2016

Multiple-rate codes based on serial concatenation and block Markov superposition transmission

Bo Liu

University of Wollongong

Follow this and additional works at: https://ro.uow.edu.au/theses

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material.

Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Recommended Citation


Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Multiple-Rate Codes Based on Serial Concatenation and Block Markov Superposition Transmission

October 24, 2016

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Master of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Bo Liu

School of Electrical, Computer and Telecommunications Engineering
Statement

I, Bo Liu, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Master of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

The following publication is related to the research work conducted in this study:


Bo Liu

March 31, 2016
# Contents

1 Introduction 13  
   1.1 Background ............................................. 13  
   1.2 Research Motivation and Objectives .................. 15  
   1.3 Research Contributions ................................. 17  
   1.4 Thesis Organisation ..................................... 18  
   1.5 Publications ............................................. 19  

2 Literature Review 20  
   2.1 Type of Codes ........................................... 20  
   2.2 Linear Block Code ....................................... 22  
      2.2.1 The Minimum Distance of a Block Code 26  
      2.2.2 Maximum a Posteriori (MAP) Decoding Rule ... 27  
      2.2.3 Log-Likelihood Algebra ......................... 29  
      2.2.4 Repetition Codes ................................... 30  
      2.2.5 Single-Parity-Check Codes ....................... 30
2.2.6 First Order Reed Muller codes 33
2.2.7 Hamming and Extended Hamming codes 38
2.3 Product Codes 40
2.4 Accumulator 42
2.5 Serially Concatenated Codes 43
2.6 Block Markov Superposition Transmission 44

3 Multiple-Rate Serially Concatenated Codes with Low Error Floors 51

3.1 Introduction 51
3.2 HSA Codes: Encoder and Decoder 53
  3.2.1 Encoder 54
  3.2.2 Decoder 55
3.3 Threshold Analysis Via Three-Dimensional EXIT Chart 58
3.4 Low Weight Profile Analysis 62
3.5 Numerical Results 65
3.6 Conclusion 67

4 Block Markov Superposition Transmission with Multiple-
Rate Basic Codes

4.1 Introduction ........................................ 68
4.2 The Composition of RMEH Codes .............. 70
4.3 Decoding Complexity Analysis ................. 72
4.4 Simulation Results .............................. 75
  4.4.1 Performance of RMEH-BMST codes ... 75
  4.4.2 Comparison of RMEH-BMST Codes and
       RSPC-BMST Codes ............................. 76
4.5 Conclusion ........................................ 78

5 Conclusion ........................................ 79
List of Figures

2.1 Decoder structure of SPC code. . . . . . . . . . . . . . . . . 31
2.2 Construction of product code. . . . . . . . . . . . . . . . . 41
2.3 The decoder structure of an accumulator. . . . . . . . . . . 42
2.4 The encoding circuit of a serial concatenated code. . . . . 44
2.5 The decoder structure of a serial concatenated code. . . . . 44
2.6 The encoding diagram of the BMST code. . . . . . . . . . 46
2.7 The factor graph representation of a BMST code with $L = 4$ and $m = 2$. . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
3.1 (a) Encoder structure of serially concatenated codes with an inner accumulator; (b) An outer product code with (extended) Hamming codes and SPC codes as component codes. 55
3.2 Iterative decoder of HSA codes with three constituent decoders. “Acc DEC”, “Ham DEC” and “SPC DEC” denote the accumulator decoder, Hamming decoder and SPC decoder, respectively. $\pi$ and $\pi^{-1}$ denote the interleaver and deinterleaver. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
3.3 Three-dimensional EXIT chart for Hamming(31,26)-SPC(32,31)-
Accumulate code at $E_b/N_0 = 3.19$ dB ........................ 61

3.4 Frame error rate performance of length-992 high-rate seri-
ally concatenated codes on the AWGN channel with BPSK
modulation. .................................................. 65

3.5 Frame error rate performance of three length-992 HSA
codes using (8,7), (16,15) and (32,31)SPC codes, repec-
tively. .................................................. 66

4.1 Performance of RMEH-BMST codes composed of (32, 6)
RM codes and (32, 26) EH codes. ....................... 75

4.2 The BER performances of RSPC-BMST code and RMEH-
BMST code when code rate is 0.75. ....................... 76

4.3 Convergence rate of RMEH-BMST codes and RSPC-BMST
codes (Rate = 12/16, SNR = 4.0 dB). ....................... 77
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The decoding complexity of SPC codes.</td>
<td>33</td>
</tr>
<tr>
<td>2.2</td>
<td>The decoding complexity of RM codes.</td>
<td>38</td>
</tr>
<tr>
<td>2.3</td>
<td>The decoding complexity of Hamming codes.</td>
<td>40</td>
</tr>
<tr>
<td>3.1</td>
<td>Ensemble-average low-weight profiles of 4 length-992 code ensembles with (31,26) Hamming code or (32,26) extended Hamming code as component codes. $A_h$ is the average number of codewords with Hamming weight $h$.</td>
<td>64</td>
</tr>
<tr>
<td>3.2</td>
<td>The parameters of 5 serially concatenated codes.</td>
<td>64</td>
</tr>
<tr>
<td>4.1</td>
<td>The parameters for RMEH codes with $n=32$.</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>The parameters for RSPC codes with $n=32$.</td>
<td>72</td>
</tr>
<tr>
<td>4.3</td>
<td>The decoding complexity of four short codes (per coding bit).</td>
<td>74</td>
</tr>
<tr>
<td>4.4</td>
<td>The decoding complexity of per coded bit in RSPC-BMST and RMEH-BMST when code rate is $12/16$.</td>
<td>75</td>
</tr>
</tbody>
</table>
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APP</td>
<td>A posteriori probability</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive white Gaussian noise</td>
</tr>
<tr>
<td>BER</td>
<td>Bit error rate</td>
</tr>
<tr>
<td>BMST</td>
<td>Block Markov superposition transmission</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary phase-shift keying</td>
</tr>
<tr>
<td>EH</td>
<td>Extended Hamming</td>
</tr>
<tr>
<td>EXIT</td>
<td>Extrinsic information transfer chart</td>
</tr>
<tr>
<td>FHT</td>
<td>Fast Hadamard Transform</td>
</tr>
<tr>
<td>LLRs</td>
<td>Log-likelihood ratios</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a Posteriori</td>
</tr>
<tr>
<td>SBME</td>
<td>Superposition block Markov encoding</td>
</tr>
<tr>
<td>SISO</td>
<td>Soft-input-soft-output</td>
</tr>
<tr>
<td>SPC</td>
<td>Single parity check</td>
</tr>
<tr>
<td>RM</td>
<td>Reed-Muller</td>
</tr>
</tbody>
</table>
Abstract

This thesis proposed two improved channel coding schemes for digital communication systems, featuring in multiple code rates and low error floors. The first scheme is based on the well-known serially concatenated codes, while the other is based on the recently proposed block Markov superposition transmission (BMST) scheme.

Serial concatenation of Hamming codes and an accumulator has been shown to achieve near capacity performance at high code rates. However, these codes usually exhibit poor error floor performance due to their small minimum distances. To overcome this weakness, we propose to replace the outer Hamming codes by product codes constructed from Hamming codes and single-parity-check (SPC) codes. In this way, the minimum distance of the outer code can be doubled, which is expected to increase the minimum distance of the serially concatenated code, thus improving the error floor performance. Moreover, the code rate can be adjusted by using different SPC codes. Three-dimensional EXIT chart is used for their convergence analysis and the found thresh-
olds are shown to approach Shannon limit closely. Averaged ensemble distance spectra of the proposed codes is also calculated and compared with the original code. Simulation results show that the proposed codes can lower the error floor by two orders of magnitudes without waterfall performance degradation at short block length.

Block Markov superposition transmission (BMST) is a recently proposed channel coding scheme for the construction of big convolutional codes from some short codes. This thesis investigates a new class of low complexity multiple-rate codes based on the recently proposed BMST scheme using the first order Reed-Muller (RM) and extended Hamming (EH) codes. Compared with the multiple-rate codes based on BMST of repetition and single-parity-check (RSPC) codes, the new codes require much smaller encoding memories (and the number of interleavers used in the encoder) to achieve the same coding gain. Moreover, the decoding of BMST-RMEH codes has lower computational complexities (approximately half that of BMST-RSPC codes) and faster convergence speed.
1.1 Background

Nowadays, the demand for reliable and high speed information transmission and data storage systems has been increasing. This demand reflects in the information exchange, process and storage of data networks with high speed and large-scale. These systems have to be designed with some special technologies to make sure that the transmitted data can be reproduced reliably [1].

In 1948, Claude Shannon attempted to take up the pioneering research of reliable data transmitting over noisy channel [1]. His main idea showed that when the data transmission rate is below the channel capacity, arbitrary small probability of bit errors can be achieved by using powerful channel coding techniques. Channel coding techniques can enhance the decoding efficiency by introducing redundancy. The redundancy enables the decoder to detect and correct errors. Thus, it can protect the transmitted data from channel impairment. After Shannon’s work, channel coding technology rapidly developed.
In 1940s, Richard Hamming discovered the first class of linear block codes, which was later published in 1950 and now called Hamming codes in his honor. These codes could detect and correct single bit error. The extended versions of these codes are called extended Hamming codes, which could detect double bits errors and correct single bit error. For historical, practical and theoretical reasons, these codes have important meaning to this day.

The proposition of iterative decoding of concatenated codes, the so-called turbo codes in 1993, is an important breakthrough of approaching the channel capacity [2]. In an iterative fashion, the component decoders of turbo decoder exchange soft extrinsic information. The idea of iterative decoding is valid in a more general sense, and the turbo principle [3] can be applied to the receiver of a communication system.

Recently, a novel technique called block Markov superposition transmission (BMST) [4], [6], [29] is proposed to construct big convolutional codes from short codes, which can be viewed as a serial concatenation of a basic code and a forward convolutional code with large constraint length. A distinguished feature of BMST is that the performance of a BMST system can be simply lower-bounded in terms of the encoding memory and the
performance of the basic code, which can be easily obtained via simula-
tion or bounding techniques. Interestingly, simulation results show that the
lower bounds can be well matched in the low bit-error-rate (BER) region.
Hence, aided by BMST, it is possible to approach the channel capacity
at any given target BER of interest for any short code. Based on BMST,
a class of fixed-length multiple-rate codes, called BMST-HT codes, was
proposed in [28], where the basic code is chosen to be short Hadamard
transform (HT) codes whose code rate can be easily adjusted by setting
the number of frozen bits. Later, still based on BMST, an even more flex-
ible and simpler construction of multiple-rate codes was proposed in [7],
where the basic code is a mixture of repetition (R) codes and single-parity-
check (SPC) code.

