國立交通大學

電信工程研究所

碩士論文

應用於 BCH 與 LDPC 串接編碼系統的 BCH 碼選擇方法以及其疊代解碼演算法

BCH Code Selection and Iterative Decoding for BCH and LDPC Concatenated Coding System

研 究 生：陳品翰

指導教授：陳伯寧　教授

中華民國一Ｏ一年七月

應用於 BCH 與 LDPC 串接編碼系統的 BCH 碼選擇方法以及其疊代解碼演算法

BCH Code Selection and Iterative Decoding for BCH and LDPC Concatenated Coding System

研 究 生：陳品翰　　Student：Pin-Han Chen

指導教授：陳伯寧　　Advisor：Po-Ning Chen

國 立 交 通 大 學

電信工程研究所

碩 士 論 文

A Thesis

Submitted to Institute of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

July 2012

Hsinchu, Taiwan, Republic of China

中華民國一Ｏ一年七月

應用於BCH與LDPC串接編碼系統的BCH碼選擇方法以及其疊代解碼演算法

學生: 陳品翰　　指導教授: 陳伯寧

國立交通大學電信工程研究所碩士班

摘要

對於兩碼串連的串接系統而言，消除由內碼解碼器所殘餘的錯誤是外碼最主要的設計目的。為了有效地消除那些殘餘的錯誤，必須基於內碼解碼器的的行為來選擇適當的外碼。在這篇碩士論文中，我們考慮一個可適用於快閃記憶體的商業標準，也就是系統化(Systematic)的BCH碼與正規類循環低密度位元檢測碼(QC-LDPC)的串接系統。我們提出一個根據內部的低密度位元檢測碼解碼器產生的錯誤模式選擇BCH碼的方法，使得在輕微的碼率損失下，可有效降低LDPC碼的錯誤遲滯現象。除此之外，對於使用代數解碼演算法的外碼解碼器，我們提出了一個基於反饋的疊代解碼演算法以改善解碼錯誤率。模擬結果顯示我們所提出的反饋疊代解碼只有些微地增加解碼複雜度，卻可以有效地改善系統錯誤率。

BCH Code Selection and Iterative Decoding for BCH and LDPC Concatenated Coding System

Student: Pin-Han Chen Advisor: Po-Ning Chen

Institute of Communications Engineering

National Chiao Tung University

Abstract

For a serial concatenated coding system, the outer code is mainly designed to remove the residual errors induced by the inner decoder. In order to efficiently eliminate these residual errors, the selection of outer codes should depend on the error behavior of the inner decoder. In this thesis, we consider the concatenated systematic BCH and QC-LDPC coding system, which may be suitable for commercial standards of flash memories. We first propose a selection method of BCH codes according to the error patterns that are produced by the inner LDPC code decoder so as to lower the error floor but retain minor rate loss. Moreover, for the algebraic outer decoder, we present a feedback-based iterative decoding algorithm to improve the decoding performance. Simulations show that our feedback iterative decoding can effectively improve the system performance with only a slight increase in decoding complexity.

Acknowledgements

First of all, I would like to extend my sincere gratitude to my advisor, Professor Po-Ning Chen, for his patient guidance and support. Secondly, I would like to express my heartfelt gratitude to Professor Chung-Hsuan Wang: I deeply appreciate your invaluable instructions in my research. Besides, I would like to show my special gratitude to Dr. Jian-Jia Weng: I have benefited a lot from your enthusiastic teaching. This thesis would not have been possible without these helps. Finally, I would like to thank Jian-Dong Du and all the members in the NTL lab for their support and friendship as well as all those who have helped me in these two years.

Contents

List of Figures

Chapter 1

Introduction

With an increasing demand for large and efficient data storage, providing reliable data access on flash memory gradually becomes an important research issue. In the last decade, Bose-Chaudhuri-Hocquenghem (BCH) codes have been widely used for data protection on flash memory [1–3]. Algebraic decoding algorithms for BCH codes are then employed when errors are detected [4]. However, as the rapid development of flash memory, the required bit error rate (BER) is more strict than before. It then occurs that the BCH codes are not adequate to achieve the desired BER. Hence, many commercial companies attempt to adopt new coding technology such as the LDPC code to alleviate this problem in the next generation flash memory standard.

Serial concatenated coding is a technique to combine several error correcting codes into a single one. In general, such a concatenated code can increase the error correcting capability dramatically. For flash memory industries, a favorable coding scheme is perhaps to concatenate systematic BCH codes with quasi-cyclic low-density parity-check (QC-LDPC) codes. From our viewpoint, this coding scheme have several advantages apart from its powerful correcting capability.

First, owing to the systematic characteristic, the decoding complexity is often not large.

Secondly, the data read operation can be directly performed by the outer BCH code decoder if the quality of read channel is good. Only when the outer code decoding fails, the inner LDPC code decoder is activated to produce more reliable estimates of the original data. Thirdly, the conventional hardware implementation of BCH code encoder and decoder can be retained. Fourthly, the special parity-check matrix structure of the QC-LDPC code [5] makes fast encoding and low complexity decoding feasible [6, 7].

