)(z(1) + i \ z(2) + i \ \ldots \ z(S - 3) + i \ z(S - 2) + i \ \infty) \quad (3.5)$$
The existence and uniqueness of $\sigma_i$ is guaranteed [21]. If the input symbol is $n$-ary, there needs total $n$ state-transition sets to construct the trellis. We can choose arbitrarily $n$ members from the family, which consists of $S$ members. The state table can be obtained by converting those transition sets in a way opposite to the process described in Section 3.1.1. The initial state of any arbitrary input block can be solved by an equation [21].

### 3.1.3 The assignment of the transmission symbols

Once an initial state is determined, a legal path of the codeword is established. A channel symbol can then be assigned for each particular state transition based on the incoming information source by referring to the transmission symbol table. Thus, a source block can be mapped to a codeword consisting of a sequence of channel symbols. The mapping between the source and the codeword is one-to-one correspondent because of the fact that any source block has a unique legal path. The channel symbols are selected from a high dimensional signal constellation. In this dissertation, a bi-orthogonal signaling set is chosen. The comparison between the choice of orthogonal constellation and bi-orthogonal one is presented in Section 4.8.

An $N$-dimensional bi-orthogonal signal constellation is represented by the following set

$$\{s_1(t), \ldots, s_N(t), -s_1(t), \ldots, -s_N(t)\}$$

(3.6)

where any element in the set has one antipodal and is orthogonal to the others $2N - 2$ elements. It has a size of $2N$. Symbolically, the set can be simply represented by

$$\{1, -1, 2, -2, \ldots, N, -N\}.$$
The optimum transmission symbol table is the one that is able to maximize the minimum Euclidean distance $d_{\text{min}}$ between any two codewords. In Section 4.1.3, it is shown that $d_{\text{min}}$ is upper bounded by the free Euclidean distance $d_{\text{free}}$. This suggests that the optimum transmission symbol table should maximize $d_{\text{min}}$ such that $d_{\text{min}} = d_{\text{free}}$. However, it has not yet been proven that this can be achieved in the general case. Section 4.1.3 tells us however, that under certain conditions, the above goal can be realized if the bi-orthogonal constellation is used.

Similar to the current TCM, HDTCM exploits the expansion of the signal constellation so as to achieve additional coding gain. However, their methods are quite different. Such difference will be mainly addressed in Chapter IV and mentioned in other parts of the dissertation too. If we assume the source message is $n$-ary, then the dimensionality of the bi-orthogonal signaling is $N$. To expand the signal constellation, we need to make sure that $2N$ should be always greater than $n$. Otherwise no redundancy can be exploited to achieve the desired coding gain.

In the high dimensional trellis-coded modulation scheme, the parallel transitions are not preferable in the aim of increasing coding gain. This analysis is given in Section 4.1.2 and Section 4.8. Thus a transmission symbol table corresponding to a trellis without parallel transitions can be built by applying the following signal constellation partitioning rules:

- Antipodal signals are assigned to those branches originating from the same state.
- All the signals are used equally often.
For a size $2N$ bi-orthogonal signal constellation, $2N$ transmission symbols are divided into $2N/n$ subsets as

$$\{1, -1, 2, -2, \ldots, n/2, -n/2\};$$
$$\{(n/2 + 1), -(n/2 + 1), (n/2 + 2), -(n/2 + 2), \ldots, n, -n\};$$
$$\vdots$$
$$\{((N - n/2 + 1), -(N - n/2 + 1), (N - n/2 + 2), -(N - n/2 + 2), \ldots, N, -N\}$$

as shown in Figure 18. Then each subset can be assigned to the $n$ outgoing branches of a particular state if the source symbol is $n$-ary. When $n = 2$, there are exactly two branches originating from any state. Thus a pair of antipodal signals can be assigned to the same state. An example of such constellation partitioning is given in Figure 19.
In this special case, the maximum Euclidean distance between the signals of those branches originating from the same state is two. It utilizes the advantages of bi-orthogonal signaling. However, if the source is $n$-ary, then the minimum Euclidean distance between the signals of those branches originating from the same state is reduced to $\sqrt{2}$. This is because of the orthogonality nature of bi-orthogonal signaling.

The detailed discussions of the state table design and the transmission symbol table design are presented in [21].

Figure 19: An example of the partitioning of transmission symbols when $N=8$, $n=2$. 
3.2 The Decoding of HDTCM

In current TCM schemes, the Viterbi algorithm is used for decoding because of its asymptotically optimum performance. The algorithm starts from an agreed-on state, then sequentially moves through the trellis stage by stage to search the most likely path. However, there is no agreed-on state in the proposed high dimensional trellis-coded modulation scheme. Legal paths can have different initial states. Therefore, the Viterbi algorithm can not be used as a decoding algorithm for HDTCM. In this section, we first introduce the maximum likelihood sequence detection algorithm, and its variations in AWGN channels. We then apply the ML sequence decoding algorithm to the decoding of HDTCM.

3.2.1 Maximum likelihood detection

Based on the observation of the received signal $x$, the receiver decides which one of the signal points in the signal space is transmitted. For optimum decoding which results in minimum probability of error, their \textit{a posteriori} probability density functions, $P(C_i|x)$ are compared for all $i$. The signal point is then selected corresponding to the largest of the set of \textit{a posteriori} probabilities. This decision criterion is called \textit{maximum a posteriori probability} (MAP) criterion \cite{38}. For equally likely transmitted signals, the MAP criterion is equivalent to the maximum likelihood (ML) criterion \cite{38}. This ML procedure compares the conditional probabilities of the received signal given each possible transmitted sequence $C_i$, $P(x|C_i)$, over all possible
and chooses the one with the largest $P(x|C_i)$. $P(x|C_i)$ is also called the likelihood of $C_i$.

![Diagram](image)

Figure 20: The maximum likelihood criterion.

When the signal is memoryless, the symbol-by-symbol detector using the above optimum decision rule will result in minimizing the probability of symbol error [7]. On the other hand, when the transmitted signal has memory, i.e., the signals transmitted in successive symbol intervals are interdependent, the optimum detector is a maximum likelihood sequence detector which bases its decision on observation of a sequence of received signals over successive signal intervals [34], rather than an individual symbol. This sequence of interdependent symbols are usually called a codeword. The ML sequence detection rule results in minimizing the probability of codeword error, instead of probability of symbol error. The ML sequence detection rule can be broken into two cases: binary hypothesis test and multi-hypothesis test.
Binary Hypothesis Test

Suppose two $L$-dimensional vectors, $C_1$ and $C_2$, are two equally probable codewords, where

$$C_i = [c_{i1}, \ldots, c_{ij}, \ldots, c_{iL}], \quad i = 1, 2.$$  \hspace{1cm} (3.7)

where $c_{ij}$ is a coordinate in the codeword space. Assume the channel is additive white Gaussian with one-sided noise power spectral density $N_0$.

The received sequence $x$ is $x = C_i + n$, where $x = [x_1, \ldots, x_j, \ldots, x_L]$, and $n = [n_1, \ldots, n_j, \ldots, n_L]$, $j = 1, \ldots, L$. $n_j$ is a zero-mean Gaussian distributed random variable. Thus $x$ is a Gaussian random vector with mean

$$m_x = \begin{cases} C_1, & \text{when } C_1 \text{ is transmitted}, \\ C_2, & \text{when } C_2 \text{ is transmitted}; \end{cases}$$

and variance

$$\sigma^2_x = N_0/2 \cdot I.$$  \hspace{1cm} (3.8)

where $I$ is an $L \times L$ identity matrix. Because the noise components are independent in the AWGN channel, the conditional probability given $C_i$ is transmitted can be expressed as

$$P(x|C_i) = \prod_{j=1}^L P(x_j|c_{ij})$$

$$= \prod_{j=1}^L \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(x_j-c_{ij})^2}{2N_0}}. \hspace{1cm} (3.8)$$

For the ML hypothesis test between codewords $C_1$ and $C_2$, we have

$$P(x|C_1) \geq P(x|C_2). \hspace{1cm} (3.9)$$
By plugging in the expression of \( P(x|C_i) \) into Equation (3.8), we have

\[
\prod_{j=1}^{L} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(x_j-c_{ij})^2}{N_0}} C_i \prod_{j=1}^{L} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(x_j-c_{ij})^2}{N_0}} C_i
\]

By taking the logarithm on both sides and doing some simplification, the following inequality is obtained

\[
\sum_{j=1}^{L} -(x_j - c_{ij})^2 C_i \geq \sum_{j=1}^{L} -(x_j - c_{ij})^2.
\]  

(3.10)

The above equation is equivalent to the following equation if it is rewritten in vector form,

\[-\|x - C_1\|^2 C_i \geq -\|x - C_2\|^2\]

(3.11)

\(\|x - C_1\|^2\) is the squared Euclidean distance between \(x\) and \(C_1\). The above equation tells us that the detector is favors the codeword which is closer to the received signal. Because the decision metrics is the Euclidean distance, it is called the minimum Euclidean distance criterion [38]. The direct implementation of this form of optimum rule is formidable because of the amount of computation involved.

By further canceling those redundant terms, we have a simplified decision rule as follows

\[(C_1 - C_2)^T x + (C_2^T C_2 - C_1^T C_1)/2 C_i \geq 0.\]

(3.12)

Notice that \(C_i^T C_i\) \((i = 1, 2)\) is the average power of the \(i\)th codeword. If codeword \(C_1\) and \(C_2\) have the same average power, then the Equation (3.12) is further reduced to

\[(C_1 - C_2)^T x \geq C_i.\]

(3.13)

As we know, the decision metrics \(C_i^T x\) is the inner product, or correlation, between \(x\) and \(C_i\). Thus a received sequence is decoded into the codeword to which the
maximum inner product is achieved. Therefore, it is termed maximum inner product criterion.

**Multi-hypothesis test**

When the samples are known to come from more than two codewords, say $M$ codewords, the above results for binary hypothesis test can be readily extended to the multiple-hypothesis test. For the maximum likelihood hypothesis test between $M$ equally likely codewords (Figure 20), $C_i$ is detected if the log-likelihood $\log P(x|C_i)$ is the largest. In AWGN channels, if each codeword has the same average power, the ML criterion is reduced to the comparison of $M$ inner products $C_i^T x$:

$$\text{If } q_k(x) = \max C_i^T x, \quad \forall i, \ i = 1, \ldots, M, \text{ then } x \in C_k. \quad (3.14)$$

Therefore, a codeword is favored if it has the maximum inner product to the received signal as shown in Figure 21. The equivalent minimum Euclidean distance criterion for multi-hypothesis test is shown in Figure 22.

**Summary**: The MAP rule is equivalent to the ML rule when the transmitted signals are equally likely. The ML criterion is equivalent to the minimum Euclidean distance criterion if the channel is AWGN. The minimum Euclidean distance criterion can be further simplified to the maximum inner product criterion if the transmitted signals have the same average power. Although both the minimum Euclidean distance criterion and the maximum inner product criterion are equivalent to the ML rule for AWGN channels, the maximum inner product criterion is preferred because it requires the least amount of computation and is easier to implement. However, in the performance analysis, the decision metric usually uses the Euclidean distance.
Choose the one with maximum inner product

Choose the one with minimum Euclidean Distance

Figure 21: The maximum inner product criterion.

Figure 22: The minimum Euclidean distance criterion.
It is simply a matter of convenience when the signal space concept is used. The development of the above criteria is shown in Figure 23. The development is unidirectional.

![Figure 23: The development of the optimum criteria.](image)

### 3.2.2 The ML decoding of HDTCM

![Figure 24: The HDTCM block encoding.](image)

The encoding of the HDTCM scheme is shown in Figure 24. The source message stream is divided into blocks. The HDTCM encoder transforms a block of $n$-ary
information source $u_i$ with block length $B$,

$$u_i = [u_{i1}, u_{i2}, \ldots, u_{iB}], \quad (3.15)$$

into a codeword with the same block length $B$,

$$C_i = [C_{i1}, C_{i2}, \ldots, C_{iB}], \quad (3.16)$$

where $u_{ij}$ ($j = 1, \ldots, B$) is an $n$-ary source symbol in $i$th information block $u_i$ and $C_{ik}$ ($k = 1, \ldots, B$) stands for the $k$th channel symbol in $i$th codeword $C_i$. $C_{ik}$ is represented by

$$C_{ik} = [c_{i[(k-1)N+1]}, \ldots, c_{i(kN)}],$$

where $c_{il}$, $l = (k-1)N + 1, \ldots, kN$ is a coordinate of $C_i$ in the codeword space (see Equation (3.7)). Because each codeword contains the same number of symbols, and each symbol has an equal amount of average power, the average codeword power $C_i^TC_i$ is the same with respect to all $i$. Thus the ML criterion is equivalent to the maximum inner product criterion, which is stated in Equation (3.14). The encoder operates independently on each block, as does the decoder.

In conventional $(n,k)^2$ linear block codes, the input alphabet $D$ and the output alphabet $q$ are usually of the same size [11], but the codeword length $n$ is much larger than the source block length $k$ so as to make the codeword space loosely-packed. The encoding of conventional linear algebraic codes is shown in Figure 25. In practical applications, $k$ usually is large if we desire the code to possess powerful decoding capability. For example, a ten-error-correcting BCH code requires $(n,k)$ be $(127,64)$. Its codebook has a total of $2^{64}$ codewords which is too large to be manageable.

\footnote{$k/n$ is the code rate. $n$ and $k$ used here are for the sake of consistency with tradition. Their meanings are different from their usage in HDTCM.}
Figure 25: The encoding of linear algebraic codes.

However, the input alphabet and output alphabet of the encoder in HDTCM scheme are of different size. The size of the input alphabet is $n$, while the size of the output alphabet is $2N$. Moreover, $2N > n$. This is why spreading happens in HDTCM, although implicitly. The difference between the proposed block codes and linear algebraic codes is that to make the codeword space loosely-packed, the proposed block codes keep codeword length $B$ unchanged while expanding the size of output alphabet; on the other hand, linear algebraic codes usually make no change of the sizes of alphabets while increasing the codeword length $n$.

In the proposed block codes, the change of $B$ affects the weight distribution of Euclidean distance, but after $B$ is greater than certain “threshold” values, the minimum Euclidean distance is not improved at all. An improvement of the minimum Euclidean distance is able to be achieved when $B$ is of moderate values. Simply making $B$ larger can not guarantee the improvement of performance. This is due to the redundancy mechanism of HDTCM scheme. The details of this analysis are presented in Section 4.1.
If \( B \) is moderate, the codeword size \( n^B \) will be manageable. This makes the use of the look-up table (codebook) feasible. The following is the description of the maximum likelihood sequence detection algorithm for the proposed scheme.

First of all, a codebook needs to be constructed. To proceed, two matrices are defined. Each source and codeword block can be written in a vector form as shown in Equation (3.15) and Equation (3.16) respectively. An information matrix can be built by stacking all the possible information blocks row upon row. We denote the matrix as \( IM \):

\[
IM = \begin{bmatrix}
    u_{11} & u_{12} & \cdots & u_{1B} \\
    u_{21} & u_{22} & \cdots & u_{2B} \\
    \vdots & \vdots & \ddots & \vdots \\
    u_{M1} & u_{M2} & \cdots & u_{MB}
\end{bmatrix}
\]

where \( M = n^B \) is the number of all possible combinations of information blocks.

After encoding the information matrix \( IM \) row by row, a corresponding codeword matrix \( CM \) can be obtained:

\[
CM = \begin{bmatrix}
    C_{11} & C_{12} & \cdots & C_{1B} \\
    C_{21} & C_{22} & \cdots & C_{2B} \\
    \vdots & \vdots & \ddots & \vdots \\
    C_{M1} & C_{M2} & \cdots & C_{MB}
\end{bmatrix}
\]

Each row vector in matrix \( CM \) is a codeword. The above process is shown in Figure 26.

By using the codebook method, the encoding process is simply reduced to generating an index for the codebook according to the incoming information source, the selected row is then the corresponding codeword. The computation of searching for legal paths is thus saved.

Assume the sample value has a normalized amplitude, either 1 or -1. We can easily map one of the bi-orthogonal signals to a point in the \( N \)-dimensional Euclidean
Figure 26: The construction of codebook.

Figure 27: The representation of bi-orthogonal signaling.
signal space, each point is represented as \( s_j = (s_{i0}, s_{i1}, \ldots, s_{i(N-1)}) \). Without loss of generality, for example, \( s_1(t) \) can be represented in the \( N \)-dimensional signal space by \([1,0,\ldots,0]_N\), and its antipodal signal \(-s_1(t)\) by \([-1,0,\ldots,0]_N\). The others follow in the same manner. Thus, the set of bi-orthogonal signaling is represented by a signal constellation in the Euclidean signal space. Figure 27 shows a way of depicting the signals as visualization in high-dimensional space is difficult. Each slot represents an axis in the eight-dimensional space.

The size of the information source matrix \( IM \) is \( n^B \times B \). Because the codeword length is \( B \) and each channel symbol is of \( N \)-dimension, then each codeword is of \( L = N \cdot B \) dimensions. Therefore, the size of the codeword matrix \( CM \) is \( n^B \times (N \cdot B) \).

### 3.2.3 Practical implementation of the ML algorithm

Although each codeword is of \( N \cdot B \) dimensions, it contains only \( B \) nonzero elements, either 1 or -1. When \( N \) is large, the codeword vector will become sparse, which means the number of non-zero samples is much smaller than the number of zero elements.

From Equation (3.14) we know that in the ML decoding only nonzero elements contribute to the correlation metrics. However, because of the dot product, zero elements still involve in the computation. It needs \( n^B \times (N \cdot B) \) multiplications and additions in each decision if \( CM \) is a full matrix.

In order to save computation, zero elements should be kept from participating in dot product. Therefore, a sparse matrix is employed by storing only those nonzero elements in the codeword matrix \( CM \), which means only a total of \( B \cdot n^B \) nonzero samples needs to be stored. The matrix \( CM \) is converted to a sparse matrix from
a full matrix by squeezing out those zero elements. Only those indexed non-zero elements will participate in the matrix multiplication.

Let the received row vector $x$ be arranged into a $(N \cdot B) \times 1$ column vector $R$ so that direct matrix multiplication can be performed. Then the ML decoding algorithm shown in Equation (3.14) is to calculate all the inner products and then select the maximum one among them:

$$[\text{max.value, index}] = \text{maximum} \ [CM \ast R]$$ (3.17)

where $\ast$ denotes matrix multiplication, “max.value” refers to the maximum inner product, and “index” refers to the row index of the codeword matrix $CM$, which corresponds to the maximum inner product. Because of one-to-one correspondence between the row vectors of information source matrix $IM$ and codeword matrix $CM$, the detected information message is thus the $index$-th row vector in $IM$.

Therefore, in each block decoding, it requires $B \cdot n^B$ multiplications and additions. The load of computation is independent of the channel symbol dimension $N$. The use of the sparse matrix reduces the computation by $N$ without degrading the performance.
CHAPTER IV

Theoretical Performance Analysis of HDTCM

This chapter is devoted to the performance analysis of the proposed high dimensional trellis-coded modulation scheme. In Section 4.1, the properties of the proposed scheme are explored. Section 4.2 gives the upper bounds and lower bounds of error probabilities. In Section 4.6 the asymptotic upper bound when signal-to-noise ratio is high is discussed. The coding gain of HDTCM is derived in Section 4.7. Finally, Section 4.8 discusses some issues.

4.1 The Properties of HDTCM

The analysis presented in Section 4.2 tells us that the error performance of HDTCM is related to the pairwise Euclidean distance. First, the uniformity of pairwise Euclidean distance is investigated. Because of the proposed scheme processes data blockwise, the minimum Euclidean distance is concerned. However, the code itself has a trellis structure, so the free Euclidean distance needs to be examined as well. Some observations of interesting cyclic properties of this state-permuted structure are also presented.
4.1.1 Uniformity

The set of pairwise squared Euclidean distances \( \{d_{ij}^2\} \) between codeword \( C_i \) and codeword \( C_j \), \( i = 1, \ldots, M; j = 1, \ldots, M; \) can be arranged into an \( M \times M \) matrix \( G \),

\[
G = \begin{bmatrix}
d_{11}^2 & d_{12}^2 & \cdots & d_{1M}^2 \\
d_{21}^2 & d_{22}^2 & \cdots & d_{2M}^2 \\
\vdots & \vdots & \ddots & \vdots \\
d_{M1}^2 & d_{M2}^2 & \cdots & d_{MM}^2
\end{bmatrix}
\]

Because of the commutativity of Euclidean distance, \( d_{ij}^2 = d_{ji}^2 \), the matrix is a symmetric matrix with all diagonal elements being zero. To conduct the performance analysis, the knowledge of all the elements in the upper triangle is needed. This means that \( M - 1 \) codewords need to be chosen as references, because \( M(M - 1)/2 \) pairs of \( d_{ij} \) must be examined.

In order to reduce the complexity of performance analysis, all that is needed is the symmetry that arises when the sum of all the elements in any row (or column) of the pairwise squared Euclidean distance matrix does not depend on the row (or column) itself. Such symmetry corresponds to having all the codewords on an equal footing and allows consideration of a single reference codeword rather than all the pairs of codewords for the computation of error probabilities. Such symmetry is called uniformity.

**Algebraic definition of uniformity**

Assume \( \mathbf{1} \) denotes an \( M \times 1 \) column vector whose elements are all one. Consequently, for any given \( M \times M \) square matrix \( \mathbf{A} \), \( \mathbf{1}^T \mathbf{A} \mathbf{1} \) is the sum of all \( M^2 \) entries of \( \mathbf{A} \). If
1 is an eigenvector of transpose matrix $A^T$, then

$$1^T A = \alpha 1^T$$

holds, where $\alpha$ is the eigenvalue with respect to the eigenvector 1. The sum of its elements in a column of matrix $A$ does not depend on the column order. We call $A$ \textit{column-uniform} [7, pp.106]. Similarly, if 1 is an eigenvector of the square matrix $B$, and

$$B 1 = \beta 1$$

where $\beta$ is the eigenvalue corresponding to the eigenvector 1. The sum of its elements in a row of matrix $B$ does not depend on the row order. In this case $B$ is \textit{row-uniform}. It can easily be proved that the product or the sum of two column- (row-) uniform matrices is itself column- (row-) uniform [7, pp.106]. Also, for an $M \times M$ matrix $A$ that is either row- or column- uniform, we have

$$1^T A 1 = M \alpha.$$ 

**Conditions for uniformity**

Assume codeword $C_i$ can be expressed as a vector of $B$ symbols as

$$C_i = [C_{i1}, \ldots, C_{ik}, \ldots, C_{iB}].$$

The squared Euclidean distance between $C_i$ and $C_j$ is equal to the sum of pairwise squared Euclidean distance between their corresponding symbols,

$$d^2(C_i, C_j) = d^2_{ij} = \sum_{k=1}^{B} d^2(C_{ik}, C_{jk}).$$  \hspace{1cm} (4.1)
If let $d_{ijk}^2 = d^2(C_{ik}, C_{jk})$, Equation (4.1) is rewritten as

$$d_{ij}^2 = \sum_{k=1}^{B} d_{ijk}^2.$$  \hspace{1cm} (4.2)

Let matrix $G = \{d_{ij}\}$, matrix $G_k = \{d_{ijk}\}$, where $i = 1, \ldots, M$; $j = 1, \ldots, M$; $k = 1, \ldots, B$ then

$$G = \sum_{k=1}^{B} G_k.$$  \hspace{1cm} (4.3)

If the sum of two uniform matrices is itself uniform, then the sum of multiple uniform matrices is also itself uniform. Thus if we want to show that $G$ is uniform, we need to show that $G_k$ is uniform for all $k$.

**Uniformity of HDTDM scheme**

As we know, the transmission symbol table guarantees that each symbol in the signaling set is used equally often. Thus in any column of the codeword matrix $CM$, each of $2N$ symbols has the same frequency of appearance of $M/(2N)$. Because of the commutativity property of pairwise Euclidean distance, $G_k$ is symmetric. Thus, examining matrix $G_k$ by columns is equivalent to examining it by rows. In each row (fixing $i$) or column (fixing $j$), there are $M/(2N)$ duplicate terms of $d_{ijk}^2$ in $G_k$ because of the duplicate appearance of each symbol. If a new matrix $H_k$ is constructed in which its elements are orderly selected from $G_k$ such that $C_{ik}$, $C_{jk}$ are a pair of two distinct symbols, then the proof that $G_k$ is uniform is reduced to the proof that $H_k$ is uniform, where $H_k$ is a $(2N) \times (2N)$ pairwise squared Euclidean distance matrix between $2N$ distinct symbols. The order of these $2N$ symbols is arranged according to their orders of first appearance in the columns of $CM$. Each $H_k$ corresponds to
a specific $N$-dimensional symbol constellation. The proof of the uniformity of $G_k$ becomes the proof of the uniformity of its corresponding signal constellation of $H_k$.