1.2 Research Motivation and Objectives

In the practical communication systems, the channel is time varying
and the data to be transmitted has different error protection needs in many
cases. In such situations, codes with varying rates are required. This re-
search aims to construct some multiple-rate codes with low error floors
based on linear block codes. To achieve these goals, this thesis introduces
the product codes into the serial concatenation. The code rate of a prod-
uct code is the product of two component codes’ code rates, as well as the minimum distance of the product code. Therefore, the code rate can be adjusted by its component codes, and the minimum distance can be greatly improved. On the other hand, high-rate serially concatenated codes with Hamming codes as the outer codes and an accumulator as the inner code has been shown to achieve near capacity performance in the waterfall region [5]. However, since the outer Hamming codes have minimum distance 3, the resulting serially concatenated codes usually have rather small minimum distances, thus leading to poor error floor performance. In Chapter 3, we use Hamming codes and SPC codes to form the multiple-rate outer codes and expect to achieve good error floor performance.

BMST is a recently proposed data transmission technique which has good advantages on bit error rate (BER) performance [6]. The BER performance can be controlled by changing its memory length. In a previous study [7], the basic codes of BMST are replaced by multiple-rate codes consisting of Repetition code and single-parity-check (SPC) code. These codes have flexible code rates, however, they require large memory lengths to achieve desired coding gains, thus resulting in high decoding complexity. Chapter 4 tries to find a new construction of basic codes to reduce the
memory lengths and hence the decoding complexity.

1.3 Research Contributions

In Chapter 3, this thesis proposes to enhance the outer Hamming codes of Hamming-Accumulate (HA) codes by using SPC codes. More specifically, the outer codes are replaced by product codes with Hamming codes and SPC codes as the two component codes. In this way, the minimum distance of outer code can be doubled, which is expected to increase the minimum distance of serially concatenated code and to improve error floor performance. The product codes can be seen as multiple-rate codes by using different SPC codes. Simulation results show that the proposed codes can lower the error floor by two orders of magnitudes without waterfall performance degradation at short block length.

Chapter 4 investigates a new class of low complexity multiple-rate codes based on the recently proposed BMST scheme. These codes, called RMEH codes, consist of first order Reed-Muller (RM) codes and extended Hamming (EH) codes. Compared with the multiple-rate RSPC codes in [7], the new codes require much smaller encoding memories (or equivalently the number of interleavers used in the encoder) to achieve the same coding gain. Moreover, the decoding of RMEH-BMST codes has lower compu-
tational complexity (approximately half that of BMST-RSPC codes) and faster convergence speed.

1.4 Thesis Organisation

This thesis is organised as follows:

Chapter 1 gives an introduction about the thesis background and related research contributions.

Chapter 2 presents the basic knowledges and specific techniques which are closely related to the research. Firstly, some related linear block codes and their decoding algorithms are introduced. Secondly, we explain the construction of product codes and the encoding/decoding of accumulator. Serial concatenated codes are briefly reviewed. Finally, we detail the construction of BMST scheme.

Chapter 3 gives a detailed introduction of (extended) Hamming-SPC-Accumulate codes (HSA). It firstly presents the encoder and decoder structures. Then, three-dimensional EXIT chart is used for their convergence analysis and low weight distance spectrum of the proposed code is also calculated. Numerical results are presented at the end of Chapter 3.

Chapter 4 focuses on BMST systems. At the beginning, we introduce the proposed basic codes (RMEH) and compare the theory parameters with
RSPC codes. Then, we compute the decoding complexities of both RMEH and RSPC codes. Numerical results are presented to demonstrate the effectiveness of the proposed basic codes.

Chapter 5 makes a summary for all the research works.

1.5 Publications


Chapter 2

Literature Review

This Chapter presents the basic knowledges of this research. The research involves four types block codes which are reviewed in Section 2.1 (repetition codes, SPC codes, RM codes and Hamming codes). Then, an optimal decision rule called maximum a posteriori probability (MAP) decoding is presented in Section 2.2. In Section 2.3, product codes, which play an important role in this research, are introduced. Section 2.4 and Section 2.5 give a detailed description of accumulator and serially concatenated codes, respectively. Finally, the BMST coding scheme and its multiple-rate basic codes are investigated in Section 2.6. In current data communication systems and storage systems, most applications are coded into the binary digits 1 and 0. Therefore, all the researches are discussed in the binary field $\mathbb{F}_2$.

2.1 Type of Codes

In the common use of present data transmission, there are two different types of codes: block codes and convolution codes. The information sequence of a block code is divided into several information blocks.
with the same length $k$. We represent the information block by $u = (u_0, u_1, \ldots, u_{k-1})$, where $u_i$ denotes the $i$-th information bit. In binary field, the total of different possible information blocks is $2^k$. The encoder of a block code transforms an information block into an $n$-tuple $v = (v_0, v_1, \ldots, v_{n-1})$, which is the so called codeword. Thus, the total number of different possible codewords is $2^k$, corresponding to the total number of possible information blocks. We call the set of $2^k$ codewords an $(n, k)$ block code with a code rate $R = k/n$. The code rate interprets the transmission efficiency of a code. As the $n$-bit codeword only depends on the corresponding $k$-bit information block, the code rate is $R = k/n$ and the encoder is memoryless (each information block is encoded independently).

As a useful binary code, the code rate $R$ is less than 1 or the information length $k$ is less than codeword length $n$. When $k$ is less than $n$, we add $n-k$ bits to each information block to form a codeword and the added bits are called redundancies (or redundant bits). These redundant bits protect the code against the channel noise. We can strengthen the protection capability of a code by reducing the code rate $R$. If the code rate is fixed, we can add more redundant bits by increasing information length $k$ and codeword
length $n$. The main problem of designing a code is the manner of arranging the redundancies to achieve a reliable information transmission.

For a convolutional code, the input information sequence of the encoder is $k$-bit information block $u$ and output coded sequence is $v$. The difference between convolutional code and block code is that each coded information sequence of convolutional code not only depends on the corresponding $k$-bit information block but also depends on the previous information blocks. The memory length of the encoder is denoted by $m$. The code can be seen as the set of all the possible output coded blocks from the encoder. As the encoder includes memory, it must be achieved with a sequential logic circuit [8].

In the binary field, if the code rates of binary convolutional codes are less than 1, redundant bits are added. Typically, we can add more redundant bits without changing the code rate. It can be achieved by increasing the memory length $m$. The main problem for a convolutional code design is the manner by which to employ memories.

### 2.2 Linear Block Code

The linear block code is a subclass of block codes. For a binary block code $C$ with codeword $c \in C$, if the modulo-2 sum of any two codewords
is also a codeword in $C$, we call the code a linear block code. We choose a linear block code because the encoding complexity can be greatly reduced than nonlinear code [1].

Generally, we use the generator matrixes and parity-check matrixes to define a linear block code. We assume the input information block $u$ has fixed length $k$ and the output codeword $c$ has length $n$, the generator matrix $G$ of this $(n, k)$ linear block code is a $k \times n$ matrix and we have:

$$c = u \cdot G. \quad (2.1)$$

For a linear block code, each row of generator matrix is a codeword. More specifically, a generator matrix of a linear block code can be formed by any $k$ different codewords of the code.

A special structure of linear block code is called the linear systematic block code. The codewords of codes with systematic structure obtain $k$ bits unaltered information digits and $(n - k)$ bits parity-check digits, which are the linear sums of the information digits. The generator matrix of a linear systematic block code can be specified by:

$$G = [P | I_k], \quad (2.2)$$
where $P$ is the $k \times (n - k)$ generator matrix of parity-check digits and $I_k$ is a $k \times k$ identity matrix.

The parity-check matrix $H$ of an $(n, k)$ linear block code is an $(n-k) \times n$ matrix. Each row of $H$ is orthogonal to the rows of generator matrix $G$. Thus, we have

$$G \cdot H^T = 0,$$

or

$$c \cdot H^T = 0.$$

If we know the parity-check matrix $H$ of a linear block code, we can find its generator matrix $G$ by matrix transformation.

**EXAMPLE**

$$H = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}$$

This is a check matrix for a $(7,4)$ block code. We denote the parity-check digits by $p_i \ (0 \leq i \leq 2)$ and the information digits by $u_j \ (0 \leq j \leq 3)$. The columns with only one “1” display the position of parity-check digits in codewords. According to the check matrix, the codeword structure can be
written as $\mathbf{c} = [p_0, p_1, u_0, p_2, u_1, u_2, u_3]$ and the constraints between check digits and information digits are

$$\mathbf{c} \cdot \mathbf{H}^T = [0 \ 0 \ 0].$$

Then, it is clear that:

$$p_0 = u_0 \oplus u_1 \oplus u_3,$$

$$p_1 = u_0 \oplus u_2 \oplus u_3,$$

$$p_2 = u_1 \oplus u_2 \oplus u_3.$$

$\oplus = XOR$

Thus, we obtain the following:

$$P_T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \Rightarrow P = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The generator matrix is formed by parity-check digits’ generator matrix $P$ and identity matrix $I_4$. We use $P_i$ to denote the $i$-th column of $P$ and $I_j$ to denote the $j$-th column of $I_4$. Based on codeword structure $\mathbf{c} =$
\[ [p_0, p_1, u_0, p_2, u_1, u_2, u_3], \] the generator matrix can be written as follows:

\[
G = [P_0, P_1, I_0, P_2, I_1, I_2, I_3] = \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}.
\]

For an \((n, k)\) linear block code \(C\), its parity-check matrix \(H\) can be seen as the generator matrix of an \((n, n - k)\) linear block code \(C^\perp\). Any codeword in \(C^\perp\) is orthogonal to any codeword in \(C\). \(C^\perp\) is called the dual code of \(C\). The dual code of a linear code is also a linear code [9].

2.2.1 The Minimum Distance of a Block Code

The minimum distance (or minimum Hamming distance) of a block code is an important parameter. It is used to measure the capabilities of a block code, in other words, it relates to the error floor performance of a block code. The Hamming distance of two codewords is defined by the number of positions at which they are different.