Based on the above concatenated coding scheme, the QC-LDPC code can be decoded by the bit flipping algorithm (BFA) or the sum-product algorithm (SPA) [8]. However, it is observed from simulations that a remarkably good BER performance is resulted at low-to-moderate signal-to-noise power ratio (SNR) but an error floor at high SNR due to some particular error patterns seems unavoidable. This error floor phenomenon is even more serious especially for QC-LDPC codes of short block length. Although the outer BCH code can improve this error floor, increasing the correcting capability of the outer code will be at the expense of code rate loss.

To seek a balance between the BER performance and the code rate, we propose a selection method for the outer BCH code when the inner QC-LDPC code is given. The idea behind our method is to examine the error patterns in the error floor region and choose an appropriate BCH code that can compensate most of the dominant error patterns.

In addition, since in our coding system, a codeword of the inner LDPC code corresponds to multiple codewords of the outer BCH code, the successful decoded results for the outer BCH codes can be properly utilized for performance enhancement. Hence, we also propose an iterative decoding strategy for this concatenated coding system. Simulations showed that the BER can be effective reduced with such an interaction between the inner and outer code decoders.

The rest of the thesis is organized as follows. In Chapter 2, some background of the concatenated coding system is presented. In Chapter 3, the selection method of BCH codes and the proposed feedback strategy for iterative decoding are given. In Chapter 4, simulation results are summarized and remaked. Chapter V concludes this work.

Chapter 2

Preliminaries

In this chapter, we provide the background knowledge on LDPC code decoders and BCH code decoders. Specifically, Section 2.1 introduces the system model we consider as well as the serial concatenated coding scheme. Section 2.2 talks about the decoding algorithms for LDPC codes. Section 2.3 gives the algorithm that will be employed for decoding the BCH codes. Section 2.4 presents the conception of dominated error patterns for LDPC codes in an error floor region; further discussion on this subject will be given in subsequent chapters.

2.1 Concatenation of LDPC and BCH coders

As shown in Figure 2.1, we consider a concatenated coding scheme that comprises β identical $(N_{\text{BCH}}, K_{\text{BCH}})$ systematic BCH codes as the outer code and an $(N_{\text{LDPC}}, K_{\text{LDPC}})$ systematic LDPC code as the inner code. In this scheme, the BCH codes are chosen to be capable of correcting t random errors, and the LDPC code adopted is a QC-LDPC code.

It can then be observed from Figure 2.1 that the total number of information bits is $\beta \cdot K_{\text{BCH}}$. These information bits are first divided into β groups of size K_{BCH}. Bits in each group are then encoded by a BCH code encoder. The resulting β BCH codewords plus q zeros are aggregated as the information bits for the LDPC encoder, where $q = K_{\text{LDPC}} - \beta \cdot N_{\text{BCH}} \geq$

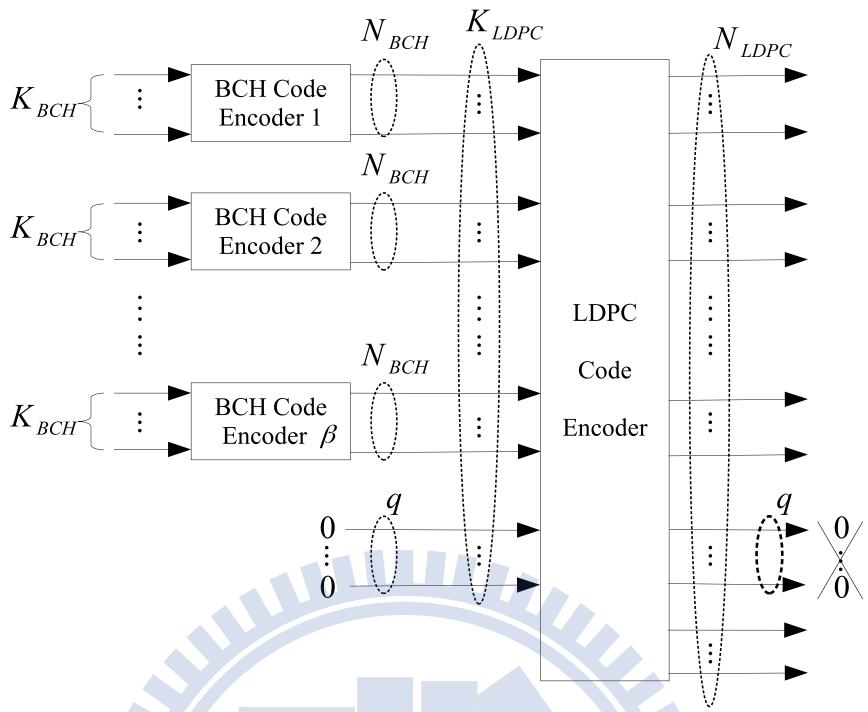

Figure 2.1: Block diagram of the concatenated coding system consisting of β $(N_{\mathrm{BCH}}, K_{\mathrm{BCH}})$ BCH codes and one $(N_{\mathrm{LDPC}}, K_{\mathrm{LDPC}})$ LDPC code. q zeros will be added at the end of these β BCH codewords, where $q = K_{\mathrm{LDPC}} - \beta \cdot N_{\mathrm{BCH}}$.

0. Afterwards, a length-N_{LDPC} LDPC codeword is generated. Note that since the LDPC code is systematic, the additional q zeros in the LDPC codeword does not need to be transmitted (as indicated by the big cross on the right of Figure 2.1). The purpose of these zero padding is to match the number of output bits of the BCH encoders to the number of required input bits of the LDPC encoder. For notational convenience, we let $\mathcal{P}$ be the set of indices corresponding to these q extra zeros in an LDPC codeword.