Figure 28: An imagined model of HDTCM encoder.

Similar to the Ungerboeck's model for rate $m/(m + 1)$ TCM schemes [7, pp. 100], the HDTCM encoder can be viewed as a combination of a rate of $k/m$ block encoder with state constraint and a memoryless mapper, as shown in Figure 28. The imagined block encoder first accepts $B$ symbols to determine the initial state and then transforms $k = \log_2 n$ bits at a time into a block $s_i$ of $m = \log_2 2N$ binary symbols that are fed to the mapper. The mapper then outputs channel symbols $x_i$. We refer to $m$-tuple $s_i$ as the label of the signal $x_i$ according to the mapping rule. The mapping between $x_i$ and $s_i$ is of one-to-one correspondence.

Assume function $f(\cdot)$ describes a one-to-one correspondence mapping between a label vector set $\Omega$ and a signal constellation $\Lambda$ as

$$s_i \xrightarrow{f(\cdot)} x_i, \quad s_i \in \Omega, \ x_i \in \Lambda.$$ 

Consider the transformation induced in the constellation, $\{f(s)|s \in \Omega, \ f(s) \in \Lambda\}$, by the addition to the label vectors of one and the same vector $\bar{s}$. As $s$ runs in $\Omega$ we
get the one-to-one correspondence mapping

\[ f(s) \rightarrow f(s \oplus \bar{s}), \quad s \in \Omega. \]  \tag{4.4}

A sufficient condition for uniformity of \( H_k \) is that Equation (4.4) be an isometry, that is, one-to-one mapping that preserves the Euclidean distances. To prove that this condition is sufficient for uniformity, observe that because of the isometry, the distance of \( f(s_1) \) from \( f(s_1 \oplus s_2) \) will be the same as the distance of \( f(s_1 \oplus \bar{s}) \) from \( f(s_1 \oplus \bar{s} \oplus s_2) \) for any \( s_1 \) and \( s_2 \), which is what we want to prove.

Figure 29 shows an example of a constellation and its labeling giving rise to a uniform HDTCM scheme. The eight-dimensions are mapped into two-dimensions for the sake of visual illustration. (A) shows the “natural” labeling of a bi-orthogonal constellation with a size of sixteen. In the “natural” labeling, the M.S.B.\(^1\) is a sign bit, 0 for positive signals and 1 for negative signals. Each positive pulse stands for the positive part of an axis, the corresponding negative pulse refers to the negative part of the same axis. Let \( \bar{s} = 0011 \). The mapping

\[ f(s) \rightarrow f(s \oplus 0011) \]

maps the ordered axes \{ 1, 2, 3, 4, 5, 6, 7, 8 \} into another ordered set \{ 4, 3, 2, 1, 8, 7, 6, 5 \}. The mapping can be separated as a reflection\(^2\) along a hyperplane to \{ 8, 7, 6, 5, 4, 3, 2, 1 \} and then a hyper-space rotation\(^3\) to \{ 4, 3, 2, 1, 8, 7, 6, 5 \}.

---

\(^1\) Most significant bit.

\(^2\) It can be characterized by a Householder matrix [39], which is a real unitary or orthogonal matrix. Householder reflections preserve lengths and angles and are isometric transformations.

\(^3\) A hyper-space rotation is equivalent to a circular shift of its ordered axes. Such shift can be characterized algebraically by an elementary rotation matrix [39]. The matrix is orthogonal, so it preserves length and angle. Thus it is an isometric transformation.
Since the combination of two isometries is an isometry itself, the mapping above is an isometry. Then $H_k$ is uniform.

For any pair of $H_k, H_j \ (k \neq j)$, an isometric mapping can be established between their corresponding constellations in a way similar to the above analysis. All that is needed is to find out corresponding $\hat{s}$. Thus if any $H_k$ can be proved to be uniform, all $H_k \ (k = 1, \ldots, B)$ are uniform. Hence, $G$ is uniform. The HDTCM scheme is therefore uniform. For a uniform HDTCM scheme, the performance analysis is irrelevant to the choice of the reference codeword.

The change in the transmission symbol table is reflected in Figure 30. (a) shows the transmission symbol table with “natural” labeling, while (b) shows the isometric
If each transmission symbol table is used to generate a codebook respectively and then pairwise squared Euclidean distance matrix is calculated, it can be found that two pairwise squared Euclidean distance matrices are identical and it matches to our analysis of uniformity.

![Diagram](image)

Figure 30: (a) The transmission symbol table; (b) the transmission symbol table of the isometric scheme.

### 4.1.2 Free Euclidean distance

The error patterns of high dimensional trellis-coded modulation scheme occur when codeword $C_i$ is decoded into another codeword $C_j$. The patterns can be categorized into two cases by their initial states:

1. $C_i$ and $C_j$ have the same initial states,

2. $C_i$ and $C_j$ have different initial states.
Because of the uniformity of the HDTCM scheme, we can choose $C_1$ as reference for the following analysis without loss of generality. In case 1, because $C_1$ and $C_j$ have the same initial states, the analysis of the free Euclidean distance is desired. The analysis is broken into two parts:

- A trellis without parallel transition,

- A trellis with parallel transition.

No parallel transitions

The definition of free Euclidean distance is given by first introducing the concept of trellis depth. The definition of trellis depth can be illustrated by a simple trellis diagram shown in Figure 31. Assume the initial state of $C_1$ is State 1. Let State 1 be the starting point. After one state transition, two diverging branches of State 1 reach to State 1 and 2 in the second time interval respectively. After one more transition, diverging branches from State 1 and State 2 reach to all the states in the third time interval exactly once. At the third transition, each state is visited twice, and so on. The trellis depth $D$ is defined as the number of transitions needed to reach all the states exactly once from an arbitrary starting state. In Figure 31, the trellis depth is $D = 2$.

For general $n$, the number of state $S$ is subject to the constraint that $S = n^D$. At the first transition, $1/n^{D-1}$ of the states are reached; at the second transition, $1/n^{D-2}$ of the states are reached; and so on. At the $D$-th transition, each state is visited exactly once. At the $(D + 1)$-th transition, $n$ branches merge to each state.
The free Euclidean distance is defined as the minimum pairwise Euclidean distance among all possible paths that diverge from and remerge to the same state at a later time. Because the first merging occurs at $(D + 1)$-th transition, $D + 1$ is often called the minimum length of error events. There are $n$ paths with error event length of $D + 1$ that start and end at the same state. For HDTCM, the free Euclidean distance can be obtained by

$$d_{free} = \text{minimum} \ \{d(C_1, C_j)|C_j \in C\}, \quad (4.5)$$

where $C$ is a set of codewords that have the same initial state as that of $C_1$.

For those $C_j$ that have distances of $d_{free}$ to $C_1$, their error patterns are identical if they have the same length of error event. Each same-length error pattern is a shifted version of others as shown in Figure 32, Figure 33 and Figure 34. In these figures, the numbers which appear above each plot are the corresponding codewords.

Figure 31: The trellis diagram when $N = 2$, $n = 2$, $D = 2$. 

Figure 32, Figure 33 and Figure 34: Illustrating the same-length error patterns for different codewords.
of the trellis path. In Figure 32, (1) is chosen as the reference, (2) and (3) are the minimum-length error events of length three. The fourth error event has a length of four. As $B$ increases, the number of different error patterns increases too. This is best manifested in Figure 34.

The definition of $d_{\text{free}}$ tells us that it depends on the trellis depth $D$ if the transmission symbol table is unchanged. Thus keeping $N, n, D$ unchanged, and simply increasing $B$, $d_{\text{free}}$ is not necessarily increased, although the distances are more diversified and there are more error patterns.

![Trellis Diagram](image)

Figure 32: Trellis diagram when $N = 2, n = 2, D = 2, B = 4$.

However, if trellis depth $D$ is increased, the minimum length of error patterns becomes larger. For a properly designed transmission symbol table, $d_{\text{free}}$ can be improved if $D$ is increased. But increasing $D$ will increase the number of states $S$, which makes the decoding scheme more complicated.
Figure 33: Trellis diagram when $N = 2$, $n = 2$, $D = 2$, $B = 5$.

Figure 34: Trellis diagram when $N = 2$, $n = 2$, $D = 2$, $B = 7$. 
**Parallel transitions**

If a trellis allows parallel transitions, then there are two or more branches connecting the same pair of states. Figure 35 (a) shows a two-state trellis with parallel transitions, while Figure 36 (a) is the counterpart with no parallel transitions. From the definition of $d_{\text{free}}$, their minimum-length error events can be determined as shown in Figure 35 (b) and Figure 36 (b), respectively.

![Figure 35](image1.png)  
**Figure 35:** (a) A trellis with parallel transition; (b) its minimum-length error event.

![Figure 36](image2.png)  
**Figure 36:** (a) A trellis without parallel transition; (b) its minimum-length error event.

The error paths are in thick solid lines and the reference paths are in thin dashed lines. The minimum length of an error event without parallel transitions is still $D + 1 = 2$. However in Figure 35 (a), the minimum length of error event is only one. Thus its $d_{\text{free}}$ is equal to the distance between the signals of parallel branches.
We conclude that if a trellis allows parallel transitions, its $d_{\text{free}}$ is independent of parameter $D$. We may say that the definition of trellis depth is irrelevant here, although it is needed to determine the number of state $S$.

In summary, for any trellis structure, $d_{\text{free}}$ can be obtained by choosing the smallest of the distances between signals associated with parallel transitions and the distances associated with a pair of paths in the trellis that originate from a common state and merge into the same state at a later time.
4.1.3 Minimum Euclidean distance

Because of the presence of second kinds of error patterns, minimum Euclidean distance needs to be examined.

Assume \( \{ C_j \} \) is the set of codewords that have the same initial states as that of \( C_1 \), \( \{ C_k \} \) is the set of codewords that have different initial states from that of \( C_1 \). The minimum Euclidean distance is minimum pairwise Euclidean distance over all possible combinations. It is equal to

\[
d_{\text{min}} = \text{minimum} \left( \{d(C_1, C_j)\}, \{d(C_1, C_k)\} \right).
\]

Because \( d_{\text{free}} = \text{minimum} \left\{d(C_1, C_j)\right\} \),

\[
d_{\text{min}} = \text{minimum} \left( d_{\text{free}}, \{d(C_1, C_k)\} \right).
\]

Thus we conclude that

\[
d_{\text{min}} \leq d_{\text{free}} \tag{4.6}
\]

which means that \( d_{\text{min}} \) is upper bounded by \( d_{\text{free}} \). If \( d_{\text{free}} < \text{minimum} \left\{d(C_1, C_k)\right\} \), the bound is tight, if \( d_{\text{free}} > \text{minimum} \left\{d(C_1, C_k)\right\} \), it is loose.

The impact of design parameters

The optimum choice of \( N \) For a \((N, n, D, B)\) HDTCM scheme, there are \( n \) diverging or merging branches at each state. There are a total of \( S \) states, so there are a total of \( S \cdot n = n^{D+1} \) diverging/remerging branches in one state transition.
The size of $N$-dimensional bi-orthogonal signaling set is $2N$. In order to design an optimum transmission symbol table, each branch should be assigned a distinct channel symbol such that no symbol duplication is incurred. This means that no symbol should appear more than once in the transmission symbol table. Then the dimension of the signal constellation $N$ needs to satisfy $N = n^{D+1}/2$.

$N$ can be chosen to be smaller than $n^{D+1}/2$, but it is not an optimum choice, because in this case we need to assign one channel symbol to more than one branch. However, if $N > n^{D+1}/2$, we have more channel symbols than needed to be assigned distinctively to all branches. Because of the nature of bi-orthogonal signaling, we can arbitrarily choose $n^{D+1}/4$ antipodal pairs out of $N$ signals. Therefore, the performance will not be improved simply by increasing $N$. Increasing the dimension $N$ means the expansion of bandwidth. If no improvement can be achieved, simply making $N > n^{D+1}/2$ will waste bandwidth, even though we do want to expand the transmitted bandwidth in spread spectrum communications. We should bear in mind that the choice of $N$ should enhance the coding gain in the mean time. Therefore, the optimum value of $N$ is given by

$$N_{opt} = n^{D+1}/2.$$

Parameter $n$  
Because $n$ is the size of input alphabet, $n$ is usually an integer of power of two. A block of $k = \log_2 n$ bits is processed at a time for each source symbol. If $n = 2$, the information is processed bit by bit. Because there are only two outgoing branches at each state, $s_i$ and its antipodal $-s_i$ can be assigned to those two branches so that the transmission symbol table is optimized. The squared Euclidean distance between these two branches is four, which is the largest possible
squared distance. Table 2 shows the weight distributions of the squared Euclidean
distance by keeping \( n = 2 \) and adjusting \( N, D, B \). \( N \) is assigned the optimum value
given by Equation (4.7), i.e., \( N = 2^D \) if \( n = 2 \). In the table, \( N = 8, n = 2, D = 3, \)
\( B = 7 \) is not listed, because \( B \neq S - 1 \) [21].

From the table, some interesting phenomena observed are stated as follows

1. When \( B = D + 1, \ s^2_{\text{min}} = 2(D + 1) \). However, if \( B = D + 2, \ s^2_{\text{min}} = 2(D + 2) \).
The maximum value of \( s_{\text{min}} \) is achieved for the first time.

2. If \( B = D + 3, \ s^2_{\text{min}} \) stays the same, but \( N_{\text{min}} \) goes down. \( N_{\text{min}} \) reaches its
minimum value.

3. If \( B \geq D + 4, \ s^2_{\text{min}} \) is still unchanged, but \( N_{\text{min}} \) becomes a little bit larger
as \( B \) becomes larger. However, the distribution of Euclidean distance is more
diversified and the average squared Euclidean distance becomes larger. The
weight center shifts to the right.

4. The weight distribution seems to have some regularity as \( B \) changes.

   • If \( B = D + 2 \), the weight \( N_{\text{max}} \) that corresponds to the maximum squared
     Euclidean distance is always \( N_{\text{max}} = B \) and \( N_{\text{min}} = n^B - 1 \).

   • If \( B \geq D + 2 \), \( N_{\text{min}} = B \) and the second weight \( N_2 = B \). And \( N_i = 2N_{i-1} \)
     if \( i \geq 3 \) for some \( i \).

   • The weight corresponding to distance \( 2B \) is always equal to the difference
     between \( n^B \) and the sum of other weights.

   • \( N_{\text{max}} = Bn^{B-(D+2)} \) in most cases.
Table 2: The weight distribution of Euclidean distances for $n = 2$ with various $N$, $D$ and $B$. 

<table>
<thead>
<tr>
<th>$N$</th>
<th>$D$</th>
<th>$B$</th>
<th>$d^2=8$</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
<th>24</th>
<th>26</th>
<th>28</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>11</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td>26</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>45</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>56</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>18</td>
<td>36</td>
<td>304</td>
<td>126</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>5</td>
<td>26</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>57</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td>106</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>207</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>18</td>
<td>403</td>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>20</td>
<td>808</td>
<td>788</td>
<td>150</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>11</td>
<td>22</td>
<td>44</td>
<td>1783</td>
<td>176</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>12</td>
<td>24</td>
<td>48</td>
<td>3807</td>
<td>192</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td>57</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>120</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>231</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>457</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>20</td>
<td>903</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>22</td>
<td>44</td>
<td>1783</td>
<td>176</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>24</td>
<td>48</td>
<td>3807</td>
<td>192</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td>120</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>247</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>484</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>20</td>
<td>963</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>22</td>
<td>1915</td>
<td>88</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>24</td>
<td>48</td>
<td>3807</td>
<td>192</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>26</td>
<td>52</td>
<td>104</td>
<td>7567</td>
<td>416</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td>247</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td>502</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>993</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>1981</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>24</td>
<td>3951</td>
<td>96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>26</td>
<td>52</td>
<td>7879</td>
<td>208</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td>502</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1013</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>2014</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>4023</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>26</td>
<td>8035</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>28</td>
<td>56</td>
<td>16047</td>
<td>224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1013</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>9</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>2036</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>9</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>4059</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>9</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>8113</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>9</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>28</td>
<td>16215</td>
<td>112</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>2036</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>4083</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>8152</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>16299</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• It is difficult to characterize such regularity mathematically. This will have negligible impact on performance.

5. The largest weight corresponds to those codewords that are orthogonal or orthogonal-equivalent. This is attributed to the characteristics of the bi-orthogonal signaling.

6. The scheme 
\[(N, n, D, B) = (2^D_1, 2, D_1, D_1 + 1)\]
has the same weight distribution as that of 
\[(N, n, D, B) = (2^D_2, 2, D_2, D_2 + 2)\] if \(D_1 = D_2 + 1\). For example, \((16, 2, 4, 6)\) and \((32, 2, 5, 6)\). This should suggest that if we increase \(N\) and \(D\) and don’t properly increase \(B\) at the same time, the scheme will not have optimum performance.

Figure 37 and Figure 38 are for the visual illustration of the cases when \((N, D) = (8, 3)\) and \((N, D) = (16, 4)\) respectively.

To explain the first phenomenon of the above observations, we remember that 
\[d_{free} \geq d_{min}\]. Table 3 through Table 6 examine \(d_{min}, d_{free}\), and the trellis paths and codewords of those error events for optimum combinations of \(N\) and \(D\) when \(n = 2\). Without specific declaration, the reference codeword corresponds to the all ones source block indexed by 1. In order to examine the relationship between \(d_{free}^2\) and \(d_{min}^2\), all the error events that have the same initial state, i.e., state 1, as that of the reference codeword are recorded. In the tables, \(L\) denotes the length of an error.

\[\text{Not every symbol of a codeword is orthogonal to its corresponding symbol of the reference codeword, but the sum of these pairwise symbol squared Euclidean distance is equal to } 2B. \text{ It looks like that these two codewords are orthogonal to each other. For example, codeword 1 1 1 1 1 and codeword 1 -1 5 -6 -4. There are three pairs of orthogonal symbols and there is one pair of antipodal symbols. Both codewords contain symbol ‘1’. The distance of these two codewords because of such duplication is compensated by the antipodal pair.}\]

\[\text{Its performance analysis is presented in Appendix A.}\]
Figure 37: The weight distribution of Euclidean distance when $N=8, n=2, D=3$. 

Figure 38: The weight distribution of Euclidean distance when $N=16, n=2, D=4$. 
event. If \( B = D + 1 \), there is only one error event that diverges from state 1 and remerges to the same state at a later time. If \( B = D + 2 \), there are two error events with minimum length of \( D + 1 \) and there is one error event with length \( D + 2 \).

Table 3 shows the improvement of \( d_{\text{min}} \), that is making \( d_{\text{min}} = d_{\text{free}} \), as \( B \) is increased from four to five. The error event is -1 5 -6 -4 with minimum length of four. As we have reasoned, \( d_{\text{free}} \) is not changed as \( B \) increases. However, \( d_{\text{min}} \) changes as \( B \) varies. \( d_{\text{min}} \) is equal to \( d_{\text{free}} \) if \( B = D + 2 \). As \( B \) further increases, \( d_{\text{free}} \) stays the same, but the weight distribution is diversified as shown in Table 2. The reason why \( d_{\text{min}} \) cannot be further improved when \( B \) is greater than certain threshold is that \( d_{\text{min}} \) is bounded by \( d_{\text{free}} \) and increasing \( B \) does not result in increasing \( d_{\text{free}} \). This is true for those schemes if \((N, n, D) = (16, 2, 4), (32, 2, 5), (64, 2, 6), \) even for \((128, 2, 7), (256, 2, 8), (512, 2, 9), (1024, 2, 10), \) whose data are not listed because of space.

<table>
<thead>
<tr>
<th>( B )</th>
<th>( d_{\text{free}}^2 )</th>
<th>( d_{\text{min}}^2 )</th>
<th>( d_{ij}^2 )</th>
<th>source</th>
<th>trellis path</th>
<th>codeword</th>
<th>( L )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td>8</td>
<td>1</td>
<td>2122</td>
<td>15641</td>
<td>-15-6-4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>121222</td>
<td>1156411</td>
<td>-1-15-6-4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>21221</td>
<td>156411</td>
<td>-15-6-4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>22212</td>
<td>158341</td>
<td>-1-15-8-3-4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The relationship between \( d_{\text{free}}^2 \) and \( d_{\text{min}}^2 \) when \((N, n, D) = (8, 2, 3)\).

Similar to the criteria of selecting good convolutional codes [17, pp. 347], the criterion of determining the optimum block length \( B \) is based on the code’s minimum Euclidean distance. The first criterion is to select a code that has maximum value of minimum Euclidean distance \( d_{\text{min}} \) for given trellis complexity (number of states) and given transmitted bandwidth, i.e., \( D \) and \( N \) are given. Then the number of
Table 4: The relationship between $d_{free}^2$ and $d_{min}^2$ when $(N, n, D) = (16, 2, 4)$.

<table>
<thead>
<tr>
<th>$B$</th>
<th>$d_{free}^2$</th>
<th>$d_{min}^2$</th>
<th>$d_{ij}^2$</th>
<th>source</th>
<th>trellis path</th>
<th>codeword</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>2 1 1 2 2</td>
<td>1 6 7 8 5 1</td>
<td>-1 6 7 -8 -5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>1 2 1 1 2 2</td>
<td>1 6 7 8 5 1</td>
<td>-1 6 7 -8 -5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>2 1 1 2 2 1</td>
<td>1 6 7 8 5 1</td>
<td>-1 6 7 -8 -5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>2 2 1 2 1 2</td>
<td>1 6 10 11 4 5 1</td>
<td>-1 6 10 -11 4 -5</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 5: The relationship between $d_{free}^2$ and $d_{min}^2$ when $(N, n, D) = (32, 2, 5)$.

<table>
<thead>
<tr>
<th>$B$</th>
<th>$d_{free}^2$</th>
<th>$d_{min}^2$</th>
<th>$d_{ij}^2$</th>
<th>source</th>
<th>trellis path</th>
<th>codeword</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>14</td>
<td>12</td>
<td>12</td>
<td>2 1 1 2 1 2</td>
<td>1 5 6 4 1</td>
<td>-1 5 -6 -4</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>1 2 1 1 2 2</td>
<td>1 2 21 22 18 19 1</td>
<td>-1 20 21 -22 18 -19</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td>2 1 1 2 1 2</td>
<td>1 2 21 22 18 19 1</td>
<td>-1 20 21 -22 18 -19</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>2 2 1 2 2 2 2</td>
<td>1 2 20 7 8 28 5 19 1</td>
<td>-1 20 7 -28 -25 -19</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 6: The relationship between $d_{free}^2$ and $d_{min}^2$ when $(N, n, D) = (64, 2, 6)$.

<table>
<thead>
<tr>
<th>$B$</th>
<th>$d_{free}^2$</th>
<th>$d_{min}^2$</th>
<th>$d_{ij}^2$</th>
<th>source</th>
<th>trellis path</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>16</td>
<td>14</td>
<td>14</td>
<td>2 1 1 1 1 2 2</td>
<td>1 8 9 10 11 12 7 1</td>
<td>-1 8 9 10 11 -12 -7</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>1 2 1 1 1 1 2 2 2</td>
<td>1 1 8 9 10 11 12 7 1</td>
<td>-1 8 9 10 11 -12 -7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>2 1 1 1 1 2 2 1</td>
<td>1 8 9 10 11 12 7 1 1 1</td>
<td>-1 8 9 10 11 -12 -7 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>2 2 1 1 1 2 1 2</td>
<td>1 8 14 15 16 17 6 7 1</td>
<td>-1 8 14 15 16 -17 6 -7</td>
</tr>
</tbody>
</table>
codewords $N_{\text{min}}$ at the minimum distance $d_{\text{min}}$ should be minimized. The selection procedure can be further refined by considering the number of codewords $N_2$ at the second smallest Euclidean distance $d_2$, and so on, until only one combination of those four parameters remains. Further discussion is elaborated on in Section 4.6.2 and Section 4.8. The optimum values of those design parameters when $n = 2$ are found to be

$$N_{\text{opt}} = 2^D,$$  

(4.8)  

$$B_{\text{opt}} = D + 3,$$  

(4.9)  

$$d_{\text{min}}^2 = 2(D + 2).$$  

(4.10)

Table 7 lists the weight distribution of some schemes in which $B \leq D$. Because $B \leq D$, each codeword has a distinct initial state. There are no pairs of codewords that have the same initial states. Interestingly, for all cases satisfying $B \leq D$, all codewords are orthogonal to each other despite of that fact that bi-orthogonal signaling is used. Their weight distribution and the squared Euclidean distance are simply characterized by $d^2 = 2B$, and $N = 2^B - 1$.