**EXAMPLE**

\[ c_1 = (111111), c_2 = (000000), c_3 = (111110) \]
The Hamming distance between \( c_1 \) and \( c_2 \) is 7, whereas that between \( c_1 \) and \( c_3 \) is 1, and that between \( c_2 \) and \( c_3 \) is 6.

For linear block codes, we introduce the concept: Hamming weight. Hamming weight of codeword \( \mathbf{c} \) is the number of nonzero elements of \( \mathbf{c} \). As shown in the example above, the Hamming weight of \( c_1 \), \( c_2 \) and \( c_3 \) is 7, 0 and 6, respectively. The minimum distance of a linear block code is equal to the minimum Hamming weight of its nonzero codewords [1].

### 2.2.2 Maximum a Posteriori (MAP) Decoding Rule

The MAP decoding method for linear block codes separately minimizes the error probability for each coded symbol [10]. It is an optimum symbol-by-symbol “soft decision” decoding rule when codewords are equiprobable. Let \( C \) be an \((n, k)\) linear block code, and \( \mathbf{c} = (c_0, c_1, ..., c_{n-1}) \in C \) is the transmitted codeword, where \( c_j \) denotes the \( j \)-th element of \( \mathbf{c} \). We assume that the binary phase-shift keying (BPSK) modulation (with mapping 0 \( \rightarrow \) +1 and 1 \( \rightarrow \) −1) is used, and the modulated signal is transmitted over an additive white Gaussian noise (AWGN) channel. The received vector can be written as follows:

\[
y_j = (1 - 2c_j) + w_j, \tag{2.3}
\]
where \( y_j \) is the \( j \)-th element in the received vector \( y = (y_0, y_1, \ldots, y_{n-1}) \) and \( w_j \sim \mathcal{N}(0, \sigma^2) \) is the \( j \)-th sample of the AWGN. We have two probability distribution functions

\[
\begin{align*}
    f_0(y_j) &= P(y_j|0) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y_j-1)^2}{2\sigma^2}}, \\
    f_1(y_j) &= P(y_j|1) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y_j+1)^2}{2\sigma^2}}.
\end{align*}
\] (2.4)

The conditional probability \( P(y_j|\alpha) \) denotes the probability of transmitting \( \alpha \) (\( \alpha \in \mathbb{F}_2 \)) when \( y_j \) was received. It can be computed by the density functions. The conditional probability of receiving a vector \( y \) under the condition that a codeword \( c \) has been transmitted can be calculated by the product

\[
P(y|c) = \prod_{j=0}^{n-1} P(y_j|c_j).
\] (2.5)

For linear block code \( C \), all the codewords have the same transmitted probability \( \frac{1}{2^k} \). Our goal is to determine the \( j \)-th transmitted symbol. To achieve this, we estimate the conditional probabilities \( P(c_j = \alpha|y) \) for all \( \alpha \in \mathbb{F}_2 \) and choose the maximum one. Let \( C_j^{(\alpha)} \) be the set of all codewords whose
The $j$-th symbol is $\alpha$. Using Bayes rule and formula 2.3, we can get

$$P(c_j = \alpha|y) = \sum_{c \in C_j^{(\alpha)}} P(c|y)$$  \hspace{1cm} (2.6)

$$= \sum_{c \in C_j^{(\alpha)}} \frac{P(y|c)}{P(y)} p(c)$$

$$= \frac{1}{q^k P(y)} \sum_{c \in C_j^{(\alpha)}} \prod_{l=0}^{n-1} P(y_l|c_l).$$

The probability $P(c)$ is called a priori probabilities. And the probabilities $P(c|y)$ and $P(c_j = \alpha|y)$ are called a posteriori probabilities (APP). Therefore, this algorithms perform maximum a posteriori probability (MAP) decoding [11].

### 2.2.3 Log-Likelihood Algebra

Instead of probabilities, using log-likelihood ratios (LLRs) is convenient [12], and these are defined as follows:

$$L(c_j) = \log \frac{P(c_j = 0)}{P(c_j = 1)}$$  \hspace{1cm} (2.7)

$L(c_j)$ corresponds to the hard decided binary value and its magnitude represents a measure for the reliability of the hard decision. We can use LLRs to evaluating the bit probabilities as: $P(c_j) = \frac{1}{1+e^{-L(c_j)}} \cdot e^{-c_j \cdot L(c_j)}$ with $c_j = 0$ or 1.


2.2.4 Repetition Codes

A repetition code is an \((n, 1)\) linear block code with only 1 bit information. Thus, it only includes two \((2^1)\) codewords, the all-zero codeword and the all-one codeword. The encoder of a repetition code repeats the input information bit \(n\) times to form a codeword. The generator matrix can be written as

\[
G = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}.
\] (2.8)

We suppose that LLR information \(L(v_j)\) is available. The soft-input-soft-output (SISO) decoding algorithm computes the a posteriori messages

\[
L(u) = \sum_{j=0}^{n-1} L(v_j)
\]

for making decisions.

The decoding of repetition codes is very simple. The linear computational complexity is \(n\) times additions.

2.2.5 Single-Parity-Check Codes

SPC codes are \((k + 1, k)\) linear block codes with one bit parity-check digit. The parity-check digit is the modulo-2 sum of all the information bits. Therefore, the modulo-2 sum of all the coded bits is zero, which
means the Hamming weights of a SPC code is even. More specifically, the minimum distance of a SPC code is 2.

The generator matrix of a SPC code with parity-check bit at the last position is given by:

\[ G = [I_k | 1_k]. \]  

(2.9)

Where \( I_k \) denotes the \( k \times k \) identity matrix and \( 1_k \) denotes the \( k \times 1 \) all "1" matrix. From formula 2.10, the check matrix of the \((k + 1, k)\) SPC code can be found (i.e., \( H = [1, 1, \ldots, 1] \)) and it is the same as the generator matrix of \((k+1,1)\) repetition code. Thus, \((k + 1, k)\) SPC code and \((k + 1, 1)\) repetition code are dual codes to each other.

![Decoder structure of SPC code.](image)

The decoding of SPC code is based on boxplus operation, where \( boxplus(a, b) = \frac{1 + e^{a+b}}{e^a + e^b} \). Suppose all the LLR information \( L_j \) is available. A forward-backward decoding algorithm (when \( n > 3 \)) can be scheduled as algorithm 2.1.

There are a total of \( 3n - 6 \) times boxplus operations, and the decoding complexity of an \((n, k)\) SPC code is shown in table 2.1.
Algorithm 2.1 Decoding Algorithm of SPC codes

**Step 1: Initialization**

Let $LF_j$ and $LB_j$ be the forward feedback and backward feedback information in LLR form for $0 \leq j \leq n - 4$. All the forward and backward information is initialized to be 0.

For $n = 3$;

$L^E_0 = \text{boxplus}(L_1, L_2)$
$L^E_1 = \text{boxplus}(L_0, L_2)$
$L^E_2 = \text{boxplus}(L_0, L_1)$

For $n > 3$;

**Step 2: Forward information calculation:**

For $j = 0, LF_0 = \text{boxplus}(L_0, L_1)$;
For $j = 1, 2, \ldots, n - 4, LF_j = \text{boxplus}(LF_{j-1}, L_j)$

**Step 3: Backward information calculation:**

For $j = n - 4, LB_{n-4} = \text{boxplus}(L_{n-2}, L_{n-1})$;
For $j = n - 5, n - 6, \ldots, 0, LB_j = \text{boxplus}(LB_{j+1}, L_{j+2})$

**Step 4 Hard decision:**

$L^E_0 = \text{boxplus}(LB_0, L_1)$,
$L^E_1 = \text{boxplus}(LB_0, L_0)$,

For $j = 2, 3, \ldots, n - 3, L^E_j = \text{boxplus}(LF_{j-2}, LB_{j-1})$
$L^E_{n-2} = \text{boxplus}(LF_{n-4}, L_{n-1})$,
$L^E_{n-1} = \text{boxplus}(LF_{n-4}, L_{n-2})$. 
Table 2.1: The decoding complexity of SPC codes.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Addition</th>
<th>Multiplication</th>
<th>Logarithm</th>
<th>Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times</td>
<td>$6n - 12$</td>
<td>$6n - 12$</td>
<td>$3n - 6$</td>
<td>$6n - 12$</td>
</tr>
</tbody>
</table>

2.2.6 First Order Reed Muller codes

Reed-Muller codes are one of the oldest well-known multiple-error-correction codes. Muller discovered these codes in 1954 and, in the same year, Reed provided the first decoding algorithm [1].

An $r$-th order Reed-Muller code can be denoted by $\text{RM}(r,m)$, where $m$ and $r$ are integers with $0 \leq r \leq m$. Here, we just consider the first order Reed-Muller code $\text{RM}(1,m)$. A first order Reed-Muller code is a $(2^m, m + 1)$ linear code with minimum distance $2^{m-1}$. The generator matrix can be defined recursively:

i. for $m = 1$, $G_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$;

ii. for $m > 1$, $G_m = \begin{bmatrix} G_{m-1} & G_{m-1} \\ 0 & \ldots & 0 & 1 & \ldots & 1 \end{bmatrix}$.