In our system model, we suppose that except for the q extra zeros, the coded bits are BPSK-modulated and transmitted over an additive white Gaussian noise (AWGN) channel. Hence, the received signal at time i can be represented as

$$y_i = x_i + w_i, \quad \forall\, i \notin \mathcal{P}$$

where $x_i \in \{-1, +1\}$ is the ith modulated symbol, and w_i denotes a zero-mean Gaussian noise sample with variance σ^2.

At the receiver side, the sum-product algorithm or bit-flipping algorithm is first employed to decode the LDPC code. The hard-decision decoding result is denoted by $\hat{\boldsymbol{z}}$. Upon reception of this hard-decision sequence $\hat{\boldsymbol{z}}$, an algebraic decoding algorithm then performs the decoding of BCH codes and outputs the estimates of the information bits. These decoded information bits are denoted by $\hat{\boldsymbol{d}}$.

2.2 Decoding Algorithm for LDPC Code

In this section, the details of bit-flipping algorithm (BFA) [8] and the sum-product algorithm (SPA) [8] are addressed.

The BFA is a hard-decision decoder for LDPC codes, while the SPA is a soft-decision decoder for LDPC codes. Both algorithms pass messages between nodes in the Tanner graph of the code. For the BFA, a variable node sends a message to its neighboring check nodes to declare itself as a 1 or 0, and a check node sends a message to its neighboring variable nodes, declaring whether the parity-check is satisfied or not. The working theory behind the SPA is similar to the BFA except that what the SPA passes between variable nodes and check nodes is the log-likelihood ratio between 0 and 1. In this thesis, the conventional SPA proposed by Gallager in his 1963 thesis is employed [8].

Before we proceed to introduce the details of the two algorithms, the notations used later must be defined first. Denote by $\boldsymbol{H}$ the parity-check matrix of a $(N_{\text{LDPC}}, K_{\text{LDPC}})$ QC-LDPC code. Then, the number of variable nodes in the Tanner graph is N_{LDPC}. Let the Tanner graph corresponding to $\boldsymbol{H}$ containing M check nodes. Put v_i and $\mathcal{M}(i)$ as the ith variable node and the set of check nodes connecting to v_i in the Tanner graph, respectively. Similarly, let c_j and $\mathcal{N}(j)$ be respectively the jth check node and the set of variable nodes connecting to c_j. With these notations, the decoders of the LDPC code can be described in the following steps.

2.2.1 The Bit-Flipping Algorithm

The Bit-Flipping Algorithm:

Step 0. Initialization: The value of the ith variable node, $\hat{z}_i$, is obtained from the ith received value y_i via hard-decision:

$$\hat{z}_i = \begin{cases} 1, & \text{if } y_i \le 0 \text{ and } i \notin \mathcal{P} \\ 0, & \text{otherwise} \end{cases}, \ \forall\, 1 \le i \le N_{LDPC}. \tag{2.1}$$

Step 1. Bit to Check Message Update: Each variable node sends its value to its neighboring check nodes. Afterwards, each check node calculates whether the parity-check is satisfied or not in terms of the messages obtained from the variable nodes. Set $Q_j = 1$ if the j-th parity-check equation is satisfied; otherwise, set $Q_j = -1$. In other words,

$$Q_j = \prod_{i \in \mathcal{N}(j)} (1 - 2\hat{z}_i)\,,\ \forall\, 1 \le j \le M \tag{2.2}$$

If all $Q_j = 1$, then output $\hat{\boldsymbol{z}}$ and stop the algorithm.

Step 2. Check to Bit Message Update: Each check node sends its Q-value to its neighboring variable nodes. Set $F_i = 1$ if all check equations associated to check nodes that are connected to the ith variable node are "not satisfied;" otherwise, set $F_i = 0$. Flip *the value of the variable node if* $F_i = 1$.[1] *Specifically,*

$$F_i = \prod_{j \in \mathcal{M}(i)} \frac{1 - Q_j}{2}, \ \forall\ 1 \leq i \leq N_{LDPC} \text{ and } i \notin \mathcal{P}. \tag{2.3}$$

$$\hat{z}_i = \begin{cases} (\hat{z}_i + 1) \bmod 2, & \text{if } F_i = 1 \\ \hat{z}_i, & \text{if } F_i = 0 \end{cases}, \ \forall\ 1 \leq i \leq N_{LDPC} \text{ and } i \notin \mathcal{P}. \tag{2.4}$$

If $\hat{\boldsymbol{z}}$ *is a valid LDPC codeword or the maximum number of iterations is reached, output* $\hat{\boldsymbol{z}}$ *and stop the algorithm; else go to Step 1.*

2.2.2 The Sum-Product Algorithm

The Sum-Product Algorithm:

Step 0. For $1 \leq i \leq N_{LDPC}$*, if* $i \notin \mathcal{P}$*, then compute the LLR* R_i *of bit* i*; else let* $R_i = \infty$*. I.e.,*

$$R_i \triangleq \begin{cases} 2y_i/\sigma^2, & \text{if } i \notin \mathcal{P} \\ \infty, & \text{if } i \in \mathcal{P} \end{cases} \tag{2.5}$$