If the size of the input alphabet $n = 4$, then the bit rate is doubled. By assigning $N$ the optimum value, which is $2^{(2D+1)}$ in this case, the weight distribution of squared Euclidean distance is shown in Table 8 as the trellis depth $D$ and block length $B$ are varied. The first thing we notice is that as $D$ increases, $N$ grows very fast — $D$ at moderate value of five, the optimum value of $N$ needs to be 2048. The next thing we experience is that the weight distribution when $n = 4$ is less regular than that when $n = 2$. The values of $d_{\text{min}}$ does not have the same relationship with $D$ and $B$ as we

\footnote{Asymptotically.}
Table 7: The weight distribution of Euclidean distances for $n = 2$ when $B \leq D$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$D$</th>
<th>$B$</th>
<th>$N_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>6</td>
<td>63</td>
</tr>
</tbody>
</table>

The weight distribution of Euclidean distances for $n = 2$ when $B \leq D$.

have experienced in $n = 2$ cases. $N_{\text{min}}$ reaches to its minimum value at $B = D + 2$ when $D = 2, 3, 5$, instead of $B = D + 3$ in $n = 2$ cases.

From experiments, it is shown that if $B \leq D$, all codewords are orthogonal to each other. This is the same as that of $n = 2$ cases. In the table, when $D = 5$, we change $B$ from four to six, it seems that the weight distribution depends on $B$. However, we are interested in finding out if $d_{\text{min}}$ is able to reach to its maximum value $d_{\text{free}}$. If $B \leq D$, there are no codewords sharing the same initial state, that is, each codeword has a distinct initial state, therefore we cannot obtain the information about $d_{\text{free}}$ for such block length. Thus we need to examine those situations when
Table 8: The weight distribution of Euclidean distances for $n = 4$ with various $N$, $D$ and $B$.

$B \geq D + 1$. When $n = 2$, $d_{\text{min}}$ is not equal to $d_{\text{free}}$ when $B = D + 1$, but it is when $B = D + 2$. So we may ask if $d_{\text{min}}$ has reached to $d_{\text{free}}$ when $B = D + 1$ in these $n = 4$ cases. We need to examine both $d_{\text{min}}$ and $d_{\text{free}}$ by analyzing their error events with minimum length of $D + 1$.

When $n = 4$, there are four codewords that start from and end to the same state, one of them is already chosen as a reference, so we only need to examine the other three codewords, whose information is listed in Table 9 when $N = 32$, $D = 2$, $B = 3$. We see that two out of three are orthogonal to the reference. The length of these two is three which is minimum, thus we have $d_{\text{free}} = 6$. The minimum Euclidean distance is six too. $6 = 3 \times 2$. This means that the worst case is that two codewords are orthogonal to each other. Thus $d_{\text{min}} = d_{\text{free}}$ at $B = D + 1$. That is why $d_{\text{min}}$ can
not be further improved as $B$ increases.

<table>
<thead>
<tr>
<th>$d_{\text{free}}^2$</th>
<th>$d_{\text{min}}^2$</th>
<th>$d_{11}^2$</th>
<th>source</th>
<th>state path</th>
<th>codeword</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>8</td>
<td>2 2 3</td>
<td>1 6 10 1</td>
<td>-1 -11 20</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>3 3 4</td>
<td>1 11 15 1</td>
<td>2 22 -30</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>4 4 2</td>
<td>1 16 5 1</td>
<td>-2 -32 -9</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 9: The relationship between $d_{\text{free}}^2$ and $d_{\text{min}}^2$ when $(N, n, D, B) = (32, 4, 2, 3)$.

Figure 39: The transmission symbol assignment of an arbitrary state when $n = 4$.

To explain why $d_{\text{free}}$ is so small, we need to look at the transmission symbol assignment which is shown in Figure 39. When $n = 2$, we are able to assign $s_i(t)$ and its antipodal $-s_i(t)$ so that the distance between the signals of two outgoing branches are two. But when $n = 4$, we need another pair, $s_j(t)$ and $-s_j(t)$, so as to assign all branches as shown in Figure 39. The minimum distance between them is reduced to $\sqrt{2}$. As $n$ increases, the growth in the number of pairs whose Euclidean distances are equal to $\sqrt{2}$ is much faster than the number of antipodal pairs whose Euclidean distances are equal to 2. Figure 78 and Figure 79 in Section A tell us that in this situation the probability of first kind of word error $P_{1e}$ becomes smaller, while the probability of second kind word error will become larger. The performance
of bi-orthogonal signaling approaches the performance of orthogonal signaling as the
dimension of bi-orthogonal signal constellation becomes larger\textsuperscript{7}.

\subsection{Cyclic properties}

In this section, some interesting observations about source, trellis path and codeword
structure are presented and a group rule is given.

Because of the state constraint applied in the encoder, every legal trellis path has
the same initial state and ending state. The path is tail-biting, or circular. For a
source with $B$ information symbols, there are $B$ state transitions, but the state path
records $B + 1$ states. In the examination of the properties of the state transition
paths or trellis paths, their ending states can be ignored because of circularity of the
trellis paths.

Table 10 through Table 13 categorize the codebooks according to their state
transitions for $N = 8, n = 2, D = 3$ with the block length $B = 1, 2, 3, 4, 5$ respectively.
In these tables $\Delta S_{1j}$ denotes the number of different symbols between two blocks of
source data, one of which is the reference block. The source-book starts from $1 \ldots 1$
and ends to $n \ldots n$ by following the conventions used in [21]. The index $j$ is the order
of a codeword in such an arrangement, and $1 \leq j \leq n^B$.

\begin{tabular}{cccccc}
index $j$ & Source Data & Trellis Path & Codeword & $d_{1j}^2$ & $\Delta S_{1j}$ \\
1 & 1 & 1 1 & 1 & 0 & 0 \\
2 & 2 & 2 2 & -2 & 2 & 1 \\
\end{tabular}

Table 10: Categorization of the codebook for (8,2,3,1) scheme.

\textsuperscript{7}See Appendix A.
Table 11: Categorization of the codebook for (8,2,3,2) scheme.

Table 10 is a special case when a codeword is equivalent to a symbol. When the source is ‘1’, its legal state path starts from state one and ends up to state one; if it is ‘2’, then the state path is from state two to state two. This is the important assumption made in the design of a state permuted trellis structure, as shown in Figure 16, so as to achieve the circular permutation we desire. Therefore, for all ‘1’ or all ‘2’ sources \( n = 2 \), their state transitions are just a repetition of the same state, either state 1 or state 2, no matter what value of \( B \) is chosen. This is also true in other tables. Therefore, in the following analysis, the examination of all ‘1’ and all ‘2’ source blocks is omitted.

Table 12: Categorization of the codebook for (8,2,3,3) scheme.
In Table 11, the trellis path of source "2 1" is a left-shifted version of that of source "1 2", so are its codewords. Table 12 shows that eight source blocks can be separated into four groups, so are their trellis paths and codewords. Each group has its own pattern. Moreover, any element in any group is a shifted version of the others in the same group, either right-shifted or left-shifted, if the number of elements of this group is greater than one. The number of shifts between elements in trellis path group or codeword group depends on the number of shifts between their corresponding elements in the source group. In this case, the source blocks with only one '2' form a group $g_1$, while the source blocks with two '2' form another group $g_2$. Each group has a distinct pattern, with respect to either source, trellis path or codeword.

As source block becomes longer, there are more patterns and the codebook can be divided into more groups as Table 13 and Table 14 tell us. Still, each group is governed by one pattern. A closer examination at those source patterns shows that those patterns can be differentiated by the number of '2's and the positions of '2's in the source blocks. If we separate the sourcebook according to the number of '2's and their positions in the source blocks into a number of groups, then the trellis path or the codeword of any source can be obtained by the shifting of the other elements in the same group. The shifting is determined by the nature of shift of '2's in the source block. It should be emphasized that the shift is a circular shift. The summary of these patterns is listed in Table 15 for different $B$, where each "Group" contains a set of the indices of the source blocks having the same patterns.

The above observations suggest that the circular shift in the source group leads to the corresponding shift of the same nature, in terms of both shifting direction
<table>
<thead>
<tr>
<th>index $j$</th>
<th>Source Data</th>
<th>Trellis Path</th>
<th>Codeword</th>
<th>$d_{1j}^2$</th>
<th>$\Delta S_{1j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>2 2 2 2</td>
<td>2 2 2 2</td>
<td>2 2 2 2</td>
<td>-2 -2 -2 -2</td>
<td>8 4</td>
</tr>
<tr>
<td>2</td>
<td>1 1 1 2</td>
<td>7 8 2 3 7</td>
<td>7 8 2 -3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 1 2 1</td>
<td>8 2 3 7 8</td>
<td>8 2 -3 7</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1 2 1 1</td>
<td>2 3 7 8 2</td>
<td>2 -3 7 8</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>2 1 1 1</td>
<td>3 7 8 2 3</td>
<td>-3 7 8 2</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 1 2 2</td>
<td>3 4 5 8 3</td>
<td>3 4 -5 -8</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1 2 2 1</td>
<td>4 5 8 3 4</td>
<td>4 -5 -8 3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>2 2 1 1</td>
<td>5 8 3 4 5</td>
<td>-5 -8 3 4</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2 1 1 2</td>
<td>8 3 4 5 8</td>
<td>-8 3 4 -5</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1 2 1 2</td>
<td>6 7 6 7 6</td>
<td>6 -7 6 -7</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>2 1 2 1</td>
<td>7 6 7 6 7</td>
<td>-7 6 -7 6</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1 2 2 2</td>
<td>5 6 4 1 5</td>
<td>5 -6 -4 -1</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>2 2 2 1</td>
<td>6 4 1 5 6</td>
<td>-6 -4 -1 5</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>2 2 1 2</td>
<td>4 1 5 6 4</td>
<td>-4 -1 5 -6</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>2 1 2 2</td>
<td>1 5 6 4 1</td>
<td>-1 5 -6 -4</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 13: Categorization of the codebook for (8,2,3,4) scheme.

Figure 40: A general encoder model.
<table>
<thead>
<tr>
<th>index $j$</th>
<th>Source Data</th>
<th>Trellis Path</th>
<th>Codeword</th>
<th>$d_{ij}^2$</th>
<th>$\Delta S_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 1 1 1</td>
<td>1 1 1 1 1 1</td>
<td>1 1 1 1 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>2 2 2 2 2</td>
<td>2 2 2 2 2 2</td>
<td>-2 -2 -2 -2 -2</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1 1 1 1 2</td>
<td>8 2 3 4 5 8</td>
<td>8 2 3 4 -5</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 1 1 2 1</td>
<td>2 3 4 5 8 2</td>
<td>2 3 4 -5 8</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1 1 2 1 1</td>
<td>3 4 5 8 2 3</td>
<td>3 4 5 8 2</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1 2 1 1 1</td>
<td>4 5 8 2 3 4</td>
<td>4 5 8 2 3</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>2 1 1 1 1</td>
<td>5 8 2 3 4 5</td>
<td>-5 8 2 3 4</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1 1 1 2 2</td>
<td>4 5 6 7 6 4</td>
<td>4 5 6 7 -6</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1 1 2 2 1</td>
<td>5 6 7 6 4 5</td>
<td>5 6 7 -6 4</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>1 2 2 1 1</td>
<td>6 7 6 4 5 6</td>
<td>6 7 -6 4 5</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>25</td>
<td>2 2 1 1 1</td>
<td>7 6 4 5 6 7</td>
<td>-7 6 4 5 6</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>2 1 1 1 2</td>
<td>6 4 5 6 7 6</td>
<td>-6 4 5 6 -7</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1 1 2 1 2</td>
<td>7 8 2 2 3 7</td>
<td>7 8 2 2 -3</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>1 2 1 2 1</td>
<td>8 2 2 3 7 8</td>
<td>8 2 2 -3 7</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>2 1 2 1 1</td>
<td>2 2 3 7 8 2</td>
<td>-2 2 -3 7 8</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1 2 1 1 2</td>
<td>2 3 7 8 2 2</td>
<td>2 -3 7 8 -2</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>2 1 1 2 1</td>
<td>3 7 8 2 2 3</td>
<td>-3 7 8 -2 2</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1 1 2 2 2</td>
<td>6 7 8 3 7 6</td>
<td>6 7 -8 3 -7</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>1 2 2 2 1</td>
<td>7 8 3 7 6 7</td>
<td>7 8 -3 7 6</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>29</td>
<td>2 2 2 1 1</td>
<td>8 3 7 6 7 8</td>
<td>-8 -3 7 6 7</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>26</td>
<td>2 2 1 1 2</td>
<td>3 7 6 7 8 3</td>
<td>-3 -7 6 7 -8</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>2 1 1 2 2</td>
<td>7 6 7 8 3 7</td>
<td>-7 6 7 -8 3</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>1 2 1 2 2</td>
<td>1 1 5 6 4 1</td>
<td>1 1 -5 -6 -4</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>2 1 2 2 1</td>
<td>1 5 6 4 1 1</td>
<td>-1 5 -6 -4 1</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>1 2 2 1 2</td>
<td>5 6 4 1 1 5</td>
<td>5 -6 -4 1 -1</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>27</td>
<td>2 2 1 2 1</td>
<td>6 4 1 1 5 6</td>
<td>-6 -4 -1 -5</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>2 1 2 1 2</td>
<td>4 1 1 5 6 4</td>
<td>-4 1 -1 -5 -6</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>2 1 2 2 2</td>
<td>8 3 4 1 5 8</td>
<td>-8 3 -4 -1 -5</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>1 2 2 2 2</td>
<td>3 4 1 5 8 3</td>
<td>3 -4 -1 -5 -8</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>31</td>
<td>2 2 2 2 1</td>
<td>4 1 5 8 3 4</td>
<td>-4 -1 5 -8 3</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>2 2 2 1 2</td>
<td>1 5 8 3 4 1</td>
<td>-1 -5 -8 3 -4</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>28</td>
<td>2 2 1 2 2</td>
<td>5 8 3 4 1 5</td>
<td>-5 -8 3 -4 1</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 14: Categorization of the codebook for (8,2,3,5) scheme.
Table 15: Patterns of source data, trellis paths and codewords when $N = 8$, $n = 2$ and $D = 3$ and their corresponding groups whose elements are the indices of source blocks.
and the number of shifts in its trellis path group and its codeword group. This is valid only among those elements that belong to the same group, not in the scope of the whole codebook. This means that if the input has some circular properties, the output will possess the same properties. It further suggests that the encoder shown in Figure 40 will possess such cyclic properties too.

Figure 41 shows the relationship between the source, trellis path, and codeword after the source is separated into two groups. The figure shows one-to-one correspondence between source group, trellis path group and codeword group. This suggests that to examine the cyclic properties of trellis paths and codewords, we only need to examine the cyclicity of the source. Once these cyclic properties are known, they will apply to the trellis path group and codeword group as well.

![Figure 41: A simple illustration of the relationship between source, trellis path and codeword.](image)

---

8For convenience of drawing.
The cyclic properties are very useful. To construct a codebook, usually we have to search the trellis paths for all \( n^B \) possible source blocks. The above observations suggest that a saving in computation during encoding can be achieved. Instead of examining a whole sourcebook, we only need to find out a number of distinct source patterns, which are only a fraction of \( n^B \). However, a problem arises that is how to identify those patterns and how can we know which sources share the same patterns. Obviously, an algorithm to the above problem is desired.

Instead of examining actual shifting among \( n \)-ary symbols in the source blocks, we look at the indices of the sources. As explained above, all ‘1’ or all ‘\( n \)’ source\(^9\) are not examined, which are indexed by ‘1’ and ‘\( n^B \)’ respectively. However, the indices in different groups shown in Table 15 seems totally irregular except for the situation in which those source blocks have only one ‘2’. Nevertheless, by hand grouping (8, 2, 3, 6) case, the following rule was found for this particular case and is applicable to the same set of \( N, n, D \) with different \( B \). The rule is to separate the set of source indices \{2, 3, 4, \ldots, n^B - 1\} into \( m \) groups, that is, \( g_1, \ldots, g_m \). The source with index ‘1’ and the one with index ‘\( n^B \)’ form their own groups respectively. The rule of grouping for \( n = 2 \) is stated as follows

- Determination of the first element in each group.

1. The first element in the first group \( g_1 \) is 2.

2. The first element in next group is the next available even number, which is closest to the first element of previous adjacent group.

3. Once an index is selected, it is declared unavailable.

\(^9\)Actually we are discussing \( n = 2 \) cases because of interests.
• Determination of increments in each group.

1. The first increment in each group is one less than the first element in this group. It is always an odd number.

2. The next increment is always a double of previous adjacent one.

• Denote the sum of present element and present increment as $X$, the next element in each group is $X$ modulo $2^B - 1$, i.e., mod $(X, 2^B - 1)$.

The first element in each source group can be called the leader of the group or leading pattern. The trellis paths and codewords of other sources can be derived from their leading patterns by the nature of shift between the leader and the elements.

Assume the codebook can be separated into $m$ groups, excluding all ‘1’ and all ‘2’ cases. The number of elements in each group is denoted by $q_i$, where $i = 1, \ldots, m$. $q_i$ may not be equal to each other. The above rule is expressed in equations as follows

$$
e_{11} = 2,$$
$$e_{i1} = \text{an available even number that is closest to } e_{(i-1)1},$$
$$e_{i2} = \mod (e_{i1} + s_{i1}, 2^B - 1),$$
$$e_{i3} = \mod (e_{i2} + s_{i2}, 2^B - 1),$$
$$\vdots$$
$$e_{ij} = \mod (e_{ij} + s_{ij}, 2^B - 1),$$
$$\vdots$$
$$e_{iq_i} = \mod (e_{i(q_i-1)} + s_{i(q_i-1)}, 2^B - 1).$$
where \( e_{ij} \) and \( s_{ij} \) are the \( j \)-th element and the \( j \)-th increment in the \( i \)-th group, respectively, as shown in Figure 42. \( j = 1, \ldots, q_i - 1 \) and \( i = 1, \ldots, m \).

Because \( s_{i1} = e_{i1} - 1 \) and \( s_{ij} = 2^{(j-1)} s_{i1} = 2^{(j-1)}(e_{i1} - 1) \), the above equations can be summarized as a recursive equation as

\[
\begin{align*}
\text{\( e_{i1} = 2 \),} \\
\text{\( e_{i1} \) = an available even number that is closest to \( e_{(i-1)1} \),} \\
\text{\( e_{i(j+1)} = \text{mod}(e_{ij} + 2^j(e_{i1} - 1), 2^B - 1) \).} \\
\end{align*}
\]

(4.11)

Figure 42: The generation of \( i \)th group.

We then apply Equation (4.11) to the (8, 2, 3, 5) case. We need to separate the index set \( \{2, 3, \ldots, 31\} \) into several groups. The procedure is briefly explained as follows, with the help of Figure 43:

1. Group \( g_1 \) starts from index 2 and the first increment is 1. The next element is obtained by \( \text{mod}(2 + 1, 31) = 3 \), and so on. Notice that the last element in \( g_1 \) returns to the first element after the modulo operation, \( \text{mod}(17 + 16, 31) = 2 \). This completes a circular shift.

2. The first element in \( g_2 \) is 4, which is the next closest available even number to 2. Notice that the last element in \( g_2 \) also returns back to the first element in \( g_2 \) after \( \text{mod}(18 + 48, 31) = 4 \). \( g_3 \) through \( g_6 \) can be obtained in a similar way. But the first element in \( g_5 \) is not 10, because 10 already appeared in \( g_3 \). The
first element of $g_6$ is 24 because all the even numbers between 12 and 22 are unavailable.

Figure 43: The grouping of (8,2,3,5) scheme.
4.2 Probability of Codeword Error $P_{ce}$

The calculation of $P_{ce}$ can be separated into two cases.

4.2.1 $P_{ce}$ of two-codeword decoding

If there are only two candidate codewords, then the detection is merely a binary hypothesis test. The simplified algorithm of ML binary hypothesis test has been shown in Equation (3.13). To obtain the probability of codeword error $P_{ce}$, we need to know the probability distribution of the decision metric.

Let $m = C_1 - C_2$, then $y = (C_1 - C_2)^T x = m^T x$. Because of linear transformation, $y$ is also a Gaussian variable with distribution:

$$y \sim N(m_y, \sigma_y^2), \quad x \in C_i. \quad (4.12)$$

where

$$m_y = m^T C_i = (C_1 - C_2)^T C_i \quad (4.13)$$

$$\sigma_y^2 = m^T m \sigma_x^2 = \frac{N_0}{2} (C_1 - C_2)^T (C_1 - C_2) \quad (4.14)$$

Note that the squared Euclidean distance between codewords $C_1$ and $C_2$ is $d^2(C_1, C_2) = (C_1 - C_2)^T (C_1 - C_2)$. Thus $m^T m = d^2(C_1, C_2)$.

The mean $m_{yi}$ and variance $\sigma_{yi}^2$ of $y$ when codeword $C_1$ or codeword $C_2$ is transmitted can be easily obtained as follows

$$m_{y1} = -m_{y2} = m_y \quad (4.15)$$

$$\sigma_{y1}^2 = \sigma_{y2}^2 = \sigma_y^2 \quad (4.16)$$

where

$$m_y = (C_1 - C_2)^T C_1 \quad (4.17)$$
\[
\sigma_y^2 = (C_1 - C_2)^T(C_1 - C_2)\sigma_x^2 \\
= d^2(C_1, C_2)\sigma_n^2.
\] (4.18)

Then we have

\[
y \sim N(m_y, \sigma_y^2), \quad x \in C_1,
\] (4.20)

\[
y \sim N(-m_y, \sigma_y^2), \quad x \in C_2.
\] (4.21)

The probability of codeword error \(P_{ce}\) is calculated by

\[
P_{ce} = P(C_1)P(y < 0|C_1) + P(C_2)P(y > 0|C_2)
\]

\[
= P(C_1) \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}\sigma_y^2} e^{-\frac{(y-m_y)^2}{2\sigma_y^2}} dy + P(C_2) \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_y^2} e^{-\frac{(y+m_y)^2}{2\sigma_y^2}} dy
\]

\[
= \frac{1}{2} [1 - Q\left(\frac{0 - m_y}{\sigma_y}\right)] + \frac{1}{2} Q\left(\frac{0 + m_y}{\sigma_y}\right)
\]

\[
= Q\left(\frac{m_y}{\sigma_y}\right)
\] (4.22)

where \(Q(\alpha)\) is defined as

\[
Q(\alpha) = \int_{\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy,
\] (4.23)

\[
Q(-\alpha) = 1 - Q(\alpha).
\]

Because \(C_1^T C_1 = C_2^T C_2\), we have

\[
d^2(C_1, C_2) = (C_1 - C_2)^T(C_1 - C_2) = 2(C_1 - C_2)^T C_1 = 2m_y
\]

then Equation (4.22) is equivalent to

\[
P_{ce} = Q\left(\frac{d(C_1, C_2)}{2\sigma_n}\right)
\] (4.24)

\(^{10}\)The subscript "ce" represents the codeword error.
4.2.2 $P_{ce}$ of multi-codeword decoding

If the number of candidate codewords is more than two, the detection problem can be categorized as a multi-hypothesis test. The multi-hypothesis test is described in Section 3.2.1. If the maximum inner product criterion is used, then the $j$-th inner product metric $y_j = C'_j x$ (Equation (3.14)) is also Gaussian distributed. The conditional probability of making a codeword error, given that codeword $j$ was transmitted, can be obtained by

$$P(ce|j) = 1 - \text{Prob}(y_j \geq y_k|C_j), \quad \forall k \neq j$$

$$= 1 - \int_{-\infty}^{\infty} dy_j \int_{-\infty}^{y_j} \ldots \int_{-\infty}^{y_j} p(y|C_j) dy'$$

(Equation (4.25))

in which

$$y = [y_1, y_2, \ldots, y_M],$$

and

$$dy' = dy_1 dy_2 \ldots dy_{j-1} dy_{j+1} \ldots dy_M.$$  

An exact expression for $P(ce|j)$ in Equation (4.25) involves finding an explicit expression of the joint multivariate p.d.f. $p(y|C_j)$, which only exists for some specific signals, such as orthogonal signals. Moreover, even if their joint p.d.f. can be found, the calculation of multi-dimensional integrals is quite involved, not to mention the difficulty in partitioning integration regions. Hence, using an upper bound is a solution to the performance analysis of $M$-ary decision problem, and is applicable to any signal set [37].