EXAMPLE
\[
\mathbf{G}_2 = \begin{bmatrix}
1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 \\
\end{bmatrix}
\quad \text{and} \quad
\mathbf{G}_3 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix}.
\]

To reduce the encoding and decoding complexity, in our research work, we found a way to obtain the systematic first order Reed-Muller codes. We assume the information block is \( \mathbf{u} = (u_0, u_1, \ldots, u_m) \) and the codeword is \( \mathbf{c} \). The encoding is scheduled as Algorithm 2.2. For example, we assume \( \mathbf{u} = (u_0, u_1, u_2, u_3) = (0, 1, 0, 1) \). According to the encoding algorithm \((m = 3)\), we do following steps:

1. \( \mathbf{c} = (u_0, u_1) = (0, 1) \)
2. for \( i = 2 \), \( u_2 = u_0 \), so \( \mathbf{c} = (c, c) = (0, 1, 0, 1) \);
3. for \( i = 3 \), \( u_3 \neq u_0 \), so \( \mathbf{c} = (c, c \oplus 1) = (0, 1, 0, 1, 1, 0, 1, 0) \);

\begin{algorithm}
\caption{Encoding algorithm for RM codes}
\begin{algorithmic}
  \State 1. \( \mathbf{c} = (u_0, u_1) \)
  \State 2. for \((i = 2; i \leq m; i + +)\)
    \hspace{1em} if \((u_i = u_0)\)
    \hspace{2em} \( \mathbf{c} = (c, c) \);
    \hspace{1em} else
    \hspace{2em} \( \mathbf{c} = (c, c \oplus 1) \).
\end{algorithmic}
\end{algorithm}

The systematic bits will appear at the position \( i = 2^j \), where \( j \) is an integer and \( 0 \leq j \leq m \).
Fast Hadamard Transform (FHT)

Hadamard matrix is a very useful matrix that it can be used to decode first order Reed-Muller code rapidly. Let $H_N = [h_{i,j}]$ be the $N \times N$ Hadamard matrix, where $N = 2^m$ and $h_{i,j} = -1^{i_0 j_0 \oplus i_1 j_1 \oplus \cdots \oplus i_{m-1} j_{m-1}}$. $i_0, ..., i_{m-1}$ and $j_0, ..., j_{m-1}$ are the binary representations of the indices $i$ and $j$, respectively. When we use BPSK modulation, zeros are replaced by ones and ones are replaced by minus ones. We can see that half of the codewords are the rows of $H_N$ and the other half are their negations. $H_N$ also can be defined recursively as:

**i.** For $m = 1$, $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$;

**ii.** For $m > 1$, $H_N = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix}$.

In the decoding algorithm, fast Hadamard transform is used to compute the product $v \cdot H_n$. It requires only $n \log_2 n$ operations for FHT to get the result. To compute $(s_0, ..., s_{n-1}) = FHT(v_0, ..., v_{n-1})$, we need to perform the following steps $\log_2 n$ times.

For $j$ from 0 to $n/2 − 1$ compute:

$$s_j = v_{2j} + v_{2j+1}, \quad s_{n/2 - 1} = v_{2j} - v_{2j+1}.$$  

For $j$ from 0 to $n − 1$ assign:
The MAP decoding for RM(1,m) codes, which is based on the FHT transform, is proposed in [10]. Firstly, the probabilities $P(y|c)$ can be fast computed by using FHT. Secondly, soft decisions $P(c_j = 0|y) - P(c_j = 1|y)$ for all transmitted bit are computed after some auxiliary computations.

At the beginning of the algorithm, we compute the vector $b = (b_0, ..., b_{n-1})$, where

$$b_j = \ln P(y_j|0) - \ln P(y_j|1), j = 0, ..., n - 1.$$ 

Next, we compute the vector $t = (t_0, ... t_{n-1}) = FHT(b_0, ..., b_{n-1})$. Let

$$s = \sum_{j=0}^{n-1} \ln P(y_j|0) + \ln P(y_j|1).$$

Then, we compute the following:

$$r_0^i = \frac{s + t}{2} = \sum_{j=0}^{n-1} \ln P(y_j|c_{ij}),$$

and

$$r_1^i = \frac{s - t}{2} = \sum_{j=0}^{n-1} \ln P(y_j|c_{ij} \oplus 1).$$

We can compute the products of the probabilities as follows:

$$v_i^0 = e^{r_0^i} = \prod_{j=0}^{n-1} P(y_j|c_{ij}) = P(y|c_i)$$
and

\[ v_i^1 = e_i^1 = \prod_{j=0}^{n-1} P(y_j | c_{ij} \oplus 1) = P(y | c_i \oplus 1) \]

Now we have the probabilities \( P(y | c) \) after the first part of the algorithm.

In order to compute soft decisions \( P(c_j = 0 | y) - P(c_j = 1 | y) \), we form a vector \( x_i = P(y | c_i) - P(y | c_i \oplus 1) \). Using FHT, we can compute \( (w_0, ..., w_{n-1}) = FHT(x_0, ..., x_{n-1}) \), where

\[ w_j = 2^{m+1} Pr(y)(Pr(c_j = 0 | y) - Pr(c_j = 1 | y)). \]

\( 2^{m+1} Pr(y) \) is a constant factor, it can be computed as

\[ q = \sum_{i=0}^{n-1} (v_i^0 + v_i^1). \]

Finally, the soft decisions for all code bits can be calculated as follows:

\[ \frac{w_j}{q} = P(c_j = 0 | y) - P(c_j = 1 | y), j = 0, ..., n - 1. \]

Since the complexity of FHT is \( n \log_2 n \) times additions and all other steps have linear complexity, the decoding complexity is shown in the Table 2.2.
Table 2.2: The decoding complexity of RM codes.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Addition</th>
<th>Multiplication</th>
<th>Logarithm</th>
<th>Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times</td>
<td>$(3 + 2 \log_2 n)n$</td>
<td>$4n$</td>
<td>$-$</td>
<td>$n$</td>
</tr>
</tbody>
</table>

2.2.7 Hamming and Extended Hamming codes

For any positive integer $m \geq 3$, there exists a $(n, k, d_{\text{min}})$ Hamming code with codeword length $n = 2^m - 1$, information block length $k = 2^m - m - 1$ and minimum distance $d_{\text{min}} = 3$. The parity-check matrix $H$ of a $(n, k, d_{\text{min}})$ Hamming code consists of all the nonzero $m$-tuples as its columns exactly once. Any code with such a check matrix $H$ is a binary Hamming code with redundancy $m$. Now, we introduce a Hamming code with special construction. The check matrix of this Hamming code is called a lexicographic check matrix. For any positive integer $m$, the $i$-th column of lexicographic check matrix is the binary representation of the integer $i$ (with least significant digit at the bottom). For example, a lexicographic check matrix of $[7, 4]$ Hamming code is written as:

$$H_3 = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
\end{bmatrix}.$$
If we add an overall parity check bit to a binary Hamming code, we get
the extended version of this Hamming code. The minimum distance is in-creased to 4. The check matrix $XH$ of an extended Hamming code can be
constructed by adding an all “0” column at the beginning of $H$ and then
adding an all “1” vector at the bottom. For example:

$$XH_3 = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}.$$

We can see that the check matrix $XH_3$ above is equivalent to the generator
matrix $G_3$ of $RM(1, 3)$. In fact, an $(2^m, 2^m - m - 1)$ extended Hamming
code with the lexicographic check matrix is a dual code to first order Reed-
Muller code $RM(1, m)$.

As (extended) Hamming codes belong to high rate codes, they obtain
large number of codewords. Thus, the common decoding algorithms, which
consider all the codewords, have high decoding complexity. However, the
dual codes of high rate codes obtain fewer codewords. In order to reduce
the decoding complexity, we choose a dual code based symbol-by-symbol
MAP decoding algorithm [13].
Let $C^\perp$ be the dual of linear code $C$ and the codeword is denoted by $c'_{ij}$, i.e., $c'_{ij}$ is the $j$-th element of the $i$-th codeword of $C^\perp$. In [10], Ashikhmin and Litsyn developed a formula

$$P(c_j = 0|y) - P(c_j = 1|y) = \frac{\sum_{i=0}^{2^{n-k}-1} \prod_{l=0}^{n-1} \rho^{c'_{il} \oplus \delta_{jl}}}{\sum_{i=0}^{2^{n-k}-1} \prod_{l=0}^{n-1} \rho^{c'_{il}}},$$

where $\rho_j = P(c_j = 0|y_j) - P(c_j = 1|y_j)$ and $\delta_{ab} = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$.

Based on this formula, we can fast compute the probability difference of transmitting 0 and 1 using FHT and Walsh-Hadamard-type transform (WHT) [10].

The decoding complexity of Hamming codes is shown in Table 2.3.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Addition</th>
<th>Multiplication</th>
<th>Logarithm</th>
<th>Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times</td>
<td>$(11 + 4 \log_2 n)n$</td>
<td>$8n$</td>
<td>$n$</td>
<td>$3n$</td>
</tr>
</tbody>
</table>

### 2.3 Product Codes

Product codes have an important application in our research on serially concatenated codes. Its concept is not complicated and the main idea is to build long block codes by two short block codes. We can denote one short linear block code by $C_1$ with parameters $(n_1, k_1, d_1)$ and the other short
linear block code by $C_2$ with parameters $(n_2, k_2, d_2)$.

A product code $P$, which consists of $C_1$ and $C_2$, can be obtained by:

I. Putting information block (with length $k_1 \times k_2$) into a $k_1 \times k_2$ matrix.

II. Encoding the $k_1$ rows by $C_2$.

III. Encoding the $n_2$ columns by $C_1$.

It is clear that the product code $P$ is an $(n, k)$ block code with $n = n_1 \times n_2$ and $k = k_1 \times k_2$. So that, the code rate $R = \frac{k_1}{n_1} \times \frac{k_2}{n_2} = R_1 \times R_2$.

The most important feature is the minimum distance $d = d_1 \times d_2$ and we can build product code with very large minimum distance. For linear block codes, the modulo-2 sum of any two or more codewords is also a codeword.

Then, we determine that each row of product code $P$ is a codeword in code $C_2$ and each column of product code $P$ is a codeword in code $C_1$. The decoding algorithm of a product code is introduced in Chapter 3.
2.4 Accumulator

An accumulator is a rate-1 recursive encoder of the form $1/(1 \oplus D)$. We use accumulator as the inner code in serial concatenation and this code will not change the code rate. Let $u^t$ denotes the input information bit at time $t$. The coded bit can be defined as

$$c^t = u^t \oplus c^{t-1}.$$ 

Fig. 2.3 shows the decoder structure of the accumulator. The LLR information $L = (L_0, L_1, ..., L_{N-1})$ is computed by the information received from channel. $LF = (LF_0, LF_1, ..., LF_{N-2})$ and $LB = (LB_0, LB_1, ..., LB_{N-2})$ denote the forward and backward feedback information, respectively. The priori information $L^A = (L_0^A, L_1^A, ..., L_{N-1}^A)$ is the other input in iterative decoding. $L^E = (L_0^E, L_1^E, ..., L_{N-1}^E)$ denotes the output extrinsic informa-
tion. The decoding algorithm of an accumulator is shown in Algorithm 2.3.