For $1 \leq i \leq N_{LDPC}$ *and* $1 \leq j \leq M$*, initial* $E_{i,j} = 0$*, where* $E_{i,j}$ *represents the extrinsic information passing from check node* c_j *to variable node* v_i*.*

Step 1. Bit to Check Message Update: Denote by $L_{i,j}$ *the extrinsic information passing from variable node* v_i *to check node* c_j*. Then, assign for* $1 \leq i \leq N_{LDPC}$*, and for* $j \in \mathcal{M}(i)$ *for a given* i*,*

$$L_{i,j} = \begin{cases} \sum_{j' \in \mathcal{M}(i), j' \neq j} E_{i,j'} + R_i, & \text{if } i \notin \mathcal{P} \\ R_i, & \text{if } i \in \mathcal{P} \end{cases} \tag{2.6}$$

Step 2. Check to Bit Message Update: Renew for $1 \leq j \leq M$*, and for* $i \in \mathcal{N}(j)$ *for a given* j*,*

$$E_{i,j} = \ln \left(\frac{1 + \prod_{i' \in \mathcal{N}(j), i' \neq i} \tanh\left(\frac{L_{i',j}}{2}\right)}{1 - \prod_{i' \in \mathcal{N}(j), i' \neq i} \tanh\left(\frac{L_{i',j}}{2}\right)} \right). \tag{2.7}$$

[1] Note that we do not need to estimate those $\hat{z}_i$ for $i \in \mathcal{P}$ because they are deterministically zero.

Step 3. Codeword Test: Compute the reliability D_i *of the* i*th bit as*

$$D_i = \begin{cases} \sum_{j \in \mathcal{M}(i)} E_{i,j} + R_i & \text{if } i \notin \mathcal{P} \\ R_i & \text{if } i \in \mathcal{P} \end{cases}, \tag{2.8}$$

For $1 \leq i \leq N_{LDPC}$, *obtain the hard-decision result:*

$$\hat{z}_i = \begin{cases} 1 & \text{if } D_i \leq 0 \\ 0 & \text{if } D_i > 0 \end{cases} \tag{2.9}$$

If $\boldsymbol{H}\hat{\boldsymbol{z}}^T = 0$, *output* $\hat{\boldsymbol{z}}$ *as the decoded result and stop the algorithm; else if the maximum number of iterations is reached, stop the algorithm; else go to Step 1.*

2.3 Algebraic Decoding Algorithm for BCH Codes

In this section, we describe the algebraic decoding algorithm for BCH codes.

Upon the completion of the LDPC code decoding, the decoded sequence $\hat{\boldsymbol{z}}$ is partitioned into β sub-sequences, and each sequence is delivered to its corresponding BCH code decoder for decoding. In this thesis, the BCH code decoder chosen is the Euclidean algorithm [4]. Since the Euclidean algorithm is a bounded distance decoder, a sub-sequence is guaranteed to be decoded to a valid codeword when this sub-sequence lies in a codeword sphere of radius t. As such, if a sub-sequence can be decoded to a valid codeword, the estimates of the information bits, i.e., $\hat{\boldsymbol{d}}$, are taken from the outputs of this BCH code decoder; otherwise, the information bits in $\hat{\boldsymbol{z}}$ are used instead. Note again that the BCH codes we adopt are systematic.

Now consider a primitive BCH code with codeword length $N_{\text{BCH}} = 2^m - 1$ and generator polynomial $g(x)$. Assume its code polynomial is

$$\boldsymbol{v}(X) = v_0 + v_1 X + \cdots + v_{N_{\text{BCH}}-1} X^{N_{\text{BCH}}-1}. \tag{2.10}$$

Let the received sequence (passed from the LDPC decoder) be

$$\boldsymbol{r}(X) = r_0 + r_1 X + \cdots + r_{N_{\text{BCH}}-1} X^{N_{\text{BCH}}-1}. \tag{2.11}$$

As a convention, we can write

$$\boldsymbol{r}(X) = \boldsymbol{v}(X) + \boldsymbol{e}(X),$$

where $\boldsymbol{e}(X)$ is the error pattern polynomial.

The first step in decoding a BCH code is to check whether $\boldsymbol{r}(X)$ is a valid code polynomial or not. If affirmative, then the information bits encoded to $\boldsymbol{r}(X)$ is surely the desired decoding output. If the answer is negative, then compute the syndrome $\boldsymbol{s}$ corresponding to the received polynomial $\boldsymbol{r}(X)$.

For a t-error correctable BCH code, the syndrome consists of $2t$ components in GF(2^m):

$$\boldsymbol{s} = (s_1, s_2, \cdots, s_{2t}), \tag{2.12}$$

where the i-th component of the syndrome is

$$s_i = \boldsymbol{r}(\alpha^i) \quad \text{for } 1 \leq i \leq 2t, \tag{2.13}$$

and α is the primitive element.

Let $\phi(X)$ be the minimum polynomial of α^i. Dividing $\boldsymbol{r}(X)$ by $\phi(X)$, we obtain the following equation:

$$\boldsymbol{r}(X) = \boldsymbol{a}(X)\phi(X) + \boldsymbol{b}(X) \tag{2.14}$$

where $\boldsymbol{b}(X)$ is the remainder with degree less than $\phi(X)$. Based on the above equation, it is obvious that (2.13) can be rewritten as $s_i = \boldsymbol{b}(\alpha^i)$ because $\phi(\alpha^i) = 0$.