The probability of codeword error $P_{ce}$ can be obtained by first finding the conditional probability of codeword error $P(ce|j)$ given that $j$th codeword is being transmitted, and finally summing this conditional probability over all possible codewords.
As we know, all the codewords are equally probable and have the same average signal power. Because of the ML criterion, the minimum Euclidean distance criterion and the maximum inner product criterion are equivalent for equally likely signals, the same average signal power transmission in AWGN channels is as shown in Figure 23. For convenience of description in signal space, assume that the minimum Euclidean distance criterion is used. Then we have

\[ P_{ce} = \sum_{j=1}^{M} P(ce|j)P(j) \]

\[ = \frac{1}{M} \sum_{j=1}^{M} P(ce|j) \]

\[ = \frac{1}{M} \sum_{j=1}^{M} \text{Prob}[d_j \geq d_k|C_j], \quad \forall k \neq j \]

where \(d_j\) is the pairwise Euclidean distance between the received sequence \(x\) and codeword \(C_j\).

The upper bound of \(P_{ce}\) can be found by applying the union bound method as follows:

\[ P_{ce} \leq \frac{1}{M} \sum_{j=1}^{M} \sum_{k=1}^{M} \text{Prob}[d_j \geq d_k|C_j]. \quad (4.26) \]

Because of symmetry, we can pair those terms in Equation (4.26) to write in the form of

\[ P_{ce} \leq \frac{1}{M} \sum_{j=1}^{M} \sum_{k=1}^{M} \{ \frac{1}{2} \text{Prob}[d_j \geq d_k|C_j] + \frac{1}{2} \text{Prob}[d_k \geq d_j|C_k] \}. \]

Then

\[ P_{ce} \leq \frac{1}{M} \sum_{j=1}^{M} \sum_{k=1}^{M} P_{ce}(j, k), \quad (4.27) \]

where

\[ P_{ce}(j, k) = \frac{1}{2} \text{Prob}[d_j \geq d_k|C_j] + \frac{1}{2} \text{Prob}[d_k \geq d_j|C_k]. \]
$P_{ce}(j, k)$ is called pairwise probability of codeword error. It is the probability of codeword error when $C_j$ is transmitted but $C_k$ is detected instead. If we put all those pairs of $P_{ce}(j, k)$ into a matrix $P$, such that

$$P = \begin{bmatrix} P_{ce}(1, 1) & P_{ce}(1, 2) & \ldots & P_{ce}(1, M) \\ P_{ce}(2, 1) & P_{ce}(2, 2) & \ldots & P_{ce}(2, M) \\ \vdots & \vdots & \ddots & \vdots \\ P_{ce}(M, 1) & P_{ce}(M, 2) & \ldots & P_{ce}(M, M) \end{bmatrix}$$

where $P_{ce}(j, j) = 0$ for all $j$. Equation (4.27) can then be rewritten as

$$P_{ce} \leq \frac{1}{M} 1^T P 1,$$

where $1$ is an $M$-vector column whose elements are all one.

Because $P_{ce}(j, k) = Q(d_{jk}/(2\sigma_n))$ and $Q(\cdot)$ is a monotonic function, from Section 4.1.1, we know that $P$ is a uniform matrix, in the sense of either row-wise or column-wise. Thus $\sum_{k=1}^{M \atop k \neq j} P_{ce}(j, k)$ is independent of the choice of $j$, which means that the row sum of $P$ is regardless of the choice of reference codeword $C_j$, i.e.,

$$\sum_{k=1}^{M \atop k \neq j} P_{ce}(j, k)$$

has the same value with respect to an arbitrary $j$. Hence

$$1^T P 1 = M \sum_{k=1}^{M \atop k \neq j} P_{ce}(j, k) = M \alpha.$$

where $\alpha = \sum_{k=1}^{M \atop k \neq j} P_{ce}(j, k)$. Because $j$ can be an arbitrary value, we can simply choose $j = 1$, with codeword 1 as reference. Thus the upper bound for probability of codeword error can be obtained by

$$P_{ce} \leq \sum_{k=2}^{M} P_{ce}(1, k)$$

(4.29)

### 4.2.3 Lower bound for probability of codeword error

In this section, the lower bound of probability of codeword error for transmission over the AWGN channel is discussed.
When a codeword is transmitted over the channel, a real-life receiver does not have any prior knowledge or side information about which codeword may have been transmitted. The received codeword is detected by guessing according to the maximum likelihood rule for optimum detection. Consider an ideal situation in which side information on the transmitted symbols is provided by a genie. If the receiver makes its decisions by optimally exploiting the side information, it is sure that it cannot be outperformed by any receiver working without the genie’s aid. The calculation of the lower bound is based on this fact. If the probability of codeword error achieved by the genie-aided receiver is denoted by \( P_G(ce) \), then the probability of codeword error for any real-life receiver \( P_{ce} \) is larger than \( P_G(ce) \). Thus, \( P_G(ce) \) is the lower bound of \( P_{ce} \).

A genie-aided receiver operates as follows. When the genie observes a sequence of transmitted symbols, or equivalently, the codeword \( C \), it tells the receiver that the transmitted sequence is either codeword \( C \) or codeword \( C' \), where \( C' \) is randomly selected from possible codewords having the smallest Euclidean distance \( d_{\text{min}} \) from \( C \). The genie-aided receiver has the prior information about which codeword may have possibly been transmitted. In this situation, the task of the receiver is to choose one out of two known codewords perturbed by white Gaussian noise. Thus, the multi-category detection problem among \( M \) possible codewords is simply reduced to a two-category detection problem between the codeword \( C \) and \( C' \). Therefore the probability of codeword error of this genie-aided receiver given codeword \( C \) is transmitted can be obtained from Equation (4.24) as follows

\[
P_G(ce|C) = Q\left( \frac{d(C, C')}{2\sigma_n} \right) = Q\left( \frac{d_{\text{min}}}{2\sigma_n} \right).
\] (4.30)
Averaging over the codeword set, the unconditional probability of codeword error \( P_G(ce) \) is

\[
P_G(ce) = \sum_C P(C)P_G(ce|C) \leq \sum_C I(C)P(C)Q\left(\frac{d_{\text{min}}}{2\sigma_n}\right)
\]

where \( I(C) \) defines the index of the codeword \( C' \) that satisfies \( d(C, C') = d_{\text{min}} \) as

\[
I(C) = \begin{cases} 
1, & \text{if minimum } [d(C, C')] = d_{\text{min}}, \forall C', \\
0, & \text{otherwise}.
\end{cases}
\]

Therefore, \( I(C) \) makes all those codewords \( C' \) whose pairwise Euclidean distances to the codeword \( C \) are larger than \( d_{\text{min}} \) be discarded from the summation in Equation (4.31). Define \( \varphi \) as

\[
\varphi = \sum_C I(C)P(C)
\]

Then we have

\[
P_G(ce) \geq \varphi Q\left(\frac{d_{\text{min}}}{2\sigma_n}\right).
\]

\( \varphi \) represents the probability that a codeword chosen at random is decoded into another codeword such that the Euclidean distance between them is \( d_{\text{min}} \). Assume \( N_{\text{min}} \) is the number of codewords whose pairwise Euclidean distances to codeword \( C \) is \( d_{\text{min}} \). Section 4.1.1 tells us that because of the uniformity of the scheme, \( N_{\text{min}} \) is independent of the choice of \( C \). Then we have

\[
\varphi = \sum_C I(C)P(C) = \frac{N_{\text{min}}}{M}.
\]

Therefore, the lower bound for the probability of codeword error is obtained as

\[
P_{ce} \geq P_G(ce) \geq \frac{N_{\text{min}}}{M} Q\left(\frac{d_{\text{min}}}{2\sigma_n}\right).
\]
4.3 Relationship between $P_{ce}$, $P_{se}$ and $P_{be}$

Now we have derived the upper bound and lower bound for $P_{ce}$. However, in order to evaluate the system's performance, probability of bit error is usually used instead of $P_{ce}$. Denote the probability of symbol error and probability of bit error by $P_{se}$ and $P_{be}$, respectively. There are no direct ways to calculate $P_{se}$ or $P_{be}$. To obtain $P_{se}$ and $P_{be}$, we have to resort to the bounding method as presented in Section 4.2.2 and Section 4.2.3. In this section, the analytical expressions of pairwise probability of symbol error and pairwise probability of bit error are derived respectively.

4.3.1 Pairwise probability of symbol error

We know that the mapping between the information matrix $IM$ and the codeword matrix $CM$ is one-to-one correspondence as shown in Figure 44. $IM$ has an alphabet size of $n = 2^k$, that is, each source symbol has $k$ bits.

![Diagram](image)

**Figure 44:** One-to-one correspondence between information matrix and codeword matrix.
If codeword \( C_X \) is transmitted but the detector selects codeword \( C_Y \) instead, then an error occurs. Because of the one-to-one correspondence between information sources and codewords, the above error means that their corresponding information block \( I_Y \) is decoded instead of \( I_X \), where

\[
I_X = [I_{x1}, I_{x2}, \ldots, I_{xB}],
\]

and

\[
I_Y = [I_{y1}, I_{y2}, \ldots, I_{yB}],
\]

\( I_X \) is the source of codeword \( C_X \) and \( I_Y \) is the source of codeword \( C_Y \). A symbol error occurs when \( I_{xi} \neq I_{yi}, \ i = 1, \ldots, B \). Denote \( \Delta S(X,Y) \) as the number of mismatched symbols between \( I_X \) and \( I_Y \). The conditional probability of a symbol being in error when codeword \( C_Y \) is favored instead of codeword \( C_X \) is

\[
P(S|C_X \rightarrow Y) = \frac{\Delta S(X,Y)}{B}.
\]

Similarly, \( P(S|C_Y \rightarrow X) \) denotes the conditional probability of a symbol being in error when codeword \( C_X \) is favored instead of the transmitted \( C_Y \). Because of commutativity, \( P(S|C_X \rightarrow Y) = P(S|C_Y \rightarrow X) \). Thus we use \( P(S|C_{X,Y}) \) to denote the pairwise conditional probability of symbol error between codeword \( C_X \) and codeword \( C_Y \).

The pairwise probability of symbol error \( P_{se}(X,Y) \) between sequence \( X \) and sequence \( Y \) is thus

\[
P_{se}(X,Y) = P(S|C_{X,Y})P_{ce}(X,Y)
= \frac{\Delta S(X,Y)}{B} \cdot P_{ce}(X,Y) \tag{4.36}
\]

where \( P_{ce}(X,Y) \) is the pairwise probability of codeword error between codeword \( C_X \) and codeword \( C_Y \).
Because \(0 \leq \triangle S(X, Y) \leq B\), we can bound \(P_{se}(X, Y)\) by

\[
0 \leq P_{se}(X, Y) \leq P_{ce}(X, Y).
\] (4.37)

### 4.3.2 Pairwise probability of bit error

Each information symbol has \(k = \log_2 n\) bits. When \(n = 2\), a symbol is equivalent to a bit. In this special situation, the pairwise probability of bit error \(P_{be}(X, Y) = P_{se}(X, Y)\). Generally, \(P_{be}\) can either be obtained from \(P_{se}\) or from \(P_{ce}\) directly. Two methods are similar. The following derivation is to obtain \(P_{be}\) from \(P_{ce}\) directly.

We can further represent a symbol in the information matrix \(IM\) in bits, such that each symbol in \(I_X\) and \(I_Y\) is represented as

\[
I_{xi} = I_{x1i}, \ldots, I_{xki};
\]

and

\[
I_{yi} = I_{y1i}, \ldots, I_{yk};
\]

respectively, where bits \(I_{xij}, I_{yij} = 0, 1; \ i = 1, \ldots, B; \ j = 1, \ldots, k.\)

A bit error is made when \(I_{xij} \neq I_{yij}, \ i = 1, \ldots, B; \ j = 1, \ldots, k.\) The number of mismatched bits between these two source bit sequences

\[
[I_{x11}, I_{x12}, \ldots, I_{x1k}, I_{x21}, I_{x22}, \ldots, I_{x2k}, \ldots, I_{xB1}, I_{xB2}, \ldots, I_{xBk}],
\]

and

\[
[I_{y11}, I_{y12}, \ldots, I_{y1k}, I_{y21}, I_{y22}, \ldots, I_{y2k}, \ldots, I_{yB1}, I_{yB2}, \ldots, I_{yBk}],
\]

is denoted as \(\triangle B(X, Y)\). Therefore the pairwise conditional probability of bit error is

\[
P(B|C_{X,Y}) = \triangle B(X, Y)/(B \cdot k).
\]
Then the pairwise probability of bit error is

\[
P_{be}(X, Y) = P(B|C_{X,Y})P_{ce}(X, Y) = \frac{\Delta B(X, Y)}{(B \cdot k) \cdot P_{ce}(X, Y)}. \tag{4.38}
\]

Note that the pairwise bit errors \(\Delta B(X, Y)\) is equivalent to the pairwise symbol errors \(\Delta S(X, Y)\) when \(n = 2\), i.e., \(\Delta B(X, Y) = \Delta S(X, Y)|_{n=2}\).

4.4 \(E_b/N_0\) and \(\sigma_n\)

One of the most important comparisons in digital communication systems is based on how efficiently the system can utilize the available signal energy to transmit information. A useful measure of this efficiency is the energy per bit. Since all systems have noise in them, the probability of error depends on that noise. The energy utilization efficiency is defined as \(E_b/N_0\), where \(E_b\) is the energy per bit and \(N_0\) is the one-sided noise spectral density. To employ this efficiency measure, we need to relate this quantity to the signal noise ratio \(P_s/P_n\), where \(P_s\) is the average symbol power and \(P_n\) is the average noise power. In following analysis, we begin by examining the relationship between average codeword energy \(E_c\), average symbol energy \(E_s\) and average bit energy \(E_b\). We then express \(\sigma_n\) in terms of \(E_b/N_0\).

4.4.1 The relationship between \(E_c\), \(E_s\) and \(E_b\)

Because each codeword contains \(B\) channel symbols and each symbol has the same average energy, we have \(E_c = BE_s\).

Even though \(2N\) bi-orthogonal signal waveforms are generated by the TCM modulator, only \(k = \log_2 n\) bits are modulated at a time, instead of \(\log_2(2N)\) bits, into a
signal waveform. Thus $E_s$ and $E_b$ can be related by

$$E_s = kE_b.$$ 

Therefore,

$$E_b = \frac{E_s}{\log_2 n} = \frac{E_c}{B \log_2 n}. \quad (4.39)$$

### 4.4.2 $\sigma_n$ in terms of $E_b/N_0$

The following formula [17] which relates $E_b/N_0$ to $P_s/P_n$

$$\frac{E_b}{N_0} = \frac{P_s W}{P_n R}$$

is well-known, where $R$ is the bit rate, $W$ is the detection bandwidth. Assume the symbol duration is $T$. Since

$$R = \frac{\log_2 n}{T} = \frac{k}{T}$$

We can then rewrite $E_b/N_0$ to the following form

$$\frac{E_b}{N_0} = \frac{P_s W T}{P_n k}$$

For FSK signaling, $WT \approx 1$. Therefore

$$\frac{E_b}{N_0} \approx \frac{P_s}{P_n} \left( \frac{1}{k} \right) \quad (4.40)$$

Assume the average symbol energy is normalized to unity, $E_s = P_s T = 1$. Because the noise variance $\sigma_n^2 = N_0/2$ and $P_n = N_0 W$, from Equation (4.40), $\sigma_n$ can be expressed in terms of $E_b/N_0$ as follows

$$\sigma_n = \frac{1}{\sqrt{2k(E_b/N_0)}} = \frac{1}{\sqrt{2(E_b/N_0) \log_2 n}}. \quad (4.41)$$
\( \sigma_n \) can also be represented in terms of \( E_s/N_0 \) or \( E_c/N_0 \) respectively, which we can summarize as follows. For convenience, \( 1/\sigma_n \) is used instead.

\[
\frac{1}{\sigma_n} = \begin{cases} 
\sqrt{\frac{2E_b \log_2 n}{N_0}}, & \text{in terms of } E_b/N_0, \\
\sqrt{\frac{2E_s}{N_0}}, & \text{in terms of } E_s/N_0, \\
\sqrt{\frac{2E_c}{B N_0}}, & \text{in terms of } E_c/N_0.
\end{cases}
\]  

(4.42)

Equation (4.24) gives us the pairwise probability of codeword error between codeword 1 and codeword \( j \) as

\[
P_{ce}(1, j) = Q \left( \frac{d_j}{2\sigma_n} \right) \tag{4.43}
\]

where \( d_j = d(C_1, C_j) \) is the Euclidean distance between codeword \( C_1 \) and codeword \( C_j \).

From Equation (4.42) and Equation (4.43), we can obtain the pairwise probability of codeword error \( P_{ce}(1, j) \) in terms of \( E_c/N_0 \) as

\[
P_{ce}(1, j) = Q \left( \sqrt{\frac{d_j^2 E_c}{2BN_0}} \right). \tag{4.44}
\]

Similarly, the pairwise probability of symbol error \( P_{se}(1, j) \) and the pairwise probability of bit error \( P_{be}(1, j) \) in terms of \( E_s/N_0 \) and \( E_b/N_0 \) can be obtained respectively as follows

\[
P_{se}(1, j) = \frac{\Delta S_j}{B} P_{ce}(1, j) = \frac{\Delta S_j}{B} Q \left( d_j \sqrt{\frac{E_s}{2N_0}} \right) \tag{4.45}
\]

\[
P_{be}(1, j) = \frac{\Delta B_j}{B \cdot k} P_{ce}(1, j) = \frac{\Delta B_j}{B \cdot k} Q \left( d_j \sqrt{\frac{E_b \log_2 n}{N_0}} \right) \tag{4.46}
\]

where \( \Delta S_j = \Delta S(1, j) \) and \( \Delta B_j = \Delta B(1, j) \) are the numbers of mismatched symbols and mismatched bits between sequence 1 and sequence \( j \), respectively.

4.5 The Upper Bounds and Lower Bounds of \( P_{se} \) and \( P_{be} \)

In this section, the analytical expression of upper bounds and lower bounds for \( P_{se} \) and \( P_{be} \) are derived respectively.
4.5.1 Upper bounds

Similar to the derivation of the upper bound for $P_{ce}$ (Section 4.2.2), the upper bounds for $P_{se}$ and $P_{be}$ can be obtained in terms of pairwise probability of symbol error $P_{se}(l, j)$ and pairwise probability of bit error $P_{be}(l, j)$ as

$$P_{se} \leq \sum_{j=2}^{M} P_{se}(1, j) = \sum_{j=2}^{M} \frac{\Delta S_j}{B} P_{ce}(1, j) = \sum_{j=2}^{M} \frac{\Delta S_j}{B} Q\left(d_j \sqrt{\frac{E_s}{2N_0}}\right)$$

(4.47)

$$P_{be} \leq \sum_{j=2}^{M} P_{be}(1, j) = \sum_{j=2}^{M} \frac{\Delta B_j}{B \cdot k} P_{ce}(1, j) = \sum_{j=2}^{M} \frac{\Delta B_j}{B \cdot k} Q\left(d_j \sqrt{\frac{E_b \log_2 n}{N_0}}\right)$$

(4.48)

respectively.

4.5.2 Lower bounds

The derivation of the lower bounds for $P_{se}$ and $P_{be}$ is similar to that of $P_{ce}$ as shown in Section 4.2.3. However, the pairwise conditional probabilities of symbol error and the pairwise conditional probabilities of bit error might be different with respect to different codeword $C'$ even though the Euclidean distance between codeword $C$ and $C'$ is still $d_{\text{min}}$. This is because $\Delta S(C, C')$ and $\Delta B(C, C')$ may not maintain the same values with respect to different $C'$.

From Equation (4.32), the lower bound of $P_{se}$ can be written as

$$P_G(se) = \sum_C P(C) P_G(se|C) = \sum_C P(C) P_G(ce|C) \Delta S_i / B$$

$$\geq \sum_C I(C) P(C) Q\left(d_{\text{min}}/2\sigma_n\right) \cdot \Delta S_i / B$$

(4.49)

where $I(C)$ is already defined in Equation (4.33) and $\Delta S_i$ is the pairwise number of symbol errors between source symbol blocks corresponding to codeword $C$ and the $i$-th $C'$. The Euclidean distance between $C$ and $C'$ is $d_{\text{min}}$. 
Because there are $N_{\text{min}}$ codewords that have pairwise Euclidean distance of $d_{\text{min}}$ to codeword $C$, an average number of symbol errors $\overline{\Delta S}$ can be defined as

$$\overline{\Delta S} = \frac{1}{N_{\text{min}}} \sum_{i=1}^{N_{\text{min}}} \Delta S_i$$

among those codewords $C'$ which have Euclidean distance $d_{\text{min}}$ to codeword $C$. Then the lower bound of probability of symbol error is

$$P_{se} \geq P_G(se) \geq \frac{N_{\text{min}} \cdot \overline{\Delta S}}{M \cdot B \cdot Q}\left(\frac{d_{\text{min}}}{2\sigma_n}\right).$$

(4.51)

In a similar manner, an average number of bit errors $\overline{\Delta B}$ can be defined as

$$\overline{\Delta B} = \frac{1}{N_{\text{min}}} \sum_{i=1}^{N_{\text{min}}} \Delta B_i.$$  

(4.52)

The lower bound for probability of bit error is thus

$$P_{be} \geq \frac{N_{\text{min}} \cdot \overline{\Delta B}}{M \cdot B \cdot k \cdot Q}\left(\frac{d_{\text{min}}}{2\sigma_n}\right),$$

(4.53)

where $k = \log_2 n$.

From Equation (4.42), the above lower bounds can be linked with $E_c/N_0$, $E_s/N_0$ and $E_b/N_0$ respectively. We have

$$P_{ce} \geq \frac{N_{\text{min}}}{M} Q\left(\frac{d_{\text{min}}}{\sqrt{E_c/N_0 \cdot 2B}}\right),$$

(4.54)

$$P_{se} \geq \frac{N_{\text{min}} \cdot \overline{\Delta S}}{M \cdot B \cdot Q}\left(\frac{d_{\text{min}}}{\sqrt{E_s/2N_0}}\right),$$

(4.55)

$$P_{be} \geq \frac{N_{\text{min}} \cdot \overline{\Delta B}}{M \cdot B \cdot k \cdot Q}\left(\frac{d_{\text{min}}}{\sqrt{E_b/2N_0}}\right).$$

(4.56)

For example, let $N = 8$, $n = 2$, $D = 3$, $B = 5$. When $n = 2$, $P_{se}$ and $P_{be}$ are equivalent to each other. Only the upper and lower bounds of $P_{ce}$ and $P_{be}$ are shown in Figure 45 and Figure 46 respectively. The bit error performance of BPSK is shown as a comparison in Figure 46.
Figure 45: The upper and lower bounds of $P_{ce}$ of (8,2,3,5) scheme.

Figure 46: The upper and lower bounds of $P_{be}$ of (8,2,3,5) scheme.
4.6 Asymptotic Upper Bound

In Section 4.2.2 and Section 4.5 we have derived union bounds for $P_{be}$, $P_{se}$ and $P_{ce}$. However, evaluation of these union bounds becomes particularly prohibitive when the size of the codebook $M$ is large. When $n$ or $B$ is large, the size of codewords $M = n^B$ can easily be an astronomical number. This will make the upper bounds too loose to be useful, especially at low signal-to-noise ratios, not to mention the amount of computation involved.

As we know, the system is designed to work at certain range of signal-to-noise ratios. We are interested in the system’s performance when $E_b/N_0$ is high enough so that the bit error performance is acceptable, rather than that at low $E_b/N_0$. Thus in this section, asymptotic upper bound is discussed. Note that the pairwise Euclidean distance is with respect to the reference codeword 1.

4.6.1 Asymptotic performance at high SNR

From Equation (4.24), we know that at a fixed SNR, the larger the pairwise Euclidean distance, the smaller the pairwise probability of error $P_{ce}(1, k)$. If $P_{ce}(1, k)$ is much smaller than the largest pairwise probability of error, we can simply ignore that item without affecting the bound significantly. In the following analysis, we examine how pairwise probability of error changes as pairwise Euclidean distance changes.