**Algorithm 2.3 Decoding algorithm of accumulator**

**Step 1: Initialization**
For $j = 0, 1, \ldots, N - 1$, compute the LLR information $L_j = \ln \frac{P(y_j = 0)}{P(y_j = 1)}$

**Step 2: Forward information**:
For $j = 0$, $LF_0 = L_0^A + L_0$;
For $j = 1, 2, \ldots, N - 2$, $LF_j = boxplus(LF_{j-1}, L_j^A) + L_j$

**Step 3: Backward information**:
For $j = N - 2$, $LB_{N-2} = boxplus(L_{N-1}^A, L_{N-1})$;
For $j = n - 3, \ldots, 0$, $LB_j = boxplus(LB_{j+1} + L_j, L_j^A)$

**Step 4 Hard decision**:
For $j = 0$, $LE_0 = boxplus(LB_0, L_0)$,
for $j = 1, 2, \ldots, N - 2$, $LE_j = boxplus(LF_{j-1}, L_j + LB_{j-1})$
for $j = N - 1$, $LE_{N-1} = boxplus(LF_{N-2}, L_{N-1})$.

2.5 **Serially Concatenated Codes**

Serially concatenated codes are a kind of concatenated codes. The advantage of these codes is the waterfall region performance can be very close to the Shannon Limit with a very long overall code length when iterative decoding is used. The encoder of a serially concatenated code which includes an outer code encoder, an interleaver and an inner code encoder is shown in Fig. 2.4.

The Fig. 2.5 shows the decoder structure of the serially concatenated code. The SISO decoders of inner code and outer code are used in the
Figure 2.4: The encoding circuit of a serial concatenated code.

Figure 2.5: The decoder structure of a serial concatenated code.

iterative decoding. \( L_{\text{Inner},n}^A \) and \( L_{\text{Outer},n}^A \) denote the \( L \)-value a priori information for Inner code decoder and Outer code decoder, respectively. Similarly, \( L_{\text{Inner},n}^E \) and \( L_{\text{Outer},n}^E \) denote the generated extrinsic \( L \)-values of the two decoders. \( L_{C,n} \) is the \( n \)-th channel observation in \( L \)-value.

Notice that, the inner code needs to be recursive so that the interleaver gain can be obtained. In this way, the bit error rate can be extremely low with very long overall codeword length.

2.6 Block Markov Superposition Transmission

The BMST is a new type of coding scheme, which can be seen as a big convolutional codes construct from short codes. It is similar to superposi-
tion block Markov encoding (SBME) [6], which has been widely used to prove multiuser coding theorems. Over an AWGN channel, the BER performance of a BMST system can be lower-bounded according to the BER performance of its basic code and the memory length [7]. This is because BSMT system, with memory length $m$, provides an extra coding gain of $10 \log_{10}(m + 1)$ dB to the basic code. Thus, the error floor performance of BMST can be controlled by memory length and it can reach extremely low if the memory length is long enough. Specifically, the system complexity increases linearly with the increasing of memory length, however, the coding gain increases logarithmically with the increasing of memory length. Thus, it is not worth increasing memory length when it is large. Generally, we choose a target BER performance of $10^{-5}$ to decide the required memory length. At this target BER performance, the waterfall region of BMST can be close to Shannon limits (within one dB away). For BMST systems with different basic codes, the decoding complexity depends on the memory length and decoding complexity of the basic codes.

The encoder structure of BMST system with memory length $m$ is depicted in Fig. 2.6. $u^{(t)}$, $v^{(t)}$ and $c^{(t)}$ denote the information sequence, the coded basic code and the output codeword at time $t$, respectively.
“ENC” denotes the basic code encoder and the basic code of BMST is a B-fold Cartesian product of a short code $Q$ (codeword denoted by $q$). Thus, the input binary sequence $u^t$ includes $B$ small binary sequences, $u^t = (u^t_0, u^t_1, ..., u^t_{B-1})$ and basic code is $v^t = (q^t_0, q^t_1, ..., q^t_{B-1})$. Node “D” is the register, $\pi_i$ denotes the $i$-th random interleaver and the total number of interleaver is $m$. This BMST code can provide $10 \log_{10}(m + 1)$ dB coding gain over AWGN channel [7]. Assume the basic code require $\gamma_R$ dB to reach the target BER ($10^{-5}$) and the Shannon limit at code rate $R$ is $\gamma^*_R$ dB. In order to approach the Shannon limit, a memory length $m \approx 10 \frac{\gamma^*_R - \gamma_R}{10R}$ is necessary. The encoding algorithm is shown in Algorithm 2.4.

Notice that the termination in Step 3 will cause a code rate loss. However the rate loss can be negligible if $L$ is very large. The decoder structure of BMST codes with $L = 4$ and $m = 2$ is shown in Fig. 2.7. The decoding algorithm starts with dividing the whole decoder into layers. The $t$-th layer corresponds to the $t$-th transmitted basic code. We focus on computing
Algorithm 2.4 Encoding Algorithm for BMST

**Step 1: Initialization**
$v^{(t)}$ is initialized to be 0 for $t < 0$.

**Step 2: Loop:**
For $t = 0, 1, \ldots, L - 1$

a. The information sequence $u^{(t)}$ is encoded to a basic code $v^{(t)}$.

b. Interleave the basic code $v^{(t-i)}$ to $w^{(i)}$ by the $i$-th interleaver, from $i = 1$ to $i = m$.

c. The output codeword $c^{(t)} = v^{(t)} + \sum_{i=1}^{m} w^{i}$.

**Step 3: Termination:**
After $L$ binary sequences are coded, as a termination, we sent $m$ all zero sequences at the end of information sequence.

---

Figure 2.7: The factor graph representation of a BMST code with $L = 4$ and $m = 2$. 
the forward feedback information and backward feedback information be-
tween neighbouring layers. An iterative decoding algorithm called iterative
forward-backward decoding algorithm is presented in [6]. In each iteration
round, we compute the forward information from layer 0 to layer $L + m - 1$
and the backward information from layer $L + m - 1$ to layer 0. When a
predefined certain stopping criterion is satisfied, the hard decision is made
on the output log-likelihood information of basic code decoder. We denote
the $L$-value channel observation at time $t$ as $y^t = (y^t_0, y^t_1, ..., y^t_{n-1})$, where
$y^t_j$ is the $j$-th component of $y^t$. $V^t$ and $U^t$ are the corresponding LLRs of
$v^t$ and $u^t$. The decoding algorithm can be scheduled as follows:

**Step 1: Initialisation**

The $L$-value information over the intermediate edges is initialised to 0. Set
a maximum iteration number $I_{\text{max}} > 0$.

**Step 2: Iteration**

For $I = 1, 2, ..., I_{\text{max}}$

Forward recursion: for $t = 0, 1, ..., L + m - 1$, the forward feedback infor-
mation from $t$-th layer to $(t + 1)$-th layer can be calculated by

Backward recursion: for $t = L + m - 1, ..., 1, 0$, the backward feedback
information from $(t + 1)$-th layer to $t$-th layer can be calculated by

\[ \text{Step 3: Hard Decision} \]

After the predefined stopping criteria is reached, the hard decisions are made on $U^t$, for $t = 0, 1, ..., L - 1$.

Another algorithm called iterative sliding-window decoding with a fixed decoding delay $d$ is presented in [29]. The iterative sliding-window decoding algorithm is similar to iterative forward-backward decoding algorithm, but the iteration range is from $t$ to $t + d$. Thus, it does not require that all the basic codewords are received. The algorithm can be started when $d + 1$ basic codewords are received and they are decoded one by one.

The decoding complexity can be divided into two parts. One part is depending on the basic code and we denote this part by $O_{\text{basic code}}$. The other part includes all the remaining steps and the complexity of these
steps relates to the memory length. For each coded bit, the complexity is written as

\[ O_{\text{perbit}} = (4m - 2)\text{boxplus} + (3m + 3)\text{additions} + 2 \cdot O_{\text{basiccode}}. \] (2.10)
3.1 Introduction

High rate codes with low error floors are of interest for some applications where high data rates and low error probabilities are required, e.g., magnetic recording systems, optical communications [16], and some future wireless transmission systems [17]. Recently, a class of high-rate serially concatenated codes with Hamming codes as the outer code and an accumulator as the inner code, termed as HA codes (or exHA codes for extended Hamming outer codes), has been shown to achieve near capacity performance in the waterfall region [5], [16]. However, since the outer Hamming codes have minimum distance 3 (or 4 for extended Hamming codes), the resulting serially concatenated codes usually have rather small minimum distances, thus leading to poor error floor performance. For example, the minimum distances of HA codes are typically 2 or 3 when overall code length is 992 (see the analysis in Section 3.4). This weakness hinders its applications in systems where low error rate is expected.
such as optical communication systems and data storage devices [18]. One way to mitigate the weakness is to optimise the interleaver design [19]. Another approach is to append a second accumulator to HA codes and form double serially concatenated codes, termed as HAA codes [20]. It is shown that HAA codes have minimum distance growing linearly with the block length and thus they are expected to achieve very good error floor performance [6]. However, due to the serial concatenation with two accumulators, iterative decoding of HAA codes incurs a non-negligible loss at the convergence threshold. For example, using (31, 26) Hamming code as the outer code, the convergence thresholds of HA codes and HAA codes are $E_b/N_0 = 2.77$ dB and 3.48 dB, respectively [20]. This implies that the serial concatenation of a second accumulator leads to a threshold loss of 0.71 dB. Thus, how to balance the performance in error floor region and waterfall region is a critical issue in the design of high rate codes with iterative decoding [16].

To increase the minimum distances of HA codes while maintaining their good decoding thresholds, this Chapter proposes to enhance the outer Hamming codes by using high-rate SPC codes. More specifically, the outer codes are replaced by product codes [21] with Hamming codes and high-
rate SPC codes as the two component codes. The resulting serially concatenated codes are called HSA codes (or exHSA codes for extended Hamming codes). Using a high-rate SPC code as one component code, the product code can double the minimum distance of Hamming code and the code rate loss can be marginal.

The remainder of this Chapter is organised as follows. Section 3.2 gives a detailed description of the encoder and the associated iterative decoder for the proposed high rate codes. Three-dimensional EXIT charts are used for analysing the iterative decoding behaviour of the proposed codes and iterative decoding threshold are determined in Section 3.3. In Section 3.4, the low-weight distance spectrum of the proposed codes is calculated, and the simulation results are presented to confirm the analysis in Section 3.5.