Since $\boldsymbol{r}(X) = \boldsymbol{v}(X) + \boldsymbol{e}(X)$, we can relate the syndrome and the error pattern through:

$$\boldsymbol{s}_i = \boldsymbol{r}(\alpha^i) = \boldsymbol{v}(\alpha^i) + \boldsymbol{e}(\alpha^i) = \boldsymbol{e}(\alpha^i) \quad \text{for } 1 \leq i \leq 2t \tag{2.15}$$

Suppose $\boldsymbol{e}(X)$ has ν errors at locations $X^{j_1},\ X^{j_2},\ \cdots,\ X^{j_\nu}$, i.e.,

$$\boldsymbol{e}(X) = X^{j_1} + X^{j_2} + \cdots + X^{j_\nu}, \quad \text{where } 0 \leq j_1 < j_2 < \cdots < j_\nu \leq N_{\mathrm{BCH}} - 1. \tag{2.16}$$

Combining (2.15) and (2.16), we have the following equations:

$$\begin{aligned} s_1 &= \alpha^{j_1} + \alpha^{j_2} + \cdots + \alpha^{j_\nu} \\ s_2 &= (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \cdots + (\alpha^{j_\nu})^2 \\ s_3 &= (\alpha^{j_1})^3 + (\alpha^{j_2})^3 + \cdots + (\alpha^{j_\nu})^3 \\ &\vdots \\ s_{2t} &= (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \cdots + (\alpha^{j_\nu})^{2t} \end{aligned} \tag{2.17}$$

where $\alpha^{j_1},\ \alpha^{j_2},\ \cdots,\ \alpha^{j_\nu}$ are unknown. So for BCH codes, the decoding process is simply to solve these equations.

As aforementioned, we use the Euclidean algorithm to decoding BCH codes in this thesis. Apparently, the exponents $j_1,\ j_2,\ \cdots,\ j_\nu$ tell us the error locations in $\boldsymbol{e}(X)$. Define the polynomial:

$$\boldsymbol{\sigma}(X) \triangleq (1+\alpha^{j_1}X)(1+\alpha^{j_2}X)\cdots(1+\alpha^{j_\nu}X) \tag{2.18}$$

$$= \sigma_0 + \sigma_1 X + \sigma_2 X^2 + \cdots + \sigma_\nu X^\nu \tag{2.19}$$

The roots of $\boldsymbol{\sigma}(X)$ are clearly $(\alpha^{j_1})^{-1},\ (\alpha^{j_2})^{-1},\ \cdots,\ (\alpha^{j_1})^{-1}$. As these roots tell the inverses of the error-location numbers, $\boldsymbol{\sigma}(X)$ is called the *error-locator polynomial*. Solving the error-locator polynomial can also give the error locations in $\boldsymbol{e}(X)$. Note that we will use the information of the error-locator polynomial in the next chapter.

Finally, we roughly summarize the decoding procedure of the BCH codes as follows. First, we compute the syndrome $\boldsymbol{s}$ from the received polynomial $\boldsymbol{r}(X)$, followed by the determination of the error locator polynomial $\boldsymbol{\sigma}(X)$. Then, we determine the error location numbers $\alpha^{j_1},\ \alpha^{j_2},\ \cdots,\ \alpha^{j_\nu}$ by finding the roots of $\boldsymbol{\sigma}(X)$, and hence we can use this information to correct the errors in $\boldsymbol{r}(X)$.

2.4 Dominated Failure Patterns: Trapping Sets

As widely reported in many research documents, decoding LDPC codes by the SPA can result in relatively large BERs at low to moderate SNR in AWGN channels. These relatively large BERs usually appear in the form of error floor at high SNR, i.e., the BERs descend much slowly or even stop decreasing at high SNR, and this error floor phenomenon is more evident especially for short block length LDPC codes.

From a series of studies, a major failure pattern that causes the error floor phenomenon is the "trapping set [9]." We say that a sub-graph of a Tanner graph is an (a, b) trapping set if it contains a variable nodes and b odd degree neighboring check nodes that are connected to these variable nodes. When these a variable nodes are erroneous while the remaining bits are correct, it can be observed that only these b check nodes can possibly send correct messages to these a variable nodes. Hence, the the trapping sets, in particular with smaller values of a and b, are the dominant error patterns in the error floor region.

Chapter 3

BCH Code Selection and Iterative Decoding for The Concatenated Coding System

In this chapter, a selection method for the outer BCH codes will be proposed.

Specifically, for an LDPC code, it has been known that the error floor is majorly contributed by the dominant trapping sets. Our initial idea is therefore that by concatenating a proper outer code, this problem can be alleviated. With this in mind, for the concatenated coding scheme in this thesis, the outer BCH codes are chosen to eliminate those trapping sets and hence improve the BER in the error floor region.

Nevertheless, increasing the error capability of the outer code is usually done at the expense of the code rate. This implies that there is a trade-off between the code rate and the BER performance in such a choice of BCH codes. It then leads to the objective in our design: to improve the BER in the error floor region without much rate loss.

Other than introducing a selection method for the outer BCH codes in Section 3.1, a decision-feedback-aided decoding is presented. In general, iterative decoding between inner and outer codes in a concatenated coding scheme must use soft-decision decoding for both the outer code and the inner code. However, in our system, the outer BCH code is hard-decision decoded. So in Section 3.2, we propose an iterative decoding algorithm for our concatenated system such that the outer BCH code does not need to be soft-decision decoded. Via this way, the inner decoder and the outer decoder can cooperate with each other to enhance the overall BER performance.