The error function erfc($\cdot$) has an inequality equation as follows:

$$\text{erfc}\left(\sqrt{\frac{x+y}{2}}\right) \leq \text{erfc}\left(\sqrt{\frac{x}{2}}\right) e^{-y/2}, \quad x \geq 0, y \geq 0. \quad (4.57)$$

The $Q$-function defined in Equation (4.23) relates to the error function erfc($\cdot$) by

$$Q(x) = \frac{1}{2} \text{erfc}\left(\frac{x}{\sqrt{2}}\right).$$
Thus we have an inequality for the $Q$-function

$$Q(\sqrt{x+y}) \leq Q(\sqrt{x})e^{-y/2}.$$ 

Then the pairwise probability of codeword error (Equation (4.44)) can be rewritten as

$$P_{ce}(1, k) = Q \left( \sqrt{\frac{d_k^2E_c}{2BN_0}} \right) = Q \left( \sqrt{\frac{[d_{min}^2 + (d_k^2 - d_{min}^2)]E_c}{2BN_0}} \right) \leq Q \left( d_{min} \sqrt{\frac{E_c}{2BN_0}} \right) \exp \left( -\frac{(d_k^2 - d_{min}^2)E_c}{4BN_0} \right)$$

(4.58)

where $d_{min}$ is the minimum pairwise Euclidean distance.

Similarly, Equation (4.45) and Equation (4.46) can be rewritten as

$$P_{se}(1, k) \leq \frac{\Delta S_k}{B} Q \left( d_{min} \sqrt{\frac{E_s}{2BN_0}} \right) \exp \left( -\frac{(d_k^2 - d_{min}^2)E_s}{4BN_0} \right)$$

(4.59)

$$P_{be}(1, k) \leq \frac{\Delta B_k}{B \cdot \log_2N} Q \left( d_{min} \sqrt{\frac{E_b \log_2^2n}{2BN_0}} \right) \exp \left( -\frac{(d_k^2 - d_{min}^2)E_b \log_2^2n}{4BN_0} \right)$$

(4.60)

We can separate $M-1$ codewords into $G$ groups, where each group includes those codewords that have the same pairwise Euclidean distance. Assume the number of codewords contained in each group is $N_t$ with Euclidean distance $d_t$, $t = 1, \ldots, G$ and $d_1 < d_2 < \ldots < d_G$. Let $N_{min}$ denote the number of codewords whose pairwise Euclidean distance is $d_{min}$, $N_1 = N_{min}$. Then upper bound for $P_{ce}$ in Equation (4.29)

$$\sum_{k=2}^{M} P_{ce}(1, k) = \sum_{t=1}^{G} N_t Q \left( d_t \sqrt{\frac{E_c}{2BN_0}} \right) \leq N_{min} Q \left( d_{min} \sqrt{\frac{E_c}{2BN_0}} \right) \cdot \left[ 1 + \sum_{t=2}^{G} \frac{N_t}{N_{min}} \exp \left( -\frac{(d_t^2 - d_{min}^2)E_c}{4BN_0} \right) \right]$$

(4.61)
where \( N_t \) satisfies

\[
\sum_{t=1}^{G} N_t = M - 1.
\]

At a fixed \( E_c/N_0 \), \( \exp\left(-\frac{(d_t^2-d_{\text{min}}^2)E_c}{4BN_0}\right) \) decreases exponentially as Euclidean distance \( d_t \) becomes greater. At a higher \( E_c/N_0 \), it decreases much faster. Even if \( N_t (2 \leq t \leq G) \) could be much larger than \( N_{\text{min}}, \frac{N_t}{N_{\text{min}}} \) is insignificant compared to the exponential term. Thus \( N_{\text{min}}Q(d_{\text{min}}\sqrt{E_c/(2BN_0)}) \) in Equation (4.61) will become a dominant term at a greater \( E_c/N_0 \). Therefore, at high signal-to-noise ratios, the performance is dominated by the minimum Euclidean distance \( d_{\text{min}} \), rather than \( d_{\text{free}} \), because \( d_{\text{min}} \leq d_{\text{free}} \).

Thus the asymptotic upper bound \( \text{AUB}_{ce} \) for \( P_{ce} \) is

\[
\text{AUB}_{ce} = N_{\text{min}}Q\left(d_{\text{min}}\sqrt{\frac{E_c}{2BN_0}}\right). \tag{4.62}
\]

We next examine the asymptotic upper bound \( \text{AUB}_{ce} \) when \( E_c/N_0 \) is small. Consider the worst case in which the decoder picks a codeword by random guessing. The probability of codeword error in this case is \( P_{ce} = (M - 1)/M \leq 1 \). The larger the code size \( M \), the closer \( P_{ce} \) is to unity. However, as the value of \( E_c/N_0 \) becomes smaller, the union bound will approach to \((M - 1)/2\), because \( P_{ce}(i,k) \leq 1/2 \). If \( M \gg 1 \), the union bound will be too loose to provide us any insight into the system if \( P_{ce}(i,k) \) is close to one half. However, in the worst case, the asymptotic upper bound approaches to \( N_{\text{min}}/2 \), which is larger than \((M - 1)/M\), but may be far smaller than \((M - 1)/2\). Thus, we conclude that the asymptotic upper bound \( \text{AUB}_{ce} \) is a valid but tighter bound even when \( E_c/N_0 \) is very small.

By similarly applying the inequality of \( Q \)-function, the following derivation is obtained for \( P_{se} \) and \( P_{be} \) respectively as
\[
\sum_{k=2}^{M} P_{se}(1, k) = \sum_{k=2}^{M} \frac{\Delta S_k}{B} Q \left( d_k \sqrt{\frac{E_s}{2N_0}} \right) \\
= \sum_{t=1}^{G} \frac{\Delta S_t}{B} N_t Q \left( d_t \sqrt{\frac{E_s}{2N_0}} \right) \\
\leq \frac{N_{\min} \Delta S_1}{B} Q \left( d_{\min} \sqrt{\frac{E_s}{2N_0}} \right) \cdot \\
\left[ 1 + \sum_{t=2}^{G} \frac{N_t \Delta S_t}{N_{\min} \Delta S_1} \exp \left( -\frac{(d_t^2 - d_{\min}^2)E_s}{4N_0} \right) \right] \tag{4.63}
\]

where \(\bar{\Delta S}_t\) is the average bit errors among those codewords that have the same distances of \(d_t\), and

\[
\bar{\Delta S}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} \Delta S_i.
\]

\[
\sum_{k=2}^{M} P_{be}(1, k) = \sum_{k=2}^{M} \frac{\Delta B_k}{B \log_2 n} Q \left( d_k \sqrt{\frac{E_b \log_2 n}{2N_0}} \right) \\
= \sum_{t=1}^{G} \frac{\Delta B_t}{B \log_2 n} N_t Q \left( d_t \sqrt{\frac{E_b \log_2 n}{2N_0}} \right) \\
\leq \frac{N_{\min} \Delta B_1}{B \log_2 n} Q \left( d_{\min} \sqrt{\frac{E_b \log_2 n}{2N_0}} \right) \cdot \\
\left[ 1 + \sum_{t=2}^{G} \frac{N_t \Delta B_t}{N_{\min} \Delta B_1} \exp \left( -\frac{(d_t^2 - d_{\min}^2)E_b \log_2 n}{4N_0} \right) \right] \tag{4.64}
\]

where \(\bar{\Delta B}_t\) is the average bit errors between codewords that have distances of \(d_t\), i.e.,

\[
\bar{\Delta B}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} \Delta B_i.
\]

By examining the asymptotic behavior of Equation (4.63) and Equation (4.64), we can obtain the asymptotic upper bounds for \(P_{se}\) and \(P_{be}\) as follows

\[
\text{AUB}_{se} = \frac{N_{\min} \cdot \Delta S}{B} Q \left( d_{\min} \sqrt{\frac{E_s}{2N_0}} \right) \tag{4.65}
\]

\[
\text{AUB}_{be} = \frac{N_{\min} \cdot \Delta B}{B \cdot \log_2 n} Q \left( d_{\min} \sqrt{\frac{E_b \log_2 n}{2N_0}} \right) \tag{4.66}
\]
where \( \text{AUB}_{se} \) and \( \text{AUB}_{be} \) denote the asymptotic upper bounds for \( P_{se} \) and \( P_{be} \) respectively. \( \bar{\Delta S} \) and \( \bar{\Delta B} \) are average symbol errors and average bit errors with respect to \( d_{\text{min}} \).

Let's examine the asymptotic performance for the case when \( (N,n,D,B) = (8,2,3,5) \). Since \( P_{be} \) and \( P_{se} \) are actually equivalent to each other when \( n = 2 \), hence only \( P_{ce} \) and \( P_{be} \) are examined. The union bound curves and asymptotic upper bound curves of \( P_{ce} \) and \( P_{be} \) are shown in Figure 47 and Figure 48 respectively. The curve for the union bound and the curve for the asymptotic upper bounds approach each other at high signal-to-noise ratios. The gap between two curves is more obvious when the signal-to-noise ratios are smaller. Moreover, if the code size is large, the gap could be widened depending on the value of \( N_{\text{min}} \) and \( d_{\text{min}} \), or the weight distribution of other Euclidean distances. This suggests that such an approximation may not be tight when signal-to-noise ratios are low or when the codeword size is large. However, at high signal-to-noise ratios, the two curves will virtually converge to each other.

The schemes in the following examples have larger code sizes. Figure 49 through Figure 52 depict the union bounds, lower bounds and asymptotic upper bounds of \( P_{be} \) of \((256,2,8,B)\) schemes with various \( B \). The union bound and lower bound are far apart in all these figures. This suggests that the bounds are loose, but the actual performance must fall in between these two bounds. One would notice that the union bound curve and the asymptotic upper bound curve are virtually identical in Figure 49, while in other three figures they are separated further. Even so, the two curves tend to converge to each other as \( E_b/N_0 \) increases. The speed of the convergence between these two curves is fastest if \( B = 13 \). This is because
Figure 47: The union and asymptotic upper bounds of $P_{ce}$ of (8,2,3,5) scheme.

Figure 48: The union and asymptotic upper bounds of $P_{be}$ of (8,2,3,5) scheme.
of the distance with the largest weight is farthest from \(d_{\text{min}}\) compared to the other situations. Their weight distribution is already recorded in Table 2. If \(B \geq D + 3\), \(N_{\text{min}}\) changes slightly. Because \(d_{\text{min}}\) stays the same, their asymptotic performance basically is unchanged. Hence, asymptotic performance can be treated as reference. Therefore the convergence speed reflects the improvement of the union bound as the weight distribution of Euclidean distance changes.

Figure 49: The union bound, lower bound and asymptotic upper bound of \(P_{\text{be}}\) of \((256,2,8,10)\) scheme.

Table 16 summarizes the results of the derivation of the upper bounds, lower bounds, and asymptotic upper bound for \(P_{\text{be}}, P_{\text{se}}\) and \(P_{\text{ce}}\) respectively.

4.6.2 On good code criteria

The criterion of selecting a good HDTMC scheme is based on the scheme’s minimum Euclidean distance for given trellis complexity (number of states) and transmitted bandwidth. Thus \(n, D\) and \(N\) are preset. Then the block length \(B\) becomes a variable.
Figure 50: The union bound, lower bound and asymptotic upper bound of $P_{be}$ of (256,2,8,11) scheme.

Figure 51: The union bound, lower bound and asymptotic upper bound of $P_{be}$ of (256,2,8,12) scheme.
Figure 52: The union bound, lower bound and asymptotic upper bound of $P_{be}$ of (256,2,8,13) scheme.

<table>
<thead>
<tr>
<th></th>
<th>Union Bound</th>
<th>Lower Bound</th>
<th>Asymptotic Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{ce}$</td>
<td>$\sum_{j=2}^{M} Q \left( d_j \sqrt{\frac{E_{s}}{2B N_0}} \right)$</td>
<td>$\frac{N_{\min}}{M} Q \left( \min d_{\min} \sqrt{\frac{E_{s}}{2B N_0}} \right)$</td>
<td>$\frac{N_{\min}}{B} Q \left( d_{\min} \sqrt{\frac{E_{s}}{2N_0}} \right)$</td>
</tr>
<tr>
<td>$P_{se}$</td>
<td>$\sum_{j=2}^{M} \frac{\Delta_{j}}{B} Q \left( d_j \sqrt{\frac{E_{s}}{2N_0}} \right)$</td>
<td>$\frac{N_{\min}}{M \cdot B} Q \left( d_{\min} \sqrt{\frac{E_{s}}{2N_0}} \right)$</td>
<td>$\frac{N_{\min}}{B} Q \left( d_{\min} \sqrt{\frac{E_{s}}{2N_0}} \right)$</td>
</tr>
<tr>
<td>$P_{be}$</td>
<td>$\sum_{j=2}^{M} \frac{\Delta B_j}{B \cdot \log_2 n} Q \left( d_j \sqrt{\frac{E_{s} \cdot \log_2 n}{2N_0}} \right)$</td>
<td>$\frac{N_{\min}}{M \cdot B \cdot \log_2 n} Q \left( d_{\min} \sqrt{\frac{E_{s} \cdot \log_2 n}{2N_0}} \right)$</td>
<td>$\frac{N_{\min}}{B \cdot \log_2 n} Q \left( d_{\min} \sqrt{\frac{E_{s} \cdot \log_2 n}{2N_0}} \right)$</td>
</tr>
</tbody>
</table>

Table 16: The upper bounds, lower bounds, and asymptotic upper bound of $P_{ce}$, $P_{se}$ and $P_{be}$. 
The first criterion is to select a scheme whose minimum Euclidean distance \( d_{\text{min}} \) achieves the maximum value. Then the number of codewords \( N_{\text{min}} \) at the minimum distance \( d_{\text{min}} \) should be minimized. The selection procedure can be further refined by considering the number of codewords \( N_2 \) at the second smallest Euclidean distance \( d_2 \), and so on. Therefore, to obtain an optimum combination of design parameters we need to vary \( B \) and then check the weight distribution of Euclidean distance.

As described in Section 4.1.3, the weight distribution of Euclidean distance is very interesting. For those HDTCM systems with binary sources, when \( B = D + 3 \), \( N_{\text{min}} \) is minimized. However, when \( B = D + 4 \), \( N_{\text{min}} \) goes up a little bit, but \( N_2 \), which is the weight of the second smallest distance, is minimized. This is a dilemma in which \( N_{\text{min}} \) cannot be minimized with other weights, such as \( N_2 \), at the same time. Thus an absolutely optimum scheme cannot be found. However, if we look at their asymptotic performances, the one with the lowest \( N_{\text{min}} \) has the best asymptotic performance. This is because both scheme have the same \( d_{\text{min}} \). Thus, we refer to those schemes with block length \( B = D + 3 \) as asymptotically good schemes.

If \( B \geq D + 3 \), the weight center (average weight) further shifts to the right as \( B \) goes higher, while \( N_{\text{min}} \) goes up slightly. Of course, \( d_{\text{min}} \) is unchanged. It is worthwhile to examine the percentage by which each distinct Euclidean distance contributes to the union bound as \( B \) changes. Here we examine \( P_{be} \) only. The weight of a distinct Euclidean distance \( d_t, t = 1, 2, \ldots, G \), is given by the following percentage formulas pertaining to Equation (4.64),

\[
\text{Percentage}_t = \frac{\Delta B_t}{B \log_2 n} N_t Q \left( d_t \sqrt{\frac{E_b \log_2 n}{2 \cdot N_0}} \right). \tag{4.67}
\]
Figure 53 shows the percentage contributions of (16,2,4,7) scheme. The $d_{\text{min}}$ curve gradually dominates the sum as $E_b/N_0$ becomes larger. This matches our analysis of asymptotic performance. But we see the dominance of the $d^2 = 14$ curve when $E_b/N_0$ falls in the range of -2 dB to 7 dB. When $B = 8$, the percentage curve of the $d^2 = 14$ and curve $d_{\text{min}}^2$ shifts to the left as shown in Figure 54. As $B$ further goes up, such shifting is persistent as revealed in Figure 55 and Figure 56 too. Comparing Figure 53 with Figure 54, we notice that the $d_{\text{min}}$ curve and the $d^2 = 14$ curve shift to the left by about 2 dB respectively when $B$ changes from seven to eight. However, if $B$ changes from eight to nine or nine to ten, such shift becomes less significant. This can be explained by the negative exponential term in Equation (4.64). As $d_t$ becomes larger, even some weight $N_t$ is large, its contribution to the union bound becomes smaller.

As we stated above, we want to minimize weight $N_t$ in the order of $N_1$ or $N_{\text{min}}$ to $N_G$. The dilemma is when $B = D + 3$, $N_{\text{min}}$ is minimized, but $N_2$ still has the largest value; when $B = D + 3$, $N_2$ is minimized but $N_{\text{min}}$ becomes a little bit larger. We have declared that $B = D + 3$ cases are asymptotically good, because at high signal to noise ratios, the performance is determined by $N_{\text{min}}$ if $d_{\text{min}}$ stays the same. But if the desired operating error probability is $10^{-5}$ or lower, can we still claimed that the optimum selection of $B$ is $D + 3$?

To answer this question we have to resort to simulations. A simulation model is shown in Figure 57 in three parts. The Monte Carlo trial method is used to simulate bit error probabilities. First we need to model each of them respectively. A test codeword is uniformly sampled from the codebook during trials.
Figure 53: The percentage contribution to the union bound of $P_{be}$ due to each distinct Euclidean distance of (16,2,4,7) scheme.

Figure 54: The percentage contribution to the union bound of $P_{be}$ due to each distinct Euclidean distance of (16,2,4,8) scheme.
Figure 55: The percentage contribution to the union bound of $P_{be}$ due to each distinct Euclidean distance of (16,2,4,9) scheme.

Figure 56: The percentage contribution to the union bound of $P_{be}$ due to each distinct Euclidean distance of (16,2,4,10) scheme.
In order to speed up simulation, we construct a codebook for the encoder beforehand so that a random codeword can be generated by uniformly generating an index, ranging from 1 to $n^B$. This is the encoder part. As for the channel, AWGN channel is considered. No fading, no jamming and other interferences. The Gaussian channel is already well understood. The relationship between noise standard deviation $\sigma_n$ and signal to noise ratios $E_b/N_0$ is shown in Equation (4.41). As for the receiver, assume perfect reception is accomplished. Thus the received signal is the transmitted codeword corrupted by Gaussian noise with respect to certain signal-to-noise ratio. An optimum maximum likelihood decoder is used. Its algorithm is presented in Section 3.2.2 and Section 3.2.3. By comparing the decoded word and the source, error probabilities $P_{be}$, $P_{se}$ and $P_{cc}$ can be obtained.

At high $E_b/N_0$, $P_{be}$ tends to be small, while at low $E_b/N_0$, $P_{be}$ tends to be large. According to the Law of Large Numbers, the more trials we conduct, the more accurate the result will be. However, because of the limitation of computing power and resources, a proper trial number should be chosen for each $E_b/N_0$ point. According to experience, usually the trial number is appropriate if there are a few hundred errors for a specific $E_b/N_0$. Thus, instead of assigning the same number to the trials of different $E_b/N_0$ points, different numbers of trials can be selected at different ranges of $P_{be}$ so as to make the simulation faster.
The simulation curves are shown in Figure 58 when $N = 16$, $n = 2$, $D = 4$ and $B = 6, 7, 8, 9, 10$ respectively. All these five schemes have the same $d_{\text{min}}$, but different weight distribution. The improvement of $P_{\text{be}}$ is noticed as $B$ becomes larger. The (16,2,4,7) scheme obviously improves over the (16,2,4,6) scheme because of the reduction in $N_{\text{min}}$. The (16,2,4,8) scheme is better than (16,2,4,7) scheme because of the largest weight shifts to $N_3$ although $N_{\text{min}}$ increases by one. However, as the largest weight further shifts to the right, its influence on the performance is weakened because of the negative exponential growth effect. Therefore, in the actual implementation, one may prefer the choice of $B = D + 4$ if we do not need to worry about computation or processing time.

![Figure 58: The simulation result of $P_{\text{be}}$ of (16,2,4,B) schemes with $B = 6, 7, 8, 9, 10$ respectively.](image-url)
4.7 Coding Gain of the HDTCM Scheme

In this section, we derive the coding gain of the HDTCM scheme over a conventional spread spectrum communication system. First of all, comparison between four spread spectrum systems is presented. Next, the formula of asymptotic coding gain is given.

4.7.1 The comparison of four spread spectrum systems

In conventional spread spectrum systems, the PN sequence is independent of the source. The spreading is accomplished by multiplying the output of the modulator by a PN sequence in DS systems. In FH systems, the PN sequence is used to switch the carrier frequency which has the same spreading effect as that in DS systems. In the following analysis, the process with such spreading effect is termed as PN spreading. The PN spreading can be treated as a process independent of modulation.

As we know, purely expanding the transmitted bandwidth by multiplying a PN sequence will not improve a system’s ability to combat the AWGN noise. This means that no performance improvement or no energy efficiency gain can be achieved. The “waterfall” performance curve of the system does not move toward the Shannon limit. However, the bandwidth expansion brings the system a processing gain, and thus provides a spread spectrum system with high immunity to narrow-band interfering signals or to other spread spectrum signals occupying the same frequency band. The processing gain is defined as the ratio of expanded transmitted bandwidth over the information bandwidth.

Figure 59 depicts four spread spectrum systems. System A is an uncoded conventional spread spectrum communication system, which contains only modulation and spreading processes. Because of the PN spreading, the system achieves a processing
Figure 59: The comparison of system’s performance: (A) an uncoded conventional spread spectrum communication system; (B) a coded conventional spread spectrum communication system; (C) a combined coding and modulation — TCM scheme; (D) a high dimensional trellis-coded modulation spread spectrum communication system.
gain denoted by $PG_A$. However, System A has no coding gain, $CG_A = 0$ dB, because channel coding is not employed.

System B is a coded conventional spread spectrum communication system. It is similar to System A, because both systems have modulation and spreading processes, but System B has one more separate process, i.e., channel coding. The channel coding is independent of the modulation in System B. The source is first channel-encoded according to various code rate specifications\textsuperscript{11}, and then is modulated and finally is spread right before transmission. Because of the channel coding and PN spreading, System B not only has a processing gain of $PG_B$, but also achieves a coding gain of $CG_B$ ($CG_B \neq 0$ dB).

Current TCM technique is used in System C. Before PN spreading, the source is coded and modulated in one single process. The system is able to obtain a processing gain of $PG_C$ because of spreading and a coding gain of $CG_C$ ($CG_C \neq 0$ dB) because of the deployment of TCM.

System D employs the proposed high dimensional trellis-coded modulation scheme. As we know, HDTCM integrates a block code with state-permuted trellis structure and an expanded high-dimensional constellation in a way somewhat similar to the conventional TCM, but actually quite different in terms of their goals and implementation. The goal of TCM is to achieve coding gain while incurring no bandwidth expansion. However, the goal of HDTCM is to intentionally spread the transmitted signals while achieving coding gain. Similar to current TCM, HDTCM achieves coding gain by expanding the signal constellation. Moreover, the expanded constellation simultaneously provides a way of accomplishing spreading. TCM uses a bandwidth

\textsuperscript{11}The coding can either be convolutional coding or block coding.
efficient two-dimensional signal constellation, such as MPSK and QAM. There is a high dimension signal constellation concept in recent modem protocols which consists of two two-dimensional signal planes, thus forming a four-dimensional constellation. However, the signal constellation in HDTCM is “truly” of high dimension, such as bi-orthogonal signaling. Moreover, the difference between their trellis structures is obvious. If we view PN spreading as an independent process, the HDTCM scheme unifies not only coding and modulation, but also spreading into one single process. This is an important insight. Without this insight, proper comparison of HDTCM systems would not have been possible. Assume system D is able to achieve a processing gain of $PG_D$ and a coding gain of $CG_D$ ($CG_D \neq 0$ dB).

Next, we want to compare these four systems in terms of their processing gains and coding gains. Let’s further assume that the transmitted signals in the above four systems use the same amount of bandwidth so as to put them on an equal footing. The transmitted bandwidth is required be $\alpha$ times larger than the information bandwidth in all systems.