3.2 HSA Codes: Encoder and Decoder

HSA codes is a class of serially concatenated codes with product codes (constructed from Hamming codes and SPC codes) as the outer code and an accumulator as the inner code. The encoder and decoder of HSA codes are detailed in the following two subsections, respectively.
3.2.1 Encoder

Figure 3.1(a) depicts the encoder structure of the serial concatenation of an outer code and an inner accumulator through an interleaver $\pi$. The use of Hamming codes and extended Hamming codes as outer codes has been considered in [16], [5]. Since the minimum distances of Hamming and extended Hamming codes are very small (3 and 4, respectively), the resultant serially concatenated codes generally exhibit small minimum distances, thus leading to high error floor performance. Here, we propose to use as the outer code the product code with Hamming and SPC component codes as shown in Fig. 3.1(b). The product code is depicted as an array, where each row is a Hamming code and each column is an SPC code. Compared to HA codes, the rates of HSA codes are reduced by a factor of $(n_r - 1)/n_r$, which is the code rate of the SPC code. It is easy to control the rate loss by adjusting the number of rows, $n_r$, in the code array. It is well known that the minimum distance ($d_{\text{min}}$) of a product code is the product of the $d_{\text{min}}$’s of its two component codes [1]. Moreover, the $d_{\text{min}}$ of SPC codes is 2. Thus, an advantage of using proposed product codes as the outer code is that the minimum distance of the outer code can be doubled. More specifically, the minimum distance of the outer code is
increased from 3 to 6 for the case of Hamming codes and from 4 to 8 for the case of extended Hamming codes.

![Diagram](image)

Figure 3.1: (a) Encoder structure of serially concatenated codes with an inner accumulator; (b) An outer product code with (extended) Hamming codes and SPC codes as component codes.

### 3.2.2 Decoder

From Fig. 3.1(b), each coded bit in the outer product code joins a Hamming code and an SPC code. After the outer product encoding, we can see from Fig. 3.1(a) that the coded bits of the outer product code are interleaved and then used as the input to the accumulator. Thus, each coded bit in the outer product code in fact joins 3 code constraints: a Hamming code, an SPC code, and the accumulator. Accordingly, the iterative decoder can be constructed by employing three soft-input/soft-output (SISO) decoders...
Figure 3.2: Iterative decoder of HSA codes with three constituent decoders. “Acc DEC”, “Ham DEC” and “SPC DEC” denote the accumulator decoder, Hamming decoder and SPC decoder, respectively. $\pi$ and $\pi^{-1}$ denote the interleaver and deinterleaver.

(i.e., Accumulator decoder, Hamming decoder and SPC decoder) as shown in Fig. 3.2. The SISO decoder for the inner accumulator can be efficiently conducted with low complexity by performing the forward-backward algorithm on its factor graph representation [11]. For SISO decoding of Hamming codes, we adopt the low complexity algorithm proposed in [10], which is based on the dual code decoding principle firstly developed by Hartmann and Rudolph in [13]. As the dual codes of extended Hamming codes are first order Reed-Muller codes whose symbol-by-symbol maximum a posteriori (MAP) decoding can be done with fast Hadamard transforms (FHTs), SISO decoding of (extended) Hamming codes can also be efficiently implemented by using FHTs. The SISO decoding of SPC codes
is exactly the same as the row decoding in LDPC codes, which can be implemented by the famous sum-product algorithm (see, e.g., [1]).

<table>
<thead>
<tr>
<th>Algorithm 3.1 Iterative decoding of HSA codes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1: Initialization</strong></td>
</tr>
<tr>
<td>Calculate the $L$-value, $L_{C,n}$, from channel observation</td>
</tr>
<tr>
<td>$L_{C,n} = \log \frac{P(y_n</td>
</tr>
<tr>
<td>and set all the extrinsic information to 0,</td>
</tr>
<tr>
<td>$L_{E_{Acc,n}}^E = L_{Ham,n}^E = L_{SPC,n}^E = 0, n = 0, 1, ..., N - 1$.</td>
</tr>
<tr>
<td><strong>Step 2: Iterative decoding</strong></td>
</tr>
<tr>
<td>At each iteration, the three constituent decoders are performed in a serial fashion.</td>
</tr>
<tr>
<td><strong>Accumulator decoder</strong>: Compute a priori information $L_{Acc,n}^A = L_{Ham,n}^E + L_{SPC,n}^E$. Then, input ${L_{Acc,n}^A}$ and ${L_{C,n}}$ to the accumulator decoder and generate extrinsic information ${L_{E_{Acc,n}}^E}$.</td>
</tr>
<tr>
<td><strong>Hamming decoder</strong>: Compute a priori information $L_{Ham,n}^A = L_{Acc,n}^E + L_{SPC,n}^E$. Then, input ${L_{Ham,n}^A}$ into the Hamming decoder and generate extrinsic information ${L_{Ham,n}^E}$.</td>
</tr>
<tr>
<td><strong>SPC decoder</strong>: Compute a priori information $L_{SPC,n}^A = L_{Ham,n}^E + L_{Acc,n}^E$. Then, input ${L_{SPC,n}^A}$ into the SPC decoder and generate extrinsic information ${L_{SPC,n}^E}$.</td>
</tr>
<tr>
<td><strong>Step 3: Decision</strong></td>
</tr>
<tr>
<td>Compute the $L$-value $L_{H,n} = L_{Acc,n}^E + L_{Ham,n}^E + L_{SPC,n}^E$. and make hard decision $\hat{x}_n$ for the $n$th bit as follows,</td>
</tr>
</tbody>
</table>
| $\hat{x}_n = \begin{cases} 
0 & L_{H,n} > 0 \\
1 & L_{H,n} \leq 0 
\end{cases}$. |
| If the maximum iteration number reached, stop decoding. Otherwise, go back to Step 2. |

Before we introduce the detailed decoding algorithm, it is necessarily to introduce some notations. $L_{Acc,n}^A$, $L_{Ham,n}^A$, and $L_{SPC,n}^A$ denote a priori information of $n$th bit in log-likelihood ratio form ($L$-value) [22] for accumulator decoder, Hamming decoder, and SPC decoder, respectively. Sim-
ilarly, $L_{Acc,n}^E$, $L_{Ham,n}^E$ and $L_{SPC,n}^E$ denote the generated extrinsic $L$-values of $n$th bit for the three decoders. The iterative decoding of the proposed codes can be performed by serially activating the three component SISO decoders, i.e., “Acc DEC”, “Ham DEC” and “SPC DEC” as shown in Fig. 3.2. The detailed decoding algorithm is summarised in Algorithm 3.1.

3.3 Threshold Analysis Via Three-Dimensional EXIT Chart

The convergence behavior of iteratively decoded systems can be accurately analyzed by using the density evolution (DE) algorithm [27]. However, as DE tracks the evolution of probability density functions (pdfs) of soft information, its computational complexity is very high. A simplified version of DE, referred to extrinsic information transfer (EXIT) chart, is proposed in [22], which uses mutual information as the surrogate of pdfs. The input-output relations of constituent decoders are depicted by EXIT functions which characterizes how a priori information transfer into extrinsic information at the SISO decoder. A decoding trajectory for the exchange of extrinsic information between constituent decoders can be visualised in an EXIT chart.

For iterative decoding systems with two component decoders, each decoder can be characterised by an EXIT function, which is usually obtained
via simulation with the assumption that the a priori decoder input follows
the symmetric Gaussian distribution [22]. Graphically, an EXIT function
can be visualised as a curve in the EXIT chart. Notice that as the extrin-
sic information from one decoder is used as the priori information for the
other decoder, the EXIT curve for the second decoder can be drawn in the
same chart for the first decoder by swapping the axes. In this way, the
convergence behavior of the iteratively decoded system with two compo-
nent decoders can be visualised by the decoding trajectory between the two
EXIT curves [22].

Later, the EXIT chart tool is further extended for the analysis of three-
dimensional parallel concatenated system by Ten Brink in [23] and three-
dimensional serially concatenated system by Tüchler in [26]. From the
encoding perspective, the proposed HSA codes can be viewed as a Hybrid
concatenation scheme, the product code and the accumulator are serially
concatenated, while the product code itself can be viewed as a parallel con-
catenation. However, as mentioned in Subsection 3.2.2 each coded bit in
the outer product code joins 3 code constraints and then an HSA code can
be treated as a parallel concatenated code by viewing the product codeword
as the “input”. In fact, the proposed decoder as shown in Fig. 3.2 has the
same structure as that for parallel concatenated code (see Fig. 3.2 in [22]). Hence, the three-dimensional EXIT chart developed for parallel concatenated codes in [23] can be adopted for the analysis of the proposed codes. As seen from Fig. 3.2, for a three-dimensional parallel concatenated code, each constituent decoder has two inputs and one output, which means the associated EXIT function is a two-input and one-output function, and is visualised as a surface rather than a curve in the case of two-dimensional EXIT chart. Now we use the Hamming decoder as an example to explain how to generate the EXIT surface for a constituent decoder. The EXIT function denoted as \( I_{Ham}^E = f_{Ham}(I_{Acc}^E, I_{SPC}^E) \), where \( I_{Acc}^E \), \( I_{Ham}^E \) and \( I_{SPC}^E \) denote the mutual information which are related to the extrinsic information generated by accumulator decoder, Hamming decoder and SPC decoder, respectively. To approximate the function, we only need a fine grid of the \((I_{Acc}^E, I_{SPC}^E)\) over the area \([0, 1]^2\), and for each point \((I_{Acc}^E, I_{SPC}^E)\) in the grid simulation is required to find the associated \( I_{Ham}^E \). The detailed procedure is similar to that in [22].