Finally, Section 3.3 introduces more iterative decoding ideas that can be used in our concatenated coding system.

3.1 The Selection of Outer BCH Codes

Since we wish to choose outer BCH codes to eliminate the ill effect of those trapping sets, a search of the dominated trapping sets must be first performed for a given inner QC-LDPC code. The collected (a, b) trapping sets are then classified according to the values of parameters a and b.

The scenario we are concerned with is as follows. Under the premise that the codeword length of the outer BCH code has been determined, and a LDPC code codeword comprises β BCH code codewords, if an (a, b) trapping set with $a = t$ occurs, those t errors might simultaneously fall into a single BCH code block in the worst case. To ensure that such an error pattern can be removed by the BCH code decoder, a t-error correctable BCH code must be chosen. In fact, by this selection, not only the trapping sets with $a \leq t$ are guaranteed to be cleaned up by the outer BCH code but also part of the trapping sets with $a > t$ can be possible corrected as long as $\beta > 1$. Note that if the code rate of the resulting concatenated system is higher than the system requirement, we may further increase the correcting capability of the BCH code for further improvement at the price of decreasing the code rate, i.e., e.g., choosing a $(t + 1)$-error correctable BCH code.

As a contrast, if the code rate of the resulting system is lower than the desire, we could increase the codeword length of BCH codes to improve the code rate without much sacrificing the error correcting capability. As such, since the value of β is decreased, the "extra" error correcting capability for the trapping sets with $a > t$ may be degraded.

3.2 Decision Feedback-Aided Iterative Decoding

After selecting proper BCH coders according to the proposed rule in the previous section, simulations are performed and show that our choice does improve the BER in the error floor region. However, a little performance loss can also be observed in the waterfall region due to the rate loss.

To compensate this BER performance degradation in the waterfall region, and also inspired by the success of iterative decoding, we design an novel iterative decoding strategy for our concatenated coding system: namely, to feedback information from the β algebraic BCH code decoders to the LDPC code decoder.

It is worth mentioning that one can of course employ a soft-decision decoding algorithm for BCH codes and result in an iterative decoding naturally; however, such approach will evidently lose the advantage of the fully developed hardware implementation of algebraic decoders. In order to maintain this hardware superiority of hard-decision BCH decoders, we propose an alternative yet simple solution.

Our proposal relies on a premise that owing to the superior error correcting capability of the inner LDPC code, the noisy received signals are mostly corrected back to its original transmission values after the LDPC code decoding. In addition, by our selection of BCH codes, it is reasonable to expect that most of the residual errors can be removed by the outer BCH code decoders. This implies that the decoding results of BCH codes are very trustworthy. Therefore, our main idea is to feedback these decoding results to the LDPC code decoder and re-do the LDPC decoding. This may make the LDPC code decoder re-generating even more reliable outputs, and hence form a positive interaction cycle between the inner and outer decoders.

Some notations that will be used later are introduced first. Let $\mathcal{S}$ denote the set of indices in an LDPC codeword of length N_{LDPC} that correspond to those BCH decoder inputs, out of which valid BCH codewords can be obtained by the BCH decoder. Hence, if β' out of β BCH decoders output valid BCH codewords (i.e., decode successfully), then the size of $\mathcal{S}$ is exactly $\beta' \cdot K_{\text{BCH}}$. Notably, the direct correspondences between the $\beta \cdot K_{\text{BCH}}$ input bits of the BCH encoders and the LDPC codeword are well defined since both the BCH codes and the LDPC code are systematic. Denote by $\hat{d}_i$ the estimate of the ith bit in an LDPC codeword obtained from the previous outer [1] iteration, where $1 \leq i \leq N_{\text{LDPC}}$. With these notations, we are ready to describe the decision feedback-aided iterative decoding for our proposed concatenated coding system.

3.2.1 Decision Feedback-Aided Iterative Decoding between Algebraic and Bit-Flipping Algorithms

In the first outer iteration of the decision feedback-aided iterative decoding, the LDPC code is decoded by the original bit-flipping algorithm as have been described in Subsection 2.2.1. Starting from the second outer iteration, the decoding procedure of the LDPC code is changed to the following:

Step 0. Initialization: With the availability of $\mathcal{S}$ and $\{\hat{d}_i\}$ obtained from the previous outer iteration, the value of the ith variable node, $\hat{z}_i$, is obtained from the ith received

[1]There are two iterations in our decision feedback-aided iterative decoding algorithm. They are the internal iterations of the LDPC decoder and the external iterations between the BCH decoders and the LDPC decoder. For convenience, the former will be referred to as *inner* iterations, while the latter will be named *outer* iterations.

value y_i via hard-decision as:

$$\hat{z}_i = \begin{cases} 1 & \text{if } y_i \leq 0 \text{ and } i \notin \mathcal{P} \cup \mathcal{S} \\ \hat{d}_i & \text{if } i \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}, \ \forall\, 1 \leq i \leq N_{LDPC}. \tag{3.1}$$

Step 1. Bit to Check Message Update: Each variable node sends its value to its neighboring check nodes. Afterwards, each check node calculates whether the parity-check is satisfied or not in terms of the messages obtained from the variable nodes. Set $Q_j = 1$ if the j-th parity-check equation is satisfied; otherwise, set $Q_j = -1$. In other words,

$$Q_j = \prod_{i \in \mathcal{N}(j)} (1 - 2\hat{z}_i), \ \forall\, 1 \leq j \leq M \tag{3.2}$$

If all $Q_j = 1$, then output $\hat{\boldsymbol{z}}$ and stop the algorithm.