For System A, the bandwidth expansion ratio is $\alpha$. Thus its processing gain $PG_A$ is given by

$$PG_A = 10 \log_{10} \alpha \text{ dB.} \quad (4.68)$$

In system B, channel coding and modulation are two separate processes, channel coding is used to increase the redundancy of the transmitted signal. To achieve coding gain it is desired to decrease the code rate. However, its by-product is to cause bandwidth expansion. Assume its code rate is $1/\beta$, then the bandwidth expansion ratio of the channel-encoder is $\beta$. Thus the amount of bandwidth expansion that can be utilized by spreading is reduced to $\alpha/\beta$. The processing gain is determined
by the ratio of the bandwidth of the spread signal (transmitted bandwidth) and the bandwidth of the signal right before spreading. The amount of bandwidth expansion resulting from the channel encoding will not provide the system any processing gain. This portion of bandwidth expansion is used to provide coding gain instead. Thus the processing gain of System B is given by

\[ PG_B = 10 \log_{10} \frac{\alpha}{\beta} = PG_A - 10 \log_{10} \beta \quad \text{dB}. \]  

(4.69)

This means that System B achieves the coding gain \( CG_B \) at the expense of the reduction in \( PG_B \), if it is compared to the uncoded System A.

In System C, because of the deployment of TCM, the coding gain is achieved by the expansion of signal constellation without expanding bandwidth. The constellation can be either MPSK or QAM. Because of its spectral efficiency, QAM is widely used, especially in the telephone channels. However, RF channels are fading channels and the decision of QAM signals is sensitive to the magnitude of the received signal, in addition to the phase. QAM is usually not preferred in spread spectrum communication systems. If a MPSK or MFSK constellation is used, the average energies spent to transmit with uncoded and coded transmission are unchanged, which means there is no “energy loss”.

Because bandwidth expansion is not incurred, the bandwidth expansion ratio for spreading is still \( \alpha \). Thus its processing gain is

\[ PG_C = PG_A = 10 \log_{10} \alpha \quad \text{dB}. \]  

(4.70)

Notice that in System C the spreading process is independent of the TCM process.

The coding gain in System D is achieved by integrating coding, modulation and spreading together as one entity. The spreading is accomplished by the use of high
dimensional signal constellation which results in the bandwidth expansion that is required by the spreading. The scheme transforms an \(n\)-ary source symbol into an \(N\)-dimensional channel symbol. For bi-orthogonal signaling, \(N = \alpha\). Thus its processing gain is given by

\[
P_{G_D} = 10 \log_{10} \left( \frac{N}{\log_2 n} \right) = 10 \log_{10} \left( \frac{\alpha}{\log_2 n} \right) \text{ dB.} \tag{4.71}
\]

The coding gains and the processing gains of these four systems are summarized in Table 17, where \(\alpha\) is the bandwidth ratio of transmitted signal to the uncoded source. The above analysis is intended to identify the difference between these systems, rather than to determine which one is better or best. Their processing gains can be compared as stated above. However, it is not fair to determine which system is better by simply looking at their coding gain provided in the references. First, we need to be aware of their corresponding reference systems. If reference systems are different, we need to come up with a common reference system for all of them. Secondly, the implementation of each scheme is different, so it is not easy to come up a common reference system. Also, the complexity, such as the number of trellis states, of each system needs to be taken into account as necessary.

In the HDTCM systems, the bandwidth expansion ratio is determined by the size of the signal constellation. As we have seen, it is less flexible for us to choose a value. However, \(N\) can be chosen first and then the transmitted bandwidth of the reference system — System A can be determined based on that. We can also make sure that the transmitted bandwidth of System B and C is the same as A. However, if we further require that the number of states \(S\) of System B, C, D be the same such that they have equal complexity, then we can not come up a proper HDTCM scheme for
comparison, because of the limited flexibility of choosing those design parameters. By the way, the complexity of convolutional encoder is determined by the constraint length.

<table>
<thead>
<tr>
<th>Gains (dB)</th>
<th>System A</th>
<th>System B</th>
<th>System C</th>
<th>System D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Gain</td>
<td>$10 \log_{10} \alpha$</td>
<td>$10 \log_{10}(\alpha/\beta)$</td>
<td>$10 \log_{10} \alpha$</td>
<td>$10 \log_{10}(\alpha/\log_2 n)$</td>
</tr>
<tr>
<td>Coding Gain</td>
<td>0</td>
<td>$CG_B$</td>
<td>$CG_C$</td>
<td>$CG_D$</td>
</tr>
</tbody>
</table>

Table 17: The coding gains and processing gains of four spread spectrum systems.

4.7.2 The coding gain of HDTCM

To measure the coding gain $CD_D$, we need to come up a reference system for the HDTCM system to make sure that the comparison is fair. Because the high dimensional trellis-coded modulation system is one of spread spectrum systems, the reference system should be a spread spectrum system too, and it should be uncoded. Thus System A is chosen as the reference system.

The footing of comparison is that two systems use the same amount of transmitted bandwidth with respect to the same source. The transmitted bandwidth must be $N$ times greater than that of the uncoded source for bi-orthogonal signaling. Also, we make sure that the average energies spent to transmit with coded and uncoded transmission are the same. In HDTCM, although the signal constellation is expanded, the average energy expenditure is not increased. Hence there is no energy loss. The bit rates of both systems are the same because of the same input.

In communications, the coding gain is referred to the savings in $E_b/N_0$ over uncoded signaling with respect to the same probability of bit error, as shown in Fig-
ure 60, where $PE_1$ is the uncoded error performance and $PE_2$ is the coded error performance. Coding gain must always be specified at some operating error probability depending on the applications, typically $10^{-5}$. Such coding gain translates into an effective improvement in the communication link — allowing either a reduction of transmitter power or antenna gain or an increase in receiver noise level or an increase in the bit rate for a given power. It is usually expressed in decibels.

Assume the uncoded error performance curve $PE_1$ is quantified by $P_{be1} = N_1 Q(\alpha)$, the coded error performance curve $PE_2$ by $P_{be2} = N_2 Q(\beta)$, where $N_1$ and $N_2$ are multipliers. At operating error probability $P_{be1} = P_{be2} = 10^{-5}$, the values of $E_b/N_0$ to achieve that performance are $\alpha = x_1$, $\beta = x_2$ for curve $PE_1$ and curve $PE_2$ respectively. If $x_1$ and $x_2$ are not expressed in decibels, then we assume $x_2 = \gamma x_1$. 
Thus $10 \log_{10} \gamma$ (dB) is the coding gain we want to calculate. In order to calculate the coding gain, we need to solve the following equation

$$N_1 Q(x_1) = N_2 Q(x_2). \quad (4.72)$$

This seems impossible to solve according to the definition of the $Q$-function. However, $Q$-function has the following exact expansion [40]

$$Q(x) = \frac{Z(x)}{x} \left\{ 1 - \frac{1}{x^2} + \frac{1 \cdot 3}{x^4} + \ldots + \frac{(-1)^n 1 \cdot 3 \ldots (2n - 1)}{x^{2n}} \right\} + R_n, \quad (4.73)$$

where

$$Z(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right),$$

and $R_n$ is the residual term,

$$R_n = (-1)^{n+1} \cdot 3 \ldots (2n + 1) \int_x^\infty \frac{Z(t)}{t^{2n+2}} dt.$$

For easy analysis, the following approximation can be obtained

$$Q(x) \approx \frac{1}{\sqrt{2\pi x}} \exp\left(-\frac{x^2}{2}\right), \quad x > 0.$$ 

The bound is tighter as $x$ increases.

Thus, Equation (4.72) becomes

$$N_1 x_2 \exp\left(-\frac{x_1^2}{2}\right) = N_2 x_1 \exp\left(-\frac{x_2^2}{2}\right),$$

$$\left(1 - \gamma^2\right)x_1^2 = 2\ln\left(\frac{N_1}{N_2}\right) + \ln \gamma). \quad (4.74)$$

The task of obtaining the coding gain is reduced to solving Equation (4.74) analytically for $\gamma$, which is not an easy task. However, we may numerically plot curves for the above equation with the help of a computer, and then locate a cross intersection
point from the graph. Hence $\gamma$ can be obtained numerically. Equation (4.74) also shows that the coding gain is dependent upon the operating error probability, since $x_1$ is determined by the specified error probability. Because of the difficulty of obtaining coding gain, the asymptotic coding gain [11, pp. 521] is introduced and the derivation of asymptotic coding gain of HDTCM systems is presented in the following paragraphs.

As we have derived, the asymptotic performance of HDTCM scheme at high signal-to-noise ratios is given by

$$P_{be_{HDTCM}} = A_{\min} Q\left(d_{\min} \sqrt{\frac{E_b \log_2 n}{2N_0}} \right),$$

(4.75)

where the multiplier

$$A_{\min} = \frac{N_{\min} \cdot \overline{AB}}{B \cdot \log_2 n}.$$ 

![Diagram](image.png)

Figure 61: (a) An uncoded MPSK spread spectrum system; (b) an uncoded MFSK spread spectrum system; (c) an HDTCM spread spectrum system.
System A can employ two different modulation schemes, either MPSK or MFSK, as shown in Figure 61. If the modulation is MPSK, the bandwidth of the signal before and after modulation keeps unchanged. Thus the processing gain for MPSK-modulated conventional uncoded spread spectrum systems is given by

$$PG_{A_{MPSK}} = 10 \log_{10} N \text{ dB.}$$  \hspace{1cm} (4.76)

The probability of bit error of MPSK is given by the following approximation [11]

$$P_{be_{uncoded-MPSK}} \approx \frac{2}{k} Q \left( \sqrt{\frac{2E_b \log_2 n}{N_0} \sin^2 \left( \frac{\pi}{n} \right)} \right). \hspace{1cm} (4.77)$$

As we know, Q-function has the following upper bound [37]

$$Q(x) = \frac{1}{2} e^{-\frac{x^2}{2}}, \quad x > 3. \hspace{1cm} (4.78)$$

The bound is looser but is more analytically convenient. If we replace the Q-function in Equation (4.75) and Equation (4.77) respectively by the above exponential bound and neglect the multipliers, we find that

$$\lim_{E_b/N_0 \to \infty} \ln \frac{P_{be_{HDTM}}}{P_{be_{uncoded-MPSK}}} = \left( \frac{d_{min}^2}{4 \sin^2 \frac{\pi}{n}} \right) \left( \frac{2E_b \log_2 n}{N_0} \sin^2 \left( \frac{\pi}{n} \right) \right). \hspace{1cm} (4.79)$$

We see that the relative energy efficiency between Equation (4.75) and Equation (4.77) is governed by the Q-function argument $d_{min}^2/(4\sin^2(\pi/n))$. The quantity, often converted to decibels, is frequently dubbed the "asymptotic coding gain," for it represents the relative energy efficiency at large SNR, where the multiplier coefficients are relatively insignificant and where the asymptotic upper bound becomes an accurate estimation of the error performance. Denote the asymptotic coding gain by $\gamma_{asym}$. The coding gain of HDTMC spread spectrum systems over MPSK-modulated conventional system is

$$\gamma_{asym_{MPSK}} = 10 \log_{10} \left( \frac{d_{min}^2}{4 \sin^2 \frac{\pi}{n}} \right) \text{ dB.} \hspace{1cm} (4.80)$$
If System A is using MFSK modulation, the signal bandwidth is expanded by $n$ times after modulation. The bandwidth expansion ratio that can be utilized by PN spreading is reduced to $N/n$. The processing gain of such MFSK modulated spread spectrum systems is thus given by

$$PG_{AMFSK} = 10 \log_{10} \frac{N}{n} \text{ dB.} \quad (4.81)$$

The probability of bit error of MFSK is bounded by [11]

$$P_{b\text{ uncoded-MFSK}} < \frac{n}{2} Q \left( \sqrt{\frac{E_b}{N_0} \log_2 n} \right). \quad (4.82)$$

In a similar way, the asymptotic coding gain over MFSK-modulated reference system is given by

$$\gamma_{asym_{MFSK}} = \log_{10} \left( \frac{d^2_{\min}}{2} \right) \text{ dB.} \quad (4.83)$$

The above result is summarized in Table 18.

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Asymptotic Coding Gain $\gamma_{asym}$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPSK</td>
<td>$10 \log_{10} \frac{d^2_{\min}}{4 \sin^2(\pi/n)}$</td>
</tr>
<tr>
<td>MFSK</td>
<td>$10 \log_{10} \frac{d^2_{\min}}{2}$</td>
</tr>
</tbody>
</table>

Table 18: The asymptotic coding gains of HDTCM spread spectrum systems over MPSK-, MFSK-modulated uncoded conventional spread spectrum systems.

If $n = 2$, the reference system is either BPSK-modulated or BFSK-modulated, the coding gain formula are reduced to

$$\gamma_{asym_{BPSK}} = 10 \log_{10} \left( \frac{d^2_{\min}}{4} \right) \text{ dB,} \quad (4.84)$$

$$\gamma_{asym_{BFSK}} = 10 \log_{10} \left( \frac{d^2_{\min}}{2} \right) \text{ dB} \quad (4.85)$$
respectively. Assume the block length $B \geq D + 2$, then the asymptotic coding gains achieved are listed in Table 19 for various $N$ and $D$. It is also shown in Figure 62 for visual illustration. The best asymptotic coding gain listed in the table is 7.78 dB with respect to BPSK when $N = 1024$ and $D = 10$. To achieve an operating error probability of $10^{-5}$, the BPSK-modulated spread spectrum system requires $E_b/N_0 \approx 9.6$ dB. This means that for a HDTCM system ($N=512$, $D=9$, $n = 2$) to operate at the same error probability, it requires $E_b/N_0 = 2.2$ dB. The specific example above shows that the performance of the proposed scheme is superior to the uncoded BPSK spread spectrum system. We notice that the asymptotic coding gain should be increased by 3.01 dB if the modulation of the reference system changes from BPSK to BFSK.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$D$</th>
<th>$\gamma_{\text{asymBPSK}}$ (dB)</th>
<th>$\gamma_{\text{asymBFSK}}$ (dB)</th>
<th>PG (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>3.01</td>
<td>6.02</td>
<td>6.02</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>3.98</td>
<td>6.99</td>
<td>9.03</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>4.77</td>
<td>7.78</td>
<td>12.04</td>
</tr>
<tr>
<td>32</td>
<td>5</td>
<td>5.44</td>
<td>8.45</td>
<td>15.05</td>
</tr>
<tr>
<td>64</td>
<td>6</td>
<td>6.02</td>
<td>9.03</td>
<td>18.06</td>
</tr>
<tr>
<td>128</td>
<td>7</td>
<td>6.53</td>
<td>9.54</td>
<td>21.07</td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>6.99</td>
<td>10.00</td>
<td>24.08</td>
</tr>
<tr>
<td>512</td>
<td>9</td>
<td>7.40</td>
<td>10.41</td>
<td>27.09</td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>7.78</td>
<td>10.79</td>
<td>30.10</td>
</tr>
</tbody>
</table>

Table 19: The asymptotic coding gain of HDTCM spread spectrum systems over BPSK or BFSK uncoded conventional spread spectrum systems when $n = 2$ and $B \geq D + 2$ and the processing gains of HDTCM systems.

If we calculate the difference between two adjacent coding gains, we have {.97 .79 .67 .58 .51 .46 .41}. We see that the pace of growth in asymptotic coding gain slows down as $N$ becomes larger and larger. When $N$ changes from four to eight, the
Asymptotic coding gain of HDTCM over reference systems

Figure 62: The asymptotic coding gain $\gamma_{\text{asym}}$ of HDTCM when $n = 2$.

Asymptotic coding gain of HDTCM over reference systems

Figure 62: The asymptotic coding gain $\gamma_{\text{asym}}$ of HDTCM when $n = 2$.

As for their processing gain, the HDTCM system and the BPSK-modulated system have the same processing gain, which is $3 \text{ dB}$ larger than that of the QFSK-modulated system. Their expressions are listed in Table 20.

If $n = 4$, the reference system is either QPSK-modulated or QFSK-modulated,
Figure 63: $\gamma_{asym_{BPSK}}$ vs. $d_{min}^2$.

<table>
<thead>
<tr>
<th>HDTCM</th>
<th>BPSK-modulated (dB)</th>
<th>BFSK-modulated (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10 \log_{10} N$</td>
<td>$10 \log_{10} N$</td>
<td>$10 \log_{10} N/2$</td>
</tr>
</tbody>
</table>

Table 20: The processing gains of HDTM systems, BPSK- and BFSK-modulated systems when $n = 2$. 
the coding gain formula is reduced to

\[
\gamma_{asym_{QPSK}} = 10 \log_{10} \left( \frac{d^2_{\text{min}}}{2} \right) \text{ dB}, \quad (4.86)
\]

\[
\gamma_{asym_{QFSK}} = 10 \log_{10} \left( \frac{d^2_{\text{min}}}{2} \right) \text{ dB}. \quad (4.87)
\]

Notice that \( \gamma_{asym_{QPSK}} = \gamma_{asym_{QFSK}} \). The choice of the modulation of the reference system does not matter in this case. However, the HDTCM system obtains more processing gain than that of the QFSK-modulated reference system. The QPSK-modulated reference system achieves the largest processing gain. The expression of processing gain is listed in Table 21. If the block length \( B \geq D + 1 \), then the asymptotic coding gains of some HDTCM schemes are listed in Table 22. Visual illustration of asymptotic coding gains is shown in Figure 64. It also shows negatively exponential growth characteristics. This conforms to the Shannon theorem.

<table>
<thead>
<tr>
<th>HDTCM</th>
<th>QPSK-modulated (dB)</th>
<th>QFSK-modulated (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 \log_{10} \frac{N}{2}</td>
<td>10 \log_{10} N</td>
<td>10 \log_{10} \frac{N}{4}</td>
</tr>
</tbody>
</table>

Table 21: The processing gains of HDTCM systems, QPSK- and QFSK-modulated systems when \( n = 4 \).

<table>
<thead>
<tr>
<th>( N )</th>
<th>( D )</th>
<th>( d^2_{\text{min}} )</th>
<th>( \gamma_{asym_{QPSK}} ) (dB)</th>
<th>( \gamma_{asym_{QFSK}} ) (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>2</td>
<td>6</td>
<td>4.77</td>
<td>4.77</td>
</tr>
<tr>
<td>128</td>
<td>3</td>
<td>8</td>
<td>6.02</td>
<td>6.02</td>
</tr>
<tr>
<td>1024</td>
<td>5</td>
<td>12</td>
<td>6.99</td>
<td>6.99</td>
</tr>
</tbody>
</table>

Table 22: The asymptotic coding gain of HDTCM spread spectrum systems over QPSK- or QFSK-modulated uncoded conventional spread spectrum systems when \( n = 4 \) and \( B \geq D + 1 \).
To compare the cases when $n = 2$ and $n = 4$ respectively, we need to look for a common ground for both systems. This is difficult, however. First, two systems have two different sources because of different bit rates. Next, although we can have two systems have the same amount of transmitted bandwidth, they will have different trellis complexity, because the number of states $S$ depends upon the trellis depth $D$ and $n$ through $S = n^D$. Third, the two reference systems are different, one is BPSK/BFSK modulated, the other is QPSK/QFSK. We cannot compare their improvement by simply looking at their asymptotic coding gains. However, we are able to conclude that for an $n = 4$ system to have a decent coding gain, the amount of bandwidth needed is enormous.
4.8 Trade-offs

The section is devoted to the discussion of some trade-offs of HDTM systems. First we analyze the influence of the trellis structure on the $d_{\text{min}}$ of HDTM schemes. Then we explore how to optimally choose those design parameters. Finally, we examine the performance if a high dimensional orthogonal constellation is used.

4.8.1 Trellis structure — with or without parallel transitions

If a trellis has parallel transitions, there are more than one branch connection between any two states. By keeping $n$ constant, the complexity (number of states) of a trellis with parallel transitions is less than the one without parallel transitions. This is because the number of diverging branches of any state is equal to the size of the input alphabet $n$ and more branches can bridge any two states if parallel transitions are allowed. The concept of trellis depth does not apply to the trellis structures with parallel transitions. The bit rate of the scheme with parallel transitions can be improved, because more bits ($k = \log_2 n$) can be processed at a time. In general, increasing $n$ will make the HDTM scheme more complex.

As we know, the asymptotic upper bound of HDTM is given by

$$\text{AUB}_{\text{be}} = \frac{N_{\text{min}} \cdot \Delta B}{B \cdot \log_2 n} \left( d_{\text{min}} \sqrt{\frac{E_b \log_2 n}{2N_0}} \right)$$

The increase in $n$ improves the error performance, if $d_{\text{min}}$ is kept constant. However, it has been found that when $n = 4$, $d_{\text{min}}$ drops compared to the schemes with the same trellis depth but $n = 2$, despite the much more complicated trellis structure and inflexible options of $N$. We next examine the effect of $d_{\text{min}}$ if we use parallel transition to reduce the complexity of the trellis when we increase $n$. Consider an ideal situation
where $d_{\text{min}} = d_{\text{free}}$. This means that with parallel transitions, $d_{\text{min}}$ is governed by the minimum distance between the signals of those parallel branches. If bi-orthogonal signaling is used, the maximum value of $d_{\text{min}}$ is equal to $\sqrt{(1 - (-1))^2} = 2$. However, for the scheme without parallel transitions, $d_{\text{min}}$ can easily be made greater than two, as listed in Table 2. Therefore the problem of choosing proper trellis structure is to do with the trade-off between $d_{\text{min}}$ and $n$. This is summarized as follows

1. **Parallel transitions**: $n$ can be made larger without making the number of states too huge. However, $d_{\text{min}}$ is limited by the minimum distance between the signals of parallel branches. The error performance is worse, because $d_{\text{min}}$ is too small to be compensated by the increment of $n$.

2. **No parallel transitions**: $n$ can also be larger, but the number of states grows exponentially because of $S = n^D$. The price is exponentially increasing complexity. $d_{\text{min}}$ could become larger too, but it is limited by the nature of bi-orthogonal signaling. The advantages of bi-orthogonal signaling gradually diminish as $n$ becomes larger.

### 4.8.2 The selection of $B$

From Section 4.1.2, we know that the free Euclidean distance depends upon the error events with minimum length $D + 1$. For the same $N$, $n$, $D$, if we want to find out $d_{\text{free}}$, we only need to examine $B = D + 1$ case. If the block length $B > D + 1$, the value of $d_{\text{free}}$ will not be improved because of the longer block.

It is always true that $d_{\text{min}} \leq d_{\text{free}}$ for any trellis. Ideally $d_{\text{min}} = d_{\text{free}}$. This means that when $B > D + 1$, the increase in $B$ will not result in the improvement of $d_{\text{min}}$. However, Table 2 tells us that when $B$ exceeds some threshold ($D + 2$ for the table),
the center of distance weight distribution shifts to the right as $B$ increases. The right-shifting of the center is characterized by the larger average squared Euclidean distance. However, the asymptotic performance is not improved. If we disregard such small change of $N_{\text{min}}$, the right-shifting of the weight center is a good thing. This is because the influence of those distances other than $d_{\text{min}}$ gradually deteriorate, especially the ones with the largest weight. It is desired that that distance value to be farther way from $d_{\text{min}}$.

Suppose such slight improvement is what we desire, then the price is exponential growth of computation in the decoder. The margin of improvement is reduced if $B$ is further increased. If computation resources are abundant, the larger $B$ is of course recommended. Otherwise, such trade-off should be taken into account.

### 4.8.3 Orthogonal signal constellation

In this dissertation, bi-orthogonal signaling is chosen as a high dimensional signal constellation. Suppose orthogonal signaling is used instead, what is the difference of their performance? Assume the trellis does not use parallel transitions.

This new scheme can still be characterized by four parameters, $(N, n, D, B)$ with the same definition as that in bi-orthogonal signaling case. However, the size of signaling set is $N$ instead, which is equal to the dimension of constellation. Similarly, for optimal transmission symbol table, the optimum value of the dimensionality $N_{\text{opt}}$ should satisfy

$$N_{\text{opt}} = n^{D+1}.$$  \hfill (4.88)

This means that if a HDTCM scheme uses orthogonal constellation, the bandwidth that it requires is twice as large as that of bi-orthogonal case for the same $n$ and $D$. 
Because each channel symbol is orthogonal to each other, the transmission symbol table is easy to generate. All the legitimate codewords are distributed in a hypersphere. The distance between any two codewords are always equal to its minimum distance $d_{\text{min}}$. The value of $d_{\text{min}}$ is determined by $B$. To make $d_{\text{min}}$ reach to its maximum value, we need to determine the optimum value of $B$. Let’s try to visualize all the codewords in a way shown in Section 3.2.2. The optimum block length $B$ is then governed by

$$n^B = N$$

so that in every column of the codebook, no duplication of channel symbol occurs. Thus the optimum block length is $B_{\text{opt}} = D + 1$. The optimum minimum squared Euclidean distance is hence $d^2_{\text{min, opt}} = 2(D + 1)$.