As an example, Fig. 3.3 shows the three-dimensional EXIT chart at \( E_b/N_0 = 3.19 \) dB for the HSA code with (31, 26) Hamming codes and (32, 31) SPC component codes. There are three surfaces in the three di-
Figure 3.3: Three-dimensional EXIT chart for Hamming(31,26)-SPC(32,31)-Accumulate code at $E_b/N_0 = 3.19$ dB

mensional EXIT chart; each surface corresponds to the extrinsic mutual information transfer characteristic of a constituent decoder which accepts the priori knowledge from other two decoders. To insure that successful decoding, it is necessary to guarantee that the trajectory can go up to $(1, 1, 1)$. Equivalently, a tunnel from $(0, 0, 0)$ to $(1, 1, 1)$ is required. Otherwise, the trajectory will get stuck and decoding cannot converge to the correct codeword. The threshold is the minimum $E_b/N_0$-value at which a tunnel from $(0,0,0)$ to $(1,1,1)$ is possible. With the help of the three dimensional EXIT chart, we can easily identify the thresholds of HSA codes by gradually tuning the value of $E_b/N_0$. As an example, the iterative decoding thresh-
old of HSA codes with the outer product code constructed from (31,26) Hamming code and (32,31) SPC code is found to be $E_b/N_0 = 2.80$ dB, which is better than the threshold, $E_b/N_0 = 3.48$ dB, of the HAA codes. Notice that the corresponding Shannon limit is $E_b/N_0 = 2.2$ dB. Hence, the proposed HSA code is about 0.6 dB away from the Shannon limit.

### 3.4 Low Weight Profile Analysis

As the error floor performance is largely determined by low weight codewords, this section examines and compares the low weight distance spectra of the proposed codes and the existing ones. For comparison purpose, we compute the low weight profiles for 3 length-992 code ensembles, i.e., HA code ensemble with (31,26) Hamming outer code, exHA code ensemble with (32, 26) extended Hamming outer code, and HSA code ensemble with the product code constructed from (31,26) Hamming code and (32,31) SPC code as the outer code. Note that the HSA code ensemble has the same code rate as the exHA code ensemble. Using the uniform interleaver concept [24], the ensemble-average weight enumerator (WE) of the code ensemble can be computed as:

$$A_h = \sum_w A_{C_0} A_{A_{cc}} \frac{A_{w,h}}{\binom{N}{w}},$$

(3.1)
where $A_h$ denotes the ensemble-average number of codewords of weight $h$, $A_{w}^{C_0}$ is the WE of the outer code $C_0$ and $A_{w,h}^{ACC}$, the input-output WE of the inner accumulator, can be written in closed form as [1]:

$$A_{w,h}^{ACC} = \binom{N - h}{\lfloor w/2 \rfloor} \left( \frac{h - 1}{\lceil w/2 \rceil - 1} \right)$$

(3.2)

The ensemble-average low-weight profiles are summarised in Table 3.1. From Table 3.1, we can see that the ensemble-average number of weight-4 codewords for HA codes is 8.5271, which implies that a randomly generated HA code has a high probability of having a minimum distance no greater than 4. In fact, 20 length-992 HA codes are constructed by randomly generating 20 interleavers and the triple impulse method in [20] was used to determine their minimum distances ($d_{\text{min}}$), among which 9 codes were found to have $d_{\text{min}} = 2$, 10 were found to have $d_{\text{min}} = 3$, and only 1 has $d_{\text{min}} = 4$. Similar results are observed for exHA codes. However, when the outer code is replaced by the product code with (31,26) Hamming code and (32,31) SPC code as component codes, the values of the low-weight profile of HSA codes are more than two orders smaller than those of HA codes and exHA codes. In this case, the triple impulse method fails to find $d_{\text{min}}$ with reasonable values. This implies that HSA codes could
Table 3.1: Ensemble-average low-weight profiles of 4 length-992 code ensembles with (31,26) Hamming code or (32,26) extended Hamming code as component codes. $A_h$ is the average number of codewords with Hamming weight $h$.

<table>
<thead>
<tr>
<th>Code</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA code</td>
<td>0</td>
<td>0.4541</td>
<td>2.4750</td>
<td>8.5271</td>
</tr>
<tr>
<td>exHA code</td>
<td>0</td>
<td>0.4692</td>
<td>1.0427</td>
<td>3.0685</td>
</tr>
<tr>
<td>HSA code</td>
<td>0</td>
<td>0</td>
<td>0.0095</td>
<td>0.0293</td>
</tr>
<tr>
<td>HAA code</td>
<td>0.0009</td>
<td>0.0010</td>
<td>0.0012</td>
<td>0.0013</td>
</tr>
</tbody>
</table>

Table 3.2: The parameters of 5 serially concatenated codes.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HA code</td>
<td>992</td>
<td>832</td>
<td>(31,26)</td>
<td>–</td>
</tr>
<tr>
<td>exHA code</td>
<td>992</td>
<td>806</td>
<td>(32,26)</td>
<td>–</td>
</tr>
<tr>
<td>HAA code</td>
<td>992</td>
<td>832</td>
<td>(31,26)</td>
<td>–</td>
</tr>
<tr>
<td>HSA code</td>
<td>992</td>
<td>806</td>
<td>(31,26)</td>
<td>(32,31)</td>
</tr>
<tr>
<td>exHSA code</td>
<td>992</td>
<td>780</td>
<td>(32,26)</td>
<td>(31,30)</td>
</tr>
</tbody>
</table>

have larger $d_{min}$’s and much better error floor performance could be expected. For comparison, we also include in Table 3.1 the low-weight profile for the length-992 HAA code ensemble with (31,26) Hamming outer code. Although the HAA code ensemble has even smaller values at the low-weight profile, it incurs a non-negligible performance in the waterfall region as stated above. In fact, at short block lengths, we find HSA codes could provide similar error floor performance as HAA codes, as will be shown in Fig. 3.4.
3.5 Numerical Results

To verify the above analysis, 5 length-992 serially concatenated codes are constructed by using random interleavers. The code parameters of the constructed 5 serially concatenated codes are listed in Table 3.2. Figure 3.4 compares their frame error rate (FER) performance on the AWGN channel with BPSK modulation. The maximum iteration number is set to 30. For each simulation point at least 50 frame errors are collected. As expected, due to their small minimum distances, the HA and exHA codes exhibit poor FER performance; the FER error floors appear around FER of $10^{-3}$. Although the HAA code outperforms the HA and exHA codes at
the error floor region, a degradation of about 0.5 dB in the waterfall region is observed. The HSA code achieves much better error floor performance compared to HA and exHA codes and its FER curve tends to show a similar slope as that of the HAA code at the high SNR region. The exHSA code achieves the best performance, which is achieved at the cost of a slight code rate reduction.

![Frame Error Rate Performance](image)

Figure 3.5: Frame error rate performance of three length-992 HSA codes using (8,7), (16,15) and (32,31) SPC codes, respectively.

In Fig. 3.5, the FER performances of three length 992 HSA codes are depicted. Their outer codes are constituted by (31, 26) Hamming code and (8,7), (16, 15) and (32,31) SPC codes, respectively. Thus, they have the same codeword length but different code rates (0.7339, 0.7863 and
0.8125). Compared with HSA code using (32, 31) SPC code, the HSA code using (16, 15) SPC code has a code rate loss of 0.0262. However, this code rate loss lead to a 2-orders of magnitude improvement at error floor region. For HSA code using (8,7) SPC code, the code rate loss is 0.0786 and the improvement is at least 4-orders of magnitude.

3.6 Conclusion

Low-weight profile analysis has revealed that randomly generated HA and exHA codes have high probabilities of producing low weight code-words, which are responsible for their poor error floor performance. To overcome this weakness, we have proposed to replace the outer codes in HA and exHA codes with product codes from (extended) Hamming codes and high-rate SPC codes. Such a replacement maintains the good waterfall performance of HA and exHA codes, while the minimum distance of the outer code is doubled, thus leading to much better error floor performance.
Chapter 4

Block Markov Superposition Transmission with

Multiple-Rate Basic Codes

4.1 Introduction

In BMST systems, some multiple-rate codes have already been used. Short Hadamard transform (HT) code, in [28], is a new class of multiple-rate code with fixed code length. Its generator matrix is an $N \times N$ Hadamard matrix, where $N = 2^p$ with $p > 0$. The code rate can be adjusted by setting different number of frozen bits. A more flexible and even simpler construction of multiple-rate codes, called RSPC codes, is proposed in [7]. This is a family of codes composed by repetition codes and SPC codes. The code rate can be adjusted by changing the proportion of repetition codes and SPC codes. Since the code rates of repetition code and SPC code are $\frac{1}{N}$ and $\frac{(N-1)}{N}$, the constructed RSPC codes have a wide code rate range from $\frac{1}{N}$ to $\frac{(N-1)}{N}$. More conveniently, the performance of RSPC codes can be predicted analytically, as is required for determining the memory length of the BMST-RSPC codes. On the other hand, this RSPC-BMST system has two disadvantages. One is the system requires very long memory length
to guarantee the BER performance and the other is the decoding of SPC codes involves a large number of logarithm and exponent operations. Logarithm and exponent operations have higher complexity than other operations such as addition, subtraction and multiplication. This leads to high decoding complexity on RSPC codes. As mentioned before, BMST system needs very long basic code length (at least $10^4$) to guarantee its BER performance. To reduce the effect of rate loss, the number of basic code $L$ is always over 1000. This means that the overall code length of BMST system is over $10^7$. Thus, the high decoding complexity will be a serious problem in the simulation.

In this Chapter, we intend to reduce the complexity of RSPC-BMST system by changing the basic codes. Since the performance of basic codes depends on the high rate codes, we consider using extended Hamming (EH) codes to replace SPC codes and using its dual codes, first order RM codes, as the low rate codes. We denote these codes as RMEH codes. With the same codeword length and code rate, the BER performance of RMEH code is better than RSPC code. Therefore, RMEH-BMST system requires fewer memory length than RSPC-BMST codes. On the other hand, the decoding complexity of RMEH codes is only half of RSPC codes’. We
expect the overall decoding complexity of BMST system can be reduced by half.

4.2 The Composition of RMEH Codes

A multiple-rate codes $C(N, K)$ can be formed by two codes, which have different code rates but the same codeword lengths. We denote the low rate code as $C_{low}(n, k_1)$ and the high rate code as $C_{high}(n, k_2)$. The code rate range of $C$ is from $\frac{k_1}{n}$ to $\frac{k_2}{n}$. Code $C$ can be described mathematically by a Cartesian product [7], which means the $C_{low}$ is used $\alpha$ times while $C_{high}$ is used $\beta$ times. The code rate can be changed by adjusting the ratio $\alpha : \beta$.