Step 2. Check to Bit Message Update: Each check node sends its Q-value to its neighboring variable nodes. Set $F_i = 1$ if all check equations associated to check nodes that are connected to the ith variable node are "not satisfied;" otherwise, set $F_i = 0$. Flip the value of the variable node if $F_i = 1$. Specifically,

$$F_i = \prod_{j \in \mathcal{M}(i)} \frac{1 - Q_j}{2}, \ \forall\, 1 \leq i \leq N_{LDPC} \text{ and } i \notin \mathcal{P} \cup \mathcal{S}. \tag{3.3}$$

$$\hat{z}_i = \begin{cases} (\hat{z}_i + 1) \bmod 2, & \text{if } F_i = 1 \\ \hat{z}_i, & \text{if } F_i = 0 \end{cases}, \ \forall\, 1 \leq i \leq N_{LDPC} \text{ and } i \notin \mathcal{P} \cup \mathcal{S}. \tag{3.4}$$

If $\hat{\boldsymbol{z}}$ is a valid LDPC codeword or the maximum number of (inner) iterations is reached, output $\hat{\boldsymbol{z}}$ and stop the algorithm; else go to Step 1.

3.2.2 Decision Feedback-Aided Iterative Decoding between Algebraic and Sum-Product Algorithms

In the first outer iteration of the decision feedback-aided iterative decoding, the LDPC code is decoded by the original sum-product algorithm as have been described in Subsection

2.2.2. Starting from the second outer iteration, the decoding procedure of the LDPC code is changed to the following:

Step 0. *For* $1 \leq i \leq N_{LDPC}$*, if* $i \notin \mathcal{P} \cup \mathcal{S}$*, then compute the LLR* R_i *of bit* i*; else if* $i \in \mathcal{P}$*, let* $R_i = \infty$*; else assign* $R_i = \frac{2\left(1-2\hat{d}_i\right)}{\sigma^2}$*. I.e.,*

$$R_i \triangleq \begin{cases} \frac{2\left(1-2\hat{d}_i\right)}{\sigma^2}, & \text{if } i \in \mathcal{S} \\ \infty, & \text{if } i \in \mathcal{P} \\ 2y_i/\sigma^2, & \text{otherwise} \end{cases} \tag{3.5}$$

For $1 \leq i \leq N_{LDPC}$ *and* $1 \leq j \leq M$*, initial* $E_{i,j} = 0$*, where* $E_{i,j}$ *represents the extrinsic information passing from check node* c_j *to variable node* v_i*.*

Step 1. *Bit to Check Message Update: Denote by* $L_{i,j}$ *the extrinsic information passing from variable node* v_i *to check node* c_j*. Then, assign for* $1 \leq i \leq N_{LDPC}$*, and for* $j \in \mathcal{M}(i)$ *for a given* i*,*

$$L_{i,j} = \begin{cases} R_i, & \text{if } i \in \mathcal{P} \cup \mathcal{S} \\ \sum_{j' \in \mathcal{M}(i), j' \neq j} E_{i,j'} + R_i, & \text{otherwise} \end{cases} \tag{3.6}$$

Step 2. *Check to Bit Message Update: Renew for* $1 \leq j \leq M$*, and for* $i \in \mathcal{N}(j)$ *for a given* j*,*

$$E_{i,j} = \ln \left(\frac{1 + \prod_{i' \in \mathcal{N}(j), i' \neq i} \tanh\left(\frac{L_{i',j}}{2}\right)}{1 - \prod_{i' \in \mathcal{N}(j), i' \neq i} \tanh\left(\frac{L_{i',j}}{2}\right)} \right). \tag{3.7}$$

Step 3. *Codeword Test: Compute the reliability* D_i *of the* i*th bit as*

$$D_i = \begin{cases} R_i & \text{if } i \in \mathcal{P} \cup \mathcal{S} \\ \sum_{j \in \mathcal{M}(i)} E_{i,j} + R_i & \text{otherwise} \end{cases}, \tag{3.8}$$

For $1 \leq i \leq N_{LDPC}$*, obtain the hard-decision result:*

$$\hat{z}_i = \begin{cases} 1 & \text{if } D_i \leq 0 \\ 0 & \text{if } D_i > 0 \end{cases} \tag{3.9}$$

If $\boldsymbol{H}\hat{\boldsymbol{z}}^T = 0$*, output* $\hat{\boldsymbol{z}}$ *as the decoded result and stop the algorithm; else if the maximum number of (inner) iterations is reached, stop the algorithm; else go to Step 1.*

3.3 Additional Modifications on Decision Feedback-Aided Iterative Decoding

From simulations, we sense that the outer BCH code decode may feedback incorrect information to the LDPC code decoder when the error correcting capability of the BCH code is too small. This occurs especially when the BCH code decoder outputs a valid but wrong codeword. In principle, this probability is roughly inversely proportional to the error correcting capability of the BCH code. In order to ensure the exactness of the feedback information from the BCH code decoder to the LDPC code decoder, we have experimented three strategies that will respectively introduce in the following subsections. In Subsection 3.3.1, an extra single parity-check (SPC) code will be added to the BCH code so as to ensure the exactness of the feedback information to the LDPC code decoder. In Subsection 3.3.2, the condition of the feedback strategy will become stricter. In Subsection 3.3.3, a modification on the outer BCH code will be tested. To ease the referring of the above three modifications, we will respectively call the modifications in Subsections 3.3.1, 3.3.2 and 3.3.3 as *Strategy* 1, *Strategy* 2 and *Strategy* 3.