If two schemes, one using bi-orthogonal signaling, the other using orthogonal signaling, have the same sources (same $n$) and use the same bandwidth (same $N$), their performance comparison between them is done as follows. From Equation (4.7) and Equation (4.88), we have $N_{\text{ortho}} = 2N_{\text{bi-ortho}}^{12}$. Then the optimum trellis depth of the one with orthogonal signaling is

$$D_{\text{ortho}} = \log_n N_{\text{ortho}} - 1,$$

and for the one with bi-orthogonal signaling is obtained from Equation (4.7)

$$D_{\text{bi-ortho}} = \log_n (2N_{\text{bi-ortho}}) - 1.$$  

The minimum squared Euclidean distances are thus

$$d^2_{\text{min, ortho, opt}} = 2\log_n N_{\text{ortho}},$$  

$$d^2_{\text{min, bi-ortho, opt}} = 2(1 + \log_n 2N_{\text{bi-ortho}}), \quad \text{when } n = 2.$$  

$^{12}$Subscript “ortho” stands for orthogonal signaling, while “bi-ortho” for bi-orthogonal signaling.
respectively. Therefore, we conclude that

\[ d_{\text{min-bi-orthopt}}^2 > d_{\text{min-orthopt}}^2. \] (4.94)

This is why the bi-orthogonal signaling is preferable to the orthogonal one.
CHAPTER V

Efficient Decoding Algorithms

The maximum likelihood decoding algorithm has been developed in Section 3.2.2 for HDTCM systems. It takes $B \cdot n^B$ multiplications and additions for an ML decoder to decode a codeword. If $n$ and $B$ are large, the ML decoding is not efficient. In this chapter, we present the search for efficient decoding algorithms by making use of the reliability information of hard-decoded symbols in a codeword.

5.1 Reliability of Hard-decoded Symbol

The maximum likelihood hard decision of bi-orthogonal signals shown in Figure 15 is an $N$-ary symbol hard decision. The decision metric is the inner product, which is the absolute value of the received magnitude. The decision is accomplished by selecting the symbol with the largest absolute signal value, and the sign of the value determines whether $s_i(t)$ or $-s_i(t)$ was most likely transmitted.

$$\begin{array}{cccc}
  r_{11} & r_{21} & \cdots & r_{B1} \\
  r_{12} & r_{22} & \cdots & r_{B2} \\
  \vdots & \vdots & \cdots & \vdots \\
  r_{1N} & r_{2N} & \cdots & r_{BN} \\
\end{array}$$

Table 23: Re-arrangement of received sequence.
If the received sequence is re-arranged in a manner shown in Table 23, where \( r_{ij} \) is the \( j \)-th coordinate in the \( i \)-th symbol (\( i = 1, \ldots, B; j = 1, \ldots, N \)), then a matrix whose elements are equal to the absolute value of \( r_{ij} \) can be obtained.

Denote the above matrix by \( P \). Then

\[
P = \begin{bmatrix}
| r_{11} | & | r_{12} | & \cdots & | r_{1B} | \\
| r_{i1} | & | r_{i2} | & \cdots & | r_{iB} | \\
| \vdots | & | \vdots | & \ddots & | \vdots | \\
| r_{iN} | & | r_{2N} | & \cdots & | r_{BN} |
\end{bmatrix}.
\] (5.1)

To optimally hard-decode the \( i \)-th symbol in the received codeword, all we need is to find out the maximum value and the index \( j \) (\( j = 1, \ldots, N \)) by a simple sorting of the \( i \)-th column, i.e., \([r_{i1}, r_{i2}, \ldots, r_{iN}]^T\). The sign and the index \( j \) will point to the symbol that is most likely. Let's call that maximum absolute value of the symbol as the symbol likelihood. The likelihood information measures the reliability, or probability of correct symbol decision. The higher the likelihood, the more reliable the decision and the higher the probability of correct decision. For the above codeword sequence, after \( B \) symbols are hard-decoded separately, we can arrange these \( B \) temperately hard-decoded symbol likelihoods in either descending or ascending order. The one with the largest value is the most reliable one, which means that this symbol decision is the most reliable among all \( B \) decisions. The reliability of symbol decision decreases as its corresponding symbol likelihood decreases.

It is desirable to have theoretical quantification of the reliability information. This may be done with the help of order statistics, but the derivation will be very complicated because of different distribution of the samples. Nonetheless, we can obtain the reliability information by conducting a simulation. Here we use error probability to show the reliability. The smaller the error probability, the high the
Table 24: The error probability of hard symbol decision at various $E_b/N_0$ for $(8,2,3,12)$ scheme.

<table>
<thead>
<tr>
<th>Reliability</th>
<th>$E_b/N_0$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0421 0.0174 0.0055 0.0015 0.0003 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.0876 0.0409 0.0155 0.0046 0.0009 0.0001 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>3</td>
<td>0.1362 0.0699 0.0288 0.0094 0.0025 0.0003 0.0001 0.0000 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>4</td>
<td>0.1875 0.1048 0.0477 0.0171 0.0046 0.0009 0.0001 0.0000 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>5</td>
<td>0.2374 0.1438 0.0727 0.0278 0.0082 0.0016 0.0002 0.0001 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>6</td>
<td>0.2879 0.1883 0.1018 0.0433 0.0130 0.0032 0.0005 0.0001 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>7</td>
<td>0.3386 0.2361 0.1365 0.0648 0.0227 0.0060 0.0009 0.0001 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>8</td>
<td>0.3916 0.2879 0.1817 0.0954 0.0362 0.0105 0.0019 0.0003 0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>9</td>
<td>0.4464 0.3404 0.2358 0.1335 0.0587 0.0192 0.0038 0.0005 0.0001 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>10</td>
<td>0.4965 0.4067 0.2967 0.1856 0.0942 0.0336 0.0082 0.0011 0.0001 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>11</td>
<td>0.5610 0.4779 0.3791 0.2628 0.1524 0.0677 0.0211 0.0038 0.0005 0.0001 0.0000 0.0000</td>
</tr>
<tr>
<td>12</td>
<td>0.6382 0.5690 0.4891 0.3823 0.2650 0.1522 0.0675 0.0210 0.0051 0.0005 0.0000 0.0000</td>
</tr>
</tbody>
</table>

Table 24 shows a simulation result of the error probabilities of symbol hard decision when $N = 8$, $n = 2$, $D = 3$, $B = 12$ according to the order of reliability. The reliability is numbered in descending order. ‘1’ refers to the most reliable one. The examination of the reliability is confined to $B$ symbols in a codeword sequence. In each trial, the decision with the largest symbol likelihood is declared as the most reliable. The total number of simulation trials is $10^5$. The difference between ‘0’ and ‘0.0000’ in the table is that ‘0’ means no error is made at all during $10^5$ trials, while ‘0.0000’ means the number of errors ranges between one to four. This is because of rounding of the MATLAB format used.

The reliability simulation is conducted in the following steps

1. Perform a temporary symbol hard-decision on $B$ symbols to obtain the symbol likelihoods of hard-decoded symbols. Sort these likelihood values and arrange
them from the greatest to the smallest.

2. Trellis path comparison: Obtain the trellis paths of the hard-decoded symbols and compare them to the corresponding trellis paths of the transmitted symbols respectively. The comparison is done according to the order of reliability\(^1\). If two paths are exactly matched, then a correct decision is declared. Otherwise, an error is announced. The number of errors is accumulated with respect to each distinct reliability.

![Error Probability of Symbol Hard Decision at Various Reliability](image)

Figure 65: The error probability of symbol hard decision at various reliability when \( B = 12 \).

Table 24 lists the probabilities of symbol error with respect to different reliabilities. In the table, the first row data corresponds to the error probabilities of the most reliable symbol at various \( E_b/N_0 \). The second row is for the second most reliable symbol, and so on, until the least reliable symbol. Figure 65 plots the error

\(^1\)From the most reliable to the least reliable.
probabilities of the first five most reliable symbols. The error probability shown in the plot is in linear scale instead of logarithm scale for better illustration. It shows that at a fixed $E_b/N_0$, the error probability becomes larger as the reliability deteriorates. For a fixed reliability symbol, its error probability becomes smaller as $E_b/N_0$ becomes larger. At high $E_b/N_0$, the error probability approaches zero. This should in turn mean that such a symbol hard-decision is very reliable. In this case, one is almost sure that this hard-decoded symbol is one of the symbols in the transmitted sequence. However, at low $E_b/N_0$, the reliability is high but not high enough. Thus one cannot make the above assertion.

Table 25 shows the reliability of hard-decoded symbols for the same $N, n, D$ while $B = 24$. Visual illustration is shown in Figure 66 for the first five most reliable ones. It shows that it has the same relationship between $E_b/N_0$, reliability and error probability as that when $B = 12$. However, we see that the first twelve most reliable symbols when $B = 24$ are more reliable than those when $B = 12$. We can conclude that as $B$ becomes larger, the reliability of hard-decoded symbols is improved for the same $N, n, D$. This is because of the increasing number of samples.

5.2 Reliability of Hard-Decoded Multi-Symbol Sub-Blocks

Section 5.1 has examined the reliability of hard-decoded symbols. The reliability of a hard-decoded symbol becomes a little higher as $B$ increases. In this section, we examine the reliability of multi-symbol sub-blocks. If a sub-block has $b$ symbols, we call it $b$-symbol sub-block. Table 26 shows the simulation result of the reliability of hard-decoded two-symbol sub-block when $N = 8, n = 2, D = 3, B = 12$. The simulation is conducted similarly to Section 5.1. The number of simulation trials is
<table>
<thead>
<tr>
<th>Reliability</th>
<th>$E_b/N_0$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0.0221</td>
</tr>
<tr>
<td>2</td>
<td>0.0436</td>
</tr>
<tr>
<td>3</td>
<td>0.0658</td>
</tr>
<tr>
<td>4</td>
<td>0.0914</td>
</tr>
<tr>
<td>5</td>
<td>0.1145</td>
</tr>
<tr>
<td>6</td>
<td>0.1404</td>
</tr>
<tr>
<td>7</td>
<td>0.1674</td>
</tr>
<tr>
<td>8</td>
<td>0.1956</td>
</tr>
<tr>
<td>9</td>
<td>0.2204</td>
</tr>
<tr>
<td>10</td>
<td>0.2464</td>
</tr>
<tr>
<td>11</td>
<td>0.2731</td>
</tr>
<tr>
<td>12</td>
<td>0.2989</td>
</tr>
<tr>
<td>13</td>
<td>0.3262</td>
</tr>
<tr>
<td>14</td>
<td>0.3542</td>
</tr>
<tr>
<td>15</td>
<td>0.3804</td>
</tr>
<tr>
<td>16</td>
<td>0.4070</td>
</tr>
<tr>
<td>17</td>
<td>0.4368</td>
</tr>
<tr>
<td>18</td>
<td>0.4658</td>
</tr>
<tr>
<td>19</td>
<td>0.4897</td>
</tr>
<tr>
<td>20</td>
<td>0.5203</td>
</tr>
<tr>
<td>21</td>
<td>0.5501</td>
</tr>
<tr>
<td>22</td>
<td>0.5858</td>
</tr>
<tr>
<td>23</td>
<td>0.6267</td>
</tr>
<tr>
<td>24</td>
<td>0.6802</td>
</tr>
</tbody>
</table>

Table 25: The error probability of hard symbol decision at various $E_b/N_0$ for (8,2,3,24) scheme.
Figure 66: The error probability of symbol hard decision at various reliability when $B = 24$.

$10^5$. The first row is the reliability of the most reliable sub-block at various $E_b/N_0$. The second row is the second most reliable sub-block, and so on, to the least reliable sub-block. Figure 67 shows the first five most reliable sub-blocks. Notice that those first five most reliable two-symbol sub-blocks have higher reliability than those of hard-decoded symbols as discussed in Section 5.1.

However, if $B = 24$, by examining the curves in Figure 66 and Figure 68, the improvement of the reliability is noticed. This is the same as the case when $b = 1$. We can conclude that when two-symbol sub-block is used, the reliability is improved. We next examine how the reliability of sub-blocks will change if $b$ is further increased. Table 28 lists the reliability of three-symbol hard-decoded sub-blocks and Figure 69 shows some partial reliability curves. However, it does not show that the reliability of two-symbol sub-blocks is better than that of three-symbol sub-blocks by comparing
Table 26: The error probability of hard-decoded two-symbol sub-block at various $E_b/N_0$ for (8,2,3,12) scheme.

<table>
<thead>
<tr>
<th>Reliability</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0281</td>
<td>0.0126</td>
<td>0.0056</td>
<td>0.0021</td>
<td>0.0007</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.0599</td>
<td>0.0281</td>
<td>0.0117</td>
<td>0.0044</td>
<td>0.0014</td>
<td>0.0005</td>
<td>0.0001</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.1060</td>
<td>0.0512</td>
<td>0.0215</td>
<td>0.0084</td>
<td>0.0030</td>
<td>0.0010</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>4</td>
<td>0.1759</td>
<td>0.0899</td>
<td>0.0394</td>
<td>0.0153</td>
<td>0.0054</td>
<td>0.0015</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.2744</td>
<td>0.1622</td>
<td>0.0784</td>
<td>0.0320</td>
<td>0.0109</td>
<td>0.0039</td>
<td>0.0011</td>
<td>0.0003</td>
<td>0.0000</td>
</tr>
<tr>
<td>6</td>
<td>0.4232</td>
<td>0.3027</td>
<td>0.1844</td>
<td>0.0909</td>
<td>0.0359</td>
<td>0.0111</td>
<td>0.0034</td>
<td>0.0006</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

Figure 67: The error probability of two-symbol sub-block hard decision at various reliability when $B = 12$. 
Figure 68 and Figure 69. Actually, it is worse. Thus, increasing $b$ does not mean reliability can be improved definitely. Tentatively, it appears that two-symbol sub-blocks are preferable.

<table>
<thead>
<tr>
<th>Reliability</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0154</td>
<td>0.0067</td>
<td>0.0032</td>
<td>0.0009</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.0274</td>
<td>0.0123</td>
<td>0.0050</td>
<td>0.0022</td>
<td>0.0006</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.0410</td>
<td>0.0191</td>
<td>0.0083</td>
<td>0.0034</td>
<td>0.0011</td>
<td>0.0003</td>
<td>0.0001</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.0610</td>
<td>0.0271</td>
<td>0.0111</td>
<td>0.0044</td>
<td>0.0016</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.0835</td>
<td>0.0382</td>
<td>0.0154</td>
<td>0.0060</td>
<td>0.0022</td>
<td>0.0006</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0.1121</td>
<td>0.0506</td>
<td>0.0213</td>
<td>0.0080</td>
<td>0.0030</td>
<td>0.0009</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0.1447</td>
<td>0.0694</td>
<td>0.0297</td>
<td>0.0108</td>
<td>0.0040</td>
<td>0.0013</td>
<td>0.0006</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>8</td>
<td>0.1888</td>
<td>0.0966</td>
<td>0.0414</td>
<td>0.0152</td>
<td>0.0054</td>
<td>0.0016</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>9</td>
<td>0.2409</td>
<td>0.1316</td>
<td>0.0590</td>
<td>0.0216</td>
<td>0.0080</td>
<td>0.0025</td>
<td>0.0006</td>
<td>0.0002</td>
<td>0.0000</td>
</tr>
<tr>
<td>10</td>
<td>0.3065</td>
<td>0.1840</td>
<td>0.0877</td>
<td>0.0341</td>
<td>0.0120</td>
<td>0.0040</td>
<td>0.0012</td>
<td>0.0002</td>
<td>0.0000</td>
</tr>
<tr>
<td>11</td>
<td>0.3923</td>
<td>0.2615</td>
<td>0.1412</td>
<td>0.0605</td>
<td>0.0210</td>
<td>0.0067</td>
<td>0.0020</td>
<td>0.0004</td>
<td>0.0001</td>
</tr>
<tr>
<td>12</td>
<td>0.5102</td>
<td>0.3896</td>
<td>0.2519</td>
<td>0.1342</td>
<td>0.0551</td>
<td>0.0189</td>
<td>0.0050</td>
<td>0.0014</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

Table 27: The error probability of hard-decoded two-symbol sub-block at various $E_b/N_0$ for (8,2,3,24) scheme.

5.3 Efficient Decoding by Using Reliability Information

The maximum likelihood decoding requires the calculation of $n^B$ codeword likelihoods and then the selection of one as the most likely transmitted codeword. Optimality is achieved because the algorithm has examined all possibilities. However, a large amount of computation is wasted in such brute-force search if the size of codebook is large. From our experience, the codeword likelihood of the transmitted codeword is usually among the top few largest at typical $E_b/N_0$ values. The goal of efficient decoding is to reduce the amount of unnecessary computation while making the performance approach that of maximum likelihood decoding. One way is to limit
Figure 68: The error probability of two-symbol sub-block hard decision at various reliability when $B = 24$.

Table 28: The probability of hard-decoded three-symbol sub-block error at various $E_b/N_0$ for (8,2,3,24) scheme.
the size of candidate codewords. By “candidate codewords” we mean that these codewords are the first few most probable ones.

Because of the inter-dependence between successive symbols, the maximum likelihood sequence detector minimizes the probability of codeword error by examining a whole sequence of symbols at a time. If we perform symbol-by-symbol maximum likelihood decisions, the inter-dependence between symbols is not utilized, thus the optimality is achieved by minimizing probability of symbol error, rather than probability of codeword error. The error probabilities in symbol hard decisions are determined by the minimum Euclidean distance between any two symbols, rather than minimum Euclidean distance between any two codewords. However, if we make symbol hard-decision upon a very reliable symbol, we are relatively certain that the actually transmitted codeword contains that hard-decoded symbol. A very reliable
symbol means that we can use it to estimate a codeword in a way that the examination of other low-probability combinations can be avoided beforehand, and thus unnecessary computation can be saved. This is true for multi-symbol sub-block hard decisions as well. Therefore, reliability information of those anchor symbols is used as a clue to search for the most probable codeword candidates. This is the starting point of searching for efficient decoding algorithms.

In hard decisions, once a decision is made the likelihood information is discarded. To make use of the reliability information, we can hard-decode a symbol/sub-block in a codeword sequence temporarily. By ‘temporarily’ we mean that we don’t accept the hard-decoded version as a final estimate to the transmitted sequence such that their likelihood information can be used later on. After obtaining $B$ symbol/sub-block likelihoods, we sort them. We call these temporarily hard-decoded symbols/sub-blocks anchors.

Because the reliability of anchors is not high enough at low $E_b/N_0$, it can not be expected that the transmitted codeword will belong to the set of codeword candidates obtained through reliability information. Thus the performance of new algorithms will not be as good as, maybe much worse than that of maximum likelihood sequence detector. However, at high $E_b/N_0$, their performance should be very close if it is not as good, because the anchors are highly reliable.

The following paragraph examines the possible efficiency margin of possible algorithms that use the reliability information of anchors over the maximum likelihood decoding algorithm. If the sub-block is of length $b$, then the total number of possible sub-block trellis paths is $S \cdot n^b = n^{D+b}$. For bi-orthogonal signaling, there are $b$ nonzero elements in each sub-block sequence. The calculation of sub-block likelihoods
of all possible $b$-symbol sub-blocks needs $bn^{D+b}$ multiplications and additions. There are $B/b$ $b$-symbol sub-blocks in a codeword sequence. Therefore, it needs $B \cdot n^{D+b}$ multiplications and additions to calculate all those sub-block likelihoods. In order to gain decoding efficiency, the other manipulations in the desired efficient algorithms should need less than $B(n^B - n^{D+b})$ multiplications and additions, where $B \cdot n^B$ is the number of multiplications and additions needed for the maximum likelihood sequence detector. Thus, the larger the size of sub-block $b$, the less the efficiency margin can be achieved. Moreover, the analysis in Section 5.2 suggests that $b$ not be greater than two.

### 5.4 Trellis Paths of Multi-Symbol Sub-Blocks

As explained in Section 4.1.2, all the states will be visited after $D$ state transitions by starting from an arbitrary state. After $D+1$ transitions, there are $n$ branches merging at each state. There are total $n^B$ legal codeword trellis paths out of $S \cdot n^B = n^{B+D}$ possibilities. This is because of the state constraint applied in the encoder. The following analysis examines the number of codewords whose legal trellis paths include the trellis path of a fixed $b$-symbol sub-block.

For a fixed $b$-symbol sub-block, the total number of all possible codeword trellis paths, legal or illegal, that include the trellis path of the sub-block is $n^{B-b}$. Because of the state constraint, only $1/S$ of them are legal, i.e., there are $n^{B-D-b}$ codewords that could include the fixed $b$-symbol sub-block. If $b > B-D$, not all of the $b$-symbol sub-blocks can be included in those codewords whose trellis paths are legal. This means that certain codewords can be constructed from some $b$-symbol sub-blocks and some can not lead to any legal codeword when $b > B-D$. If $b = B-D$, there is only
one codeword that can be constructed from any \( b \)-symbol sub-block. If \( b < B - D \), the number of codewords that can be constructed from a fixed \( b \)-symbol sub-block is \( n^{(B-D-b)} \). The above conclusion is regardless of the position of the sub-block in the sequence.

For example, a simple trellis is shown in Figure 70, where \( D = 2 \), \( n = 2 \) and \( B = 6 \). The thick solid line represents the sub-block with length of two. There are four legal codeword trellis paths can be constructed from it. The shifting of the position of the sub-block may change the initial state and the ending state of the legal codeword trellis paths, but the number of legal paths that can be constructed...
from the sub-block is always equal to four. For the same trellis, Figure 71 shows that there is only one legal codeword trellis path that can be extended over the fixed sub-block with $b = 4$. Figure 72 shows that if the length of a sub-block is equal to five, only half of all possible sub-blocks with $b = 5$ can be extended to certain legal paths. The figure shows one that no codeword can be built from it.

In summary, for $b$-symbol sub-block being a part of a legal sequence, $b$ has to satisfy the following inequality

$$b \leq B - D.$$  (5.2)
5.5 Efficient Decoding Algorithms

Assume the received signal is arranged into a $N \times B$ matrix as shown in Equation (5.1). A temporary symbol hard-decision is made by selecting the one with the largest symbol likelihood column by column. Vector $R$ contains the likelihoods of those temporary hard-decoded symbols. Sort the symbol likelihood vector $R$, then select $L$ top reliable anchors.

In the following sections, we present two algorithms that make use of anchors to reduce the size of the codeword candidates so as to make the decoding efficient and faster\(^2\). The more efficient the algorithm, the faster it would be. However, the trade-off is between performance and efficiency. The gain in efficiency has to be paid by the degradation of performance. The goal is to have a good algorithm whose performance is close to that of the maximum likelihood sequence detector, but is much more efficient and faster. To achieve this goal, the new algorithms should somewhat resemble the way the maximum likelihood sequence detector works, that is, calculating and comparing several likelihoods, but confining these operations to a limited set of candidates. The number of candidates should be much smaller than the size of codebook $n^B$. One algorithm makes use of the codebook, the other makes use of the trellis structure instead. Both algorithms are sub-optimum. The optimality is traded for speed and efficiency. These algorithms are expected to work well in the high $E_b/N_0$ ranges in which anchors are of high reliability. In low $E_b/N_0$ regions,

\(^2\)The programs are written in MATLAB. The MATLAB script is executed by direct compilation of each script sentence. We can not conclude which one is faster by simply comparing the execution time of the maximum likelihood algorithm and that of the new algorithm. MATLAB is optimized for matrix and vector operations. The ML decoding algorithm is merely a multiplication of a matrix and a vector. For a better comparison, the programs should be written in C. This is recommended for future research. Here, only the method and the performance of the new algorithms are presented.
their performance will severely deteriorate. In this case, the gained efficiency in computation can not compensate the loss in performance.

5.5.1 Algorithm A: The use of a codebook

Algorithm A describes a way to pinpoint those most likely codeword candidates. It uses the codebook and anchors to reduce the size of the set of codeword candidates. The decision is then made by choosing the one with the largest likelihood among those candidates. The flowchart of Algorithm A is shown in Figure 73. Algorithm A is fully described by the flowchart and the following statements. The number of anchors \( K \) ranges between 1 and \( B \), i.e., \( 1 \leq K \leq B \).