We denote the codeword length $N = n(\alpha + \beta)$ and the information block length $K = \alpha k_1 + \beta k_2$. The code rate $\frac{k}{n}$ can be calculated as follows:

$$\frac{k}{n} = \frac{\alpha}{\alpha + \beta} \cdot \frac{k_1}{n} + \frac{\beta}{\alpha + \beta} \cdot \frac{k_2}{n}. \quad (4.1)$$

Generally, we set $\alpha + \beta = k_2 - k_1$, then formula 4.1 can be simplified by:

$$\frac{k}{n} = \frac{k_1 + \beta}{n} = \frac{k_2 - \alpha}{n}. \quad (4.2)$$

In formula 4.2, the code rate will be changed by $\frac{1}{n}$ when $\beta$ (or $\alpha$) is changed by 1.

For example, we choose $(32, 6)$ first order RM code and $(32, 26)$ ex-
Table 4.1: The parameters for RMEH codes with n=32.

<table>
<thead>
<tr>
<th>Rate</th>
<th>3/16</th>
<th>4/16</th>
<th>5/16</th>
<th>6/16</th>
<th>7/16</th>
<th>8/16</th>
<th>9/16</th>
<th>10/16</th>
<th>11/16</th>
<th>12/16</th>
<th>13/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>$\gamma^*(dB)$</td>
<td>-5.3</td>
<td>-3.8</td>
<td>-2.6</td>
<td>-1.6</td>
<td>-0.7</td>
<td>0.2</td>
<td>1.0</td>
<td>1.8</td>
<td>2.6</td>
<td>3.4</td>
<td>4.3</td>
</tr>
<tr>
<td>$\gamma(dB)$</td>
<td>1.9</td>
<td>8.34</td>
<td>8.34</td>
<td>8.34</td>
<td>8.34</td>
<td>8.35</td>
<td>8.36</td>
<td>8.36</td>
<td>8.36</td>
<td>8.36</td>
<td>8.36</td>
</tr>
<tr>
<td>$m$</td>
<td>5</td>
<td>14</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Extended Hamming code to constitute a RMEH code, the parameters are shown in Table 4.1. The sum of $\alpha$ and $\beta$ is 10 and the code rate range is from 3/16 to 13/16. $\gamma$ denotes the required SNR for RMEH code to reach a target BER of $10^{-5}$ and $\gamma^*$ denotes the corresponding Shannon limit. As introduced before, the coding gain over AWGN channel can be calculated by $10 \log_{10}(m + 1)$. Thus, the required memory length is $m \approx 10^{\frac{2-\gamma^*}{10}} - 1$.

From table 4.1, we can see the values of $\gamma$ are all around 8.36 dB (except at rate 3/16). This is due to the performance of a RM code is much better than a EH code with the same codeword length. Thus, the performances of RMEH code are limited by extended Hamming code.

Table 4.2 shows the parameters of a RSPC code, which is formed by (32, 1) repetition codes and (32, 31) SPC codes. To make a fair comparison, the code rate range of RSPC code is the same with RMEH code in Table 4.1. We can see that the performances of RSPC code are limited by SPC codes (the values of $\gamma$ are all around 10.9 dB). Compared with
Table 4.2: The parameters for RSPC codes with n=32.

<table>
<thead>
<tr>
<th>Rate</th>
<th>3/16</th>
<th>4/16</th>
<th>5/16</th>
<th>6/16</th>
<th>7/16</th>
<th>8/16</th>
<th>9/16</th>
<th>10/16</th>
<th>11/16</th>
<th>12/16</th>
<th>13/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>25</td>
<td>23</td>
<td>21</td>
<td>19</td>
<td>17</td>
<td>15</td>
<td>13</td>
<td>11</td>
<td>9</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>β</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
<td>21</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>$\gamma^*$ (dB)</td>
<td>-5.3</td>
<td>-3.8</td>
<td>-2.6</td>
<td>-1.6</td>
<td>-0.7</td>
<td>0.2</td>
<td>1.0</td>
<td>1.8</td>
<td>2.6</td>
<td>3.4</td>
<td>4.3</td>
</tr>
<tr>
<td>$\gamma$ (dB)</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
<td>10.9</td>
</tr>
<tr>
<td>m</td>
<td>41</td>
<td>29</td>
<td>21</td>
<td>17</td>
<td>13</td>
<td>11</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

RMEH codes, there is a $10.9 - 8.36 = 2.54$ dB gap at each code rate (except at rate 3/16). This is due to the performance of SPC codes is worse than extended Hamming codes. At each code rate (except at rate 3/16), the memory length of RSPC code is about twice as much as the memory length of RMEH codes.

### 4.3 Decoding Complexity Analysis

In Section 2.6, we have defined the decoding complexity of each coded bit in BMST as follows:

$$O_{\text{perbit}} = (4m - 2)boxplus + (3m + 3)\text{additions} + 2 \cdot O_{\text{basic code}}.$$  

It is clear that the complexity of first two parts is linear correlation with memory length $m$. Now we focus on the decoding complexity of basic
code. For a multiple-rate code, it can be defined as follows:

$$O_{basiccode} = \frac{\alpha}{\alpha + \beta}O_{low} + \frac{\alpha}{\alpha + \beta}O_{high}. \quad (4.3)$$

$O_{low}$ and $O_{high}$ denote the decoding complexity per coded bit in low rate code and high rate code, respectively. To compare the decoding complexity of RSPC codes and RMEH codes, we need to know the decoding complexity of the four component codes. In Chapter 2, we already computed their decoding complexity. Now, their decoding complexity per coded bit is shown in Table 4.3. We record 6 different types of operations (addition, subtraction, multiplication, division, logarithm and exponent). To simplify the comparison, we use a program to calculate the execution time of the 6 operations. In fact, addition and subtraction have the same execution time, so they are recorded together. Multiplication and division have the same execution time and they are recorded together. Moreover, the execution time of addition, subtraction, multiplication and division is far less than the execution time of logarithm and exponent. In other words, we can ignore the influences of additions and multiplications and only need to compare the numbers of logarithms and exponents. Furthermore, the decoding complexity of high rate codes is much higher than low rate codes.
Thus, we can focus on the decoding complexity of high rate codes. In the decoding of SPC code, there are $3 - 6/n$ logarithms and $6 - 12/n$ exponents. For EH code, there are only 1 logarithm and 3 exponents. So we believe that the decoding complexity of EH code is about half that of SPC code. With half of the memory length and half of the basic code’s decoding complexity, the decoding complexity of RMEH-BMST system should be only half of RSPC-BSMT system’s.

Now, we choose a certain code rate $12/16$ to make a comparison. Table 4.4 shows the decoding complexity of each coded bit in RSPC-BMST with $m = 4$ and RMEH-BMST with $m = 2$. For RSPC-BMST, the number of exponents is 36.625, which is about double of the number 17.6 in RMEH-BMST. The number of logarithms in RSPC-BMST is 18.3125, which is more than double of 7.8 logarithms in RMEH-BMST. This matches well with the analysis above.
Table 4.4: The decoding complexity of per coded bit in RSPC-BMST and RMEH-BMST when code rate is 12/16.

<table>
<thead>
<tr>
<th></th>
<th>Addition/Subtraction</th>
<th>Multiplication/Division</th>
<th>Logarithm</th>
<th>Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSPC(m=4)</td>
<td>52.5584</td>
<td>36.625</td>
<td>18.3125</td>
<td>36.625</td>
</tr>
<tr>
<td>RMEH(m=2)</td>
<td>79.4</td>
<td>27.2</td>
<td>7.8</td>
<td>17.6</td>
</tr>
</tbody>
</table>

Figure 4.1: Performance of RMEH-BMST codes composed of (32, 6) RM codes and (32, 26) EH codes.

4.4 Simulation Results

4.4.1 Performance of RMEH-BMST codes

Fig. 4.1 presents the performance of RMEH-BMST codes in Table 4.1, with $L = 1000$ and $B = 31$. The iterative sliding-window decoding algorithm is used and the maximum iteration number is 18. The BER performance curve of each rate is shown with its lower bound curve. It is clear
that the performances match well with their lower bound. The waterfall region performance of each code rate is close to Shannon limit within one dB away.

4.4.2 Comparison of RMEH-BMST Codes and RSPC-BMST Codes

Fig. 4.2 shows the BER performance of RSPC-BMST code and RMEH-BMST code at code rate 12/16. The performance of RMEH-BMST codes with memory length 2 is almost the same with RSPC-BMST codes with memory length 4. However, it becomes a 4-orders of magnitude improvement when memory length of RMEH increases to 4. If we set the same
Figure 4.3: Convergence rate of RMEH-BMST codes and RSPC-BMST codes (Rate = 12/16, SNR = 4.0 dB).

decoding delay for these two systems, it will take RSPC-BMST codes 940613 ms to come out one frame (with $L = 1000$) result and take RMEH-BMST codes 494639 ms to come out one frame result. This is close to the numerical analysis in Section 4.3. In fact, the decoding delay of RMEH-BMST is only half of RSPC-BMST, so RMEH-BMST codes can come out one frame result with only 269268 ms. Fig. 4.3 shows the convergence rate of RMEH-BMST codes and RSPC-BMST codes when iterative forward-backward decoding algorithm is used at code rate 12/16. The performance of RMEH-BMST codes converges within 70 times iterations and the performance of RSPC-BMST codes converges after 110 times iterations. This
can be another advantage of RMEH-BMST system.

4.5 Conclusion

In this Chapter, we design a multiple-rate RMEH codes by using first order RM codes and extended Hamming codes. Due to the few requirement on memory length, the decoding complexity of RMEH-BMST codes is only half of RSPC-BMST codes. This saves much time on BMST simulations. Numerical results show that the performances of different rate RMEH-BMST codes are very close to the channel capacity. When iterative forward-backward decoding algorithm is used, the convergence rates of RMEH-BMST codes are faster than RSPC-BMST codes.
Conclusion

This thesis aims to design multiple-rate codes with low error floors under AWGN channel. We learn some existing coding schemes and try to optimise them. Serial concatenation of Hamming codes and an accumulator usually exhibits poor error floor performance due to their small minimum distances. In Chapter 3, we propose to replace the outer Hamming codes by product codes constructed from Hamming codes and SPC codes. In this way, the minimum distance of the outer code can be doubled and the code rate can be changed if we use different SPC codes. At the same codeword length, the error floor performances of proposed codes have at least two orders of magnitude improvement. On the other hand, we use extended Hamming codes and first order RM codes to constitute multiple-rate RMEH codes and these codes are integrated into the BMST system. Compared with the existing RSPC-BMST codes, RMEH-BMST codes have lower decoding complexity, but the code rate range has minor loss.
References