1. First of all, the codebook is used to construct two template index matrices, \( \text{POSITIVE} \) and \( \text{NEGATIVE} \), whose elements are the indices of different anchors with respect to their specific positions in the sequence. Matrix \( \text{POSITIVE} \) is for positive anchors, Matrix \( \text{NEGATIVE} \) is for negative anchors. Both matrices are of the same sizes. The structure of matrix \( \text{POSITIVE} \) and matrix \( \text{NEGATIVE} \) is shown in Figure 74. There are \( 2N \) channel symbols, the repeat rate of each symbol in every column is thus \( n^B/2N \).

2. The indices of likely codewords can be located from the above template index matrices according to the anchors’ positions in the sequence. Each anchor points to the indices of \( n^B/2N \) codewords.

3. Compare the likelihood of the resulted size-reduced candidate set and then make a final decision. The one with the largest codeword likelihood is favored and its index is used to recover its corresponding source block.
Take the absolute value of the received signal block

Select the maximum symbol likelihood for each symbol

Select K most reliable anchors from B temporarily hard-decoded symbols

Search those possible codewords including any one of the anchors

Compare the codeword likelihoods among this limited set of candidates

The estimated transmitted source data

Figure 73: A suboptimum decoding algorithm that reduces the size of codeword candidates by using codebook — Algorithm A.
If an anchor has very high reliability, then it is almost certain that the transmitted codeword includes the anchor. If $B$ anchors are selected, then there would be as many as $B$ possible anchors pointing to the same index of a certain codeword. That means that the index of this codeword is possibly repeated as many as $B$ times in the candidate set. Thus duplicate calculation of codeword likelihoods occurs. The elimination of those duplicate entries is optional. It depends on how much amount of computation is wasted in the duplicate computation compared to the amount of computation needed to eliminate those duplicate entries.

Note that in Algorithm A, each anchor is used to locate those likely codewords, then compare the likelihoods of those candidates and select the one with the highest value. Thus, the order of the reliability of anchors are not important. If $B$ anchors
are selected, then the sorting of these anchors is not necessary. Otherwise sorting is required, because we need to choose \( K \) out of \( B \) symbols.

The number of anchors selected affects processing speed and performance of decoding. If more anchors are selected, the decoding performance approaches that of the maximum likelihood sequence detector, especially at moderate \( E_b/N_0 \), but with more computation, thus less efficiency is gained. For example, the error performance of Algorithm A is compared with that of maximum likelihood decoding in Figure 75 when \( N = 16, n = 2, D = 4, B = 6 \). The number of trials is \( 10^5 \).

![Performance Comparison between Algorithm A and ML](image)

Figure 75: The error performance of Algorithm A.

Figure 75 tells us that the performance of Algorithm A gradually approaches to that of the maximum likelihood sequence detector as \( E_b/N_0 \) becomes larger. The comparison in efficiency with the maximum likelihood algorithm should be done

\(^{3}\text{At these } E_b/N_0 \text{ values, the anchors are reliable, but not highly reliable.} \)
by detailing the number of multiplications and additions of the new algorithm\textsuperscript{4}. From the simulation, one would notice that Algorithm A can correctly decoded some codewords that can not be done by the maximum likelihood sequence detector. It can not be claimed that Algorithm A is superior to maximum likelihood sequence detector in the sense of performance simply because of this fact, because one would also notice the opposite phenomenon. And the probability that this situation happens is greater than that of the previous one.

5.5.2 Improvement of Algorithm A using multi-symbol sub-blocks

Algorithm A presents a way to reduce the size of candidate codewords by the use of anchor symbols. Each anchor points to $n^B/2N$ possible codewords that include this anchor. If $K$ anchors are chosen, the size of candidates is $K \cdot n^B/2N$ including the duplication of indices, where several anchors may point to the same codeword. In order to make the decoding more efficient, the size of candidate codewords should be further reduced.

If $b = 1$, the number of codewords that include one of the anchor symbols is thus $n^{B-D-1}$. If $B$ is large, than $n^{B-D-1}$ is large too. However, all we need is to pick out the correct codeword. Even if the anchor is included in the correct codeword, other $n^B/2N − 1$ candidates to which this anchor points still require computation. By examining those codeword candidates, we find out that those with larger codeword likelihoods are usually selected by these $K$ anchors, but some with very low likelihoods are also selected. Remember that so far each anchor is independently used. The inter-dependency between anchors has not yet been examined. If we examine those

\textsuperscript{4}This is recommended for future research.
anchors at the same time, those unlikely selected candidates can be easily eliminated. Thus, we should take inter-dependency into account to further reduce the number of candidate codewords. The solution is to use multi-symbol sub-blocks.

Section 5.4 gives the number of codewords that include a $b$-symbol sub-block as

$$\frac{n^B}{S \cdot n^b} = n^{B-D-b}. \quad (5.3)$$

if $b \leq B - D$. This means that if anchors are two-symbol sub-blocks, the number of codeword candidates that include this anchor is reduced by $1/n$ times than that of one-symbol anchor case. Therefore, in order to improve the efficiency of Algorithm A, larger sub-blocks need to be adopted. Similar to the construction of POSITIVE and NEGATIVE matrices, a template index matrix whose elements are indices of codewords that include an individual sub-block can also be constructed.

However, Section 5.2 gives the amount of multiplications and additions needed in calculating all the sub-block likelihoods of $b$-symbol sub-block is $B \cdot n^{D+b}$, which grows exponentially with $b$. Moreover, the template index matrix should become larger so as to accommodate all possible sub-block combinations. This will compromise our efforts of making the algorithm efficient. Thus, a hierarchical structure is proposed.

By “hierarchical structure” we mean that:

1. $b$-symbol sub-blocks are used to significantly reduce the size of candidate codewords.

2. Within each $b$-symbol sub-block, anchor symbols or anchor sub-blocks with smaller sub-block sizes are used to reduce the computation of the likelihoods of all $b$-symbol sub-blocks. The selection of anchor symbol or anchor sub-block
depends on the value of $b$. If $b$ is large, sub-blocks are preferred, otherwise, anchor symbols are used.

In this way, the computation of each base unit ($b$-symbol sub-block) is sharply reduced.

A disadvantage of using the codebook is that the construction of the template index matrix is not practical when $B$ is a large number, even though tabulation saves computation in encoding. To make the decoding algorithm efficient at large $B$, Algorithm B is proposed. The goal of Algorithm B is also to reduce the number of candidates. However, the reduction is accomplished by exploiting the trellis structure.

5.5.3 Algorithm B: The use of trellis structure

The maximum likelihood sequence detector favors the codeword with the largest codeword likelihood. The codeword likelihood is the sum of $B$ symbol likelihoods or $B/b$ sub-block likelihoods. If the codeword likelihood is the largest, it is expected that the codeword include some reliable anchors. In this algorithm, only two anchors are selected from $B/b$ temporarily hard-decoded sub-blocks. Anchor A is the most reliable and Anchor B is the second most reliable. Then by using trellis structure, a whole trellis path can be built from these two anchors. Several segments piece the whole path together. The maximum likelihood algorithm is used to select the most likely one for each segment. This assures that the choice of every segment is locally optimum in the hope of achieving global optimality for the whole path.

As we know, if $b < B - D$, a sub-block can be guaranteed to be a part of certain legal sequences. Thus Anchor A contributes the largest sub-block likelihood to the likely codeword, while Anchor B is the second most prominent contributor. Assume
the probability of error of Anchor A is $p_1$, and $p_2$ for Anchor B. The high reliability means $p_1$ and $p_2$ are very small. The probability that both anchors are in error is given by $p_1 \cdot p_2$, which is much smaller than $p_1$ or $p_2$. The probability that both anchors are correct is $(1 - p_1) \cdot (1 - p_2)$, which very close to one. The probability that A is in error but B is correct is $p_1 \cdot (1 - p_2)$ and the probability that B is in error but A is correct is $p_2 \cdot (1 - p_1)$. These two probabilities approach to $p_1$ and $p_2$ respectively.

If one of the anchors is in error, the algorithm has difficulty. However, this does not mean that errors must happen in these cases. The imperfect reliability of the anchors can be partly detected by the relative position of anchors in the sequence and the state constraint. Also, remember that the ML detector makes errors. Thus, we can not expect that Algorithm B will not make errors.

In general, a trellis path can be segmented into one to three parts by two anchors as shown in Figure 76. If two anchors happen to be at both ends, then there is only one segment. If either A or B is at one of the ends, then there are two segments. Otherwise, there are three segments.

Figure 76: The segmentation of trellis paths by two anchors: (a) both are end blocks; (b) both are in the middle; (c) one of them is the middle and the other is a end block.
Assume Anchor A and B are $L$ sub-blocks apart, or $L \cdot b$ symbols apart. Then the number of possible paths between these two anchors is $n^{L \cdot b - D}$. So if $L \cdot b \geq D$, Anchor A and B can always be lined up by certain segment paths. We can always use the state table to obtain those segment paths. However, if $L \cdot b < D$, we may not be able to find a segment path that can bridge A and B. If this is the case, then either A or B may not be correct. When a whole trellis path is reconstructed, the ending state and the starting state need to be checked so that the state constraint is satisfied. Otherwise, the path is not a legal trellis path. An illegal path may occasionally have very high likelihood, but it is not of interest. Recursive search ends when a legal path is found.

![Figure 77: The segmentation of a trellis path by two anchor sub-blocks.](image)

In Figure 77, two anchor sub-blocks are in the middle. There are four possible paths in Segment B. The maximum likelihood sequence detector is used to select an optimum segment path for Segment B. There are also four possible paths in Segment A and C, respectively. Although we can still apply the maximum likelihood sequence detector to choose an optimum segment path for Segment A or C independently, those two paths may not satisfy the state constraint. That means if we simply apply the maximum likelihood sequence detector to Segment A or C separately, the
path obtained is locally optimum within that segment, rather than globally optimum within the whole sequence. Global optimality is desired in order to select the most likely codeword. Thus we need to find out those possible legal paths by jointly taking Segment A and C into account. Then we calculate the combined path likelihoods for those legal combinations and select the one with the largest combined likelihood.

As for the example shown in Figure 77, legal combinations are the ones in which the starting state in Segment A is equal to the ending state in Segment B. Thus there are only four possible combined paths which need to be examined.
CHAPTER VI

Conclusions

6.1 Conclusions

In this dissertation, performance of high dimensional trellis-coded modulation in
AWGN channels is analyzed theoretically and an optimum maximum likelihood de­
coder is developed. Efficient decoding methods are also investigated.

HDTCM integrates a block code with state-permuted trellis structure and an ex­
panded high-dimensional signal constellation. HDTCM was originally proposed to
combine coding and modulation in spread spectrum communications into a single
process. Later it is found that HDTCM in fact combines not only coding and mod­
ulation, but also PN spreading into one single process, if PN spreading is viewed as
an independent process. This insight is very important. Without it, the coding gain
of the HDTCM systems would be difficult to analyze and the advantages of HDTCM
would be hard to identify.

The performance analysis is achieved by first identifying differences between cur­
rent TCM and HDTCM. It then concludes that the performance analysis of HDTCM
can not be readily borrowed from that of current TCM. A self-established systematic
analysis is thus derived. Those differences also allow us to realize that the Viterbi
algorithm can not be used as a decoding algorithm in HDTCM.
A general picture of HDTCM has been unveiled through the theoretical analysis of those pinpointed differences. The performance analysis can be summarized into three parts:

- Investigation of the properties of HDTCM.
- Theoretical analysis of error probabilities.
- Derivation of coding gain.

The dissertation proves that HDTCM with bi-orthogonal signal constellation is uniform. Such uniformity allows the choice of an arbitrary reference such that performance analysis is greatly simplified and becomes systematic. The proof also points out that proving the uniformity of HDTCM is equivalent to proving the uniformity of the signal constellation engaged.

Free Euclidean distance of HDTCM is analyzed with respect to its trellis structure. Minimum Euclidean distance is investigated subsequently. It shows that minimum Euclidean distance is bounded by free Euclidean distance. Next, the influence of four design parameters $N$, $n$, $D$, $B$ is discussed and their optimum values are given by examining the minimum Euclidean distance and the weight distribution of distinctive Euclidean distances.

The cyclic properties of HDTCM is investigated and a grouping rule is given. A whole sourcebook can be separated into a number of groups, where each group has its own distinctive pattern. The trellis path and codeword of any element can be generated from the leading pattern according to the nature of shift between this element and the leader of this group. The rule suggests a way to save computation during
encoding by examining those distinctive patterns only. Other possible applications of these cyclic properties are mentioned in Section 6.2.

Probability of bit error is obtained by first analyzing probability of codeword error. Probability of codeword error is bounded by union upper bound and lower bound. The analytical expressions of these two bounds are given. In a similar way, the union bounds and lower bounds for $P_{se}$ and $P_{be}$ are also given. Next, the asymptotic performance of HDTCM at high signal-to-noise ratios is discussed. The percentage contribution of each distinct Euclidean distance is studied and minimum Euclidean distance is identified as the determining factor of system's performance. Good code criteria are discussed and a scheme with $B = D + 3$ is claimed to be an asymptotically good scheme when $n = 2$.

Four kinds of spread spectrum communication systems are compared in terms of system configuration. Their differences are identified and their processing gains are compared. An uncoded conventional spread spectrum system is selected as a reference for an HDTCM spread spectrum system in order to calculate its coding gain. The modulation of this reference system can be either MFSK or MPSK. Either one can be chosen, but the coding gain obtained would differ by 3 dB. Next, the analytical expression of asymptotic coding gain is derived. When $n = 2$, the range of asymptotic coding gain of HDTCM is between 3 dB and 8 dB for reasonable spreading.

The ML decoding algorithm is developed for HDTCM. Practical implementation of this optimum algorithm is also considered. To speed up decoding, two efficient algorithms that make use of the reliability of anchor symbols/sub-blocks are presented. One algorithm makes use of codebook, the other makes use of trellis structure.
6.2 Future Research

**TD/CCSK post-processing unit**: The transform domain/cyclic code shift keying (TD/CCSK) scheme developed in [35] could be used as a post-processing unit for the proposed HDTCM scheme in order to further increase the immunity of this system to the multi-path in a fading HF channel.

**Cyclic properties**: First, the mathematical nature of the pronounced cyclic properties could be examined and then the generalization of the grouping rule could be performed. The cyclic properties have provided a way to save computation during encoding by examining only a limited number of patterns. The author conjectures that such properties could also be used in finding efficient decoding methods by trying to eliminate those illegal state combinations beforehand. Another benefit goes to the performance analysis. We only need to locate the leading patterns that have a distance of $d_{\text{min}}$ instead of generating a whole codebook.

**Performance analysis in fading channel**: The performance analysis of HDTCM systems in AWGN channel, that is presented in this dissertation, can be treated as a benchmark for future research. However, spread spectrum communications channel is an RF channel. To have a closer idea of the system’s actual performance, a proper model of a fading channel needs to be selected so that the performance of the system in a fading multi-path HF channel can be investigated.
Other high dimensional signal constellations: The high dimensional signal constellation chosen for HDTCM in this dissertation is the bi-orthogonal signaling set. The performance of orthogonal signal constellation is also examined. The analysis shows that if bi-orthogonal signaling is used, the size of input alphabet $n$ prefers to be two. There may exist other proper high dimensional signal constellations that could compensate for the shortcomings of bi-orthogonal signaling for at higher bit rate or greater $n$.

Decoding algorithms — Algorithm A and B: It is recommended to analyze the number of multiplications and additions needed for each detection in both algorithms presented in Chapter V. For convenient comparison, it is recommended to realize both Algorithm A and B in C programming language so that the speed can be compared by directly recording the execution time of the simulation.

Efficient decoding algorithm by the use of cyclic properties: The cyclic properties of HDTCM schemes could be used in the decoding algorithm — Algorithm B to more efficiently find a segment path. It should be faster than the way that simply uses the state table.
BIBLIOGRAPHY


Appendix A

The Performance of Bi-Orthogonal Signaling

If the signal dimension is $N$, then the size of the bi-orthogonal signaling set is $2N$. The matched filter detection of bi-orthogonal signals with size of $2N$ can base on $N$ orthogonal basis signals, $s_1(t), \ldots, s_N(t)$, which is shown in Figure 15. The decision is then based upon which matched filter output has the largest absolute value, and the sign of the output corresponding to the one with largest absolute value determines whether $s_i(t)$ or $-s_i(t)$ was most likely transmitted.

Assume $z_i$ is the received signal when $s_i(t)$ is transmitted over an AWGN channel, then

$$z_i(t) = s_i(t) + n(t), \quad i = 1, \ldots, N.$$ 

Without loss of generality, suppose $s_1(t)$ is transmitted, then based on the above decision rule, the conditional probability of correct detection $P_{C|s_1}$ is

$$P_{C|s_1} = \text{Prob} [ z_1 > 0, |z_1| > (|z_2|, \ldots, |z_N|) | s_1 ]$$

$$= \int_0^\infty \cdots \int_{-z_1}^{z_1} P(z|s_1) \, d(z)$$

(A.1)

where $P(z|s_i)$ is the conditional multivariate Gaussian probability density function corresponding to a set of $N$ orthogonal signals $\{s_1(t), \ldots, s_N(t)\}$ when $s_i(t)$ is trans-
mitted.

\[ P(z|s_i) = \prod_{j=1}^{N} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(z_j - m_{ij})^2}{N_0}} \]

where the mean for \( j \)th element in \( i \)th signal is

\[ m_{ij} = \begin{cases} 1 & , i = j \\ 0 & , i \neq j \end{cases} \]

That is,

\[ P(z|s_1) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(z_1 - 1)^2}{N_0}} \prod_{j=2}^{N} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{z_j^2}{N_0}} \]

\[ \int_{-z_1}^{z_1} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{z_j^2}{N_0}} dz_j = \frac{1}{2} - Q(z_1 \sqrt{\frac{2}{N_0}}) \]

Thus, the conditional probability of correct detection is

\[ P_{C|s_1} = \int_0^\infty \left[ \frac{1}{2} - Q(z_1 \sqrt{\frac{2}{N_0}}) \right]^{(N-1)} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(z_1 - 1)^2}{N_0}} dz_1 \quad \text{(A.2)} \]

By making the following variable transformation,

\[ y = \frac{z_1 - 1}{\sqrt{N_0}/2} \]

we have

\[ \frac{z_1}{\sqrt{N_0}/2} = y + \sqrt{\frac{2}{N_0}} \]

Therefore the Equation (A.2) is changed to

\[ P_{C|s_1} = \int_{-\sqrt{\frac{2}{N_0}}}^{\infty} \left[ \frac{1}{2} - Q(y + \sqrt{\frac{2}{N_0}}) \right]^{(N-1)} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \quad \text{(A.3)} \]

The average probability of correct symbol detection is

\[ P_C = \sum_{i=1}^{2N} P_{C|s_i} P(s_i) = P_{C|s_1} = P_{C|s_1} \quad \text{(A.4)} \]

The probability of symbol error is thus equal to

\[ P_{se} = 1 - P_C = 1 - P_{C|s_1} \quad \text{(A.5)} \]
In the bi-orthogonal signaling, there are two kinds of word errors. The first kind of word error is occurred when the antipodal of the transmitted signal is selected instead of the transmitted signal itself. The probability of this event is denoted by $P_{1e}$ and is given by

$$P_{1e} = \text{Prob} \left[ z_1 < 0, |z_1| > (|z_2|, \ldots, |z_N|) | s_1 \right]$$

$$= \frac{1}{(N-1) \text{ fold}} \int_{-\infty}^{0} \int_{z_1}^{-z_1} \cdots \int_{z_1}^{-z_1} P(z_1) d(z)$$

Notice that in Equation (A.6), the limits of the integral are a little bit different from that in Equation (A.1). By applying similar algebraic manipulations as that in the derivation of $P_{s|s_1}$, we have

$$P_{1e} = \int_{-\infty}^{-\sqrt{N_0}} \left[ -\frac{1}{2} + Q(y + \sqrt{\frac{2}{N_0}}) \right]^{(N-1)} e^{-\frac{y^2}{2}} dy \quad (A.6)$$

An error of second kind $P_{2e}$ occurs when any one of the $2N - 2$ signals that are orthogonal to the transmitted signal is selected. In Equation (A.5), $P_{se}$ is the sum of both $P_{1e}$ and $P_{2e}$. Because $P_{2e}$ is hard to calculate analytically, $P_{2e}$ is obtained by

$$P_{2e} = P_{se} - P_{1e} \quad (A.7)$$

Figure 78 and Figure 79 show the quantitative difference between $P_{1e}$ and $P_{2e}$. Each probability of error is calculated for different $N$, for example, $N = 2, 4$.

Figure 78 shows that as the dimension of the signal space $N$ increases, the probability of first kind of word error $P_{1e}$ becomes smaller, while the probability of second kind word error will become larger as shown in Figure 79. This means that as $N$ increases, errors of second kind will dominate the error performance. When $N$ is large enough, the error performance of bi-orthogonal signaling will approach that of
Figure 78: Probability of first kind word error

Figure 79: Probability of second kind word error
the orthogonal signaling. This is because no matter what is the value of $N$, there is always only one signal that is antipodal to the transmitted one, but there are $2N - 2$ signals that are orthogonal to the transmitted one.

Suppose we use $N_b$ bits to represent all the possible signal waveforms. Assume that complementary data symbols (for example, 0 is complementary to 1) are encoded into complementary channel symbols\(^1\) so as to minimize the probability that a symbol error will cause all data bits to be in error. For example, if $N_b = 3$, then $s_1(t)$ is coded by 000, then $-s_1(t)$ is coded by 111, etc. Therefore, if an error of the first kind is made, there are exactly $N_b$ bits in error which means all bits are in error. Therefore, the conditional bit error probability given that an error of the first kind occurred is exactly one.

When an error of the second kind is made, the average number of data bits in error is

$$\frac{\sum_{k=1}^{N_b-1} k C^k_{N_b}}{\sum_{k=1}^{N_b-1} C^k_{N_b}} = \frac{N_b}{2}$$

where $C^k_{N_b} = \frac{N_b!}{(N_b-k)!k!}$. So the probability of a bit being in error given that an error of the second kind was made is exactly equal to one-half. The summation here extends only to $N_b - 1$ because errors of the first kind are not allowed.

The total bit error probability for bi-orthogonal symbols is then given by

$$P_{be} = P_{1e} + P_{2e}/2 \quad (A.8)$$

By jointly solving Equation (A.7) and Equation (A.8), we have

$$P_{be} = \frac{1}{2}(P_{se} + P_{1e}) \quad (A.9)$$

\(^1\) $-s_i(t)$ is complementary to, or antipodal to $s_i(t)$
An error of first kind is less likely than an error of second kind. Compared with $P_{2e}$, $P_{1e}$ is usually much smaller. As $N_b$ becomes larger, $P_{1e}$ can be omitted. The following approximation is quite accurate when $N_b \geq 3$ [37].

$$P_{be} \approx \frac{P_{se}}{2} \quad (A.10)$$
ABSTRACT


THE PERFORMANCE ANALYSIS AND DECODING OF HIGH DIMENSIONAL TRELLIS-CODED MODULATION FOR SPREAD SPECTRUM COMMUNICATIONS (179pp.)

Director of Dissertation: Dr. Jeffrey C. Dill

High dimensional trellis-coded modulation (HDTCM) is a new trellis-coded modulation scheme aimed at applications in spread spectrum communications. This dissertation presents theoretical performance analysis of HDTCM and investigation of its decoding algorithms. HDTCM integrates a block code with a state-permuted trellis structure and an expanded high-dimensional signal constellation. Bi-orthogonal signaling is chosen as the signal constellation in this dissertation. HDTCM unifies not only coding and modulation, but also PN spreading into one single process. The properties of the HDTCM scheme are investigated. First, HDTCM with bi-orthogonal signal constellation is shown to be uniform, thus an arbitrary reference can be chosen and the performance analysis is simplified. Then minimum Euclidean distance is found to be bounded by free Euclidean distance. Optimum selection of design parameters is accomplished by examining the weight distribution of Euclidean distance. The cyclic property of HDTCM is also studied and a grouping rule is given. Next, theoretical error performance of HDTCM is analyzed by deriving analytical expressions for lower bounds and upper bounds of error probabilities. Then an asymptotic upper bound at high signal-to-noise ratios is discussed. The minimum Euclidean