3.3.1 Strategy 1: Concatenation of SPC, BCH and LDPC Codes

There exists an implicit condition for executing the feedback scheme, i.e., at least one BCH code decoder decodes successfully. Hence, the error correcting capability of the outer BCH code must not be too small; otherwise, the decoded result may still be wrong and another valid BCH codeword is outputted even if the BCH code decoder decodes successfully.

To further secure the exactness of the feedback information, we add an extra single parity-check (SPC) code to double-check the validity of the output BCH codewords. This will only result in a small rate loss, and the additional system complexity is almost minimized.

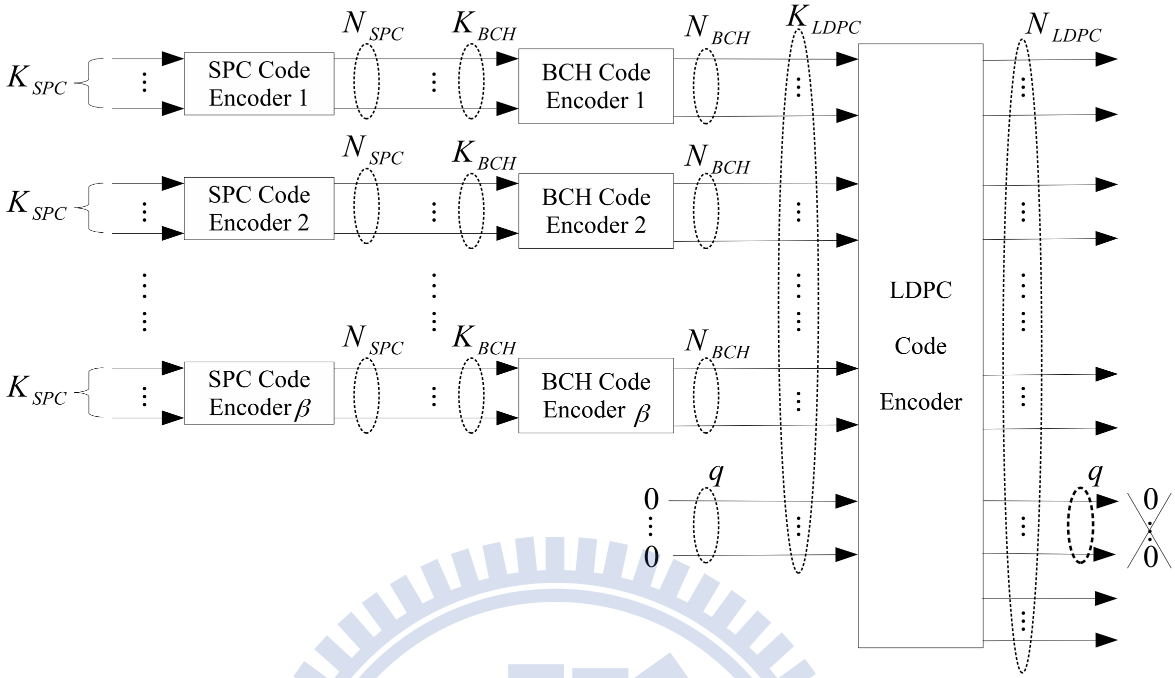

Figure 3.1: Block diagram of concatenation of the SPC, BCH and LDPC codes.

Consider the new system that comprises β identical $(N_{\text{SPC}}, K_{\text{SPC}})$ systematic SPC codes, β identical $(N_{\text{BCH}}, K_{\text{BCH}})$ systematic BCH codes and an $(N_{\text{LDPC}}, K_{\text{LDPC}})$ systematic LDPC code as shown in Figure 3.1. In this scheme, the number of information bits becomes βK_{SPC}, and $N_{\text{SPC}} = K_{\text{SPC}} + 1$. Specifically, the βK_{SPC} bits are first divided to β disjoint groups of size K_{SPC}. Each group is then independently encoded by its corresponding SPC code encoder. This results in β SPC codewords.

The SPC codewords are then fed into the following BCH code encoders, which generate β BCH codewords. Afterwards, the output β BCH codewords are aggregated as the information bits to the LDPC code encoder, which in turns generate an LDPC codeword of length N_{LDPC}. Similar to what have been stated in Section 2.1, q zeros will be padded to β BCH codewords in order to balance the total number of bits in β BCH codewords and the required number of bits for the LDPC code encoder.

As aforementioned, the SPC code is used to detect the errors that the BCH code decoders make. It can cooperate with either the BFA or the SPA when being applied to the decision feedback-aided iterative decoding. The procedures are basically the same as what we have described in the previous section. In the first outer iteration, the LDPC code decoder produces decoding outputs $\hat{\boldsymbol{z}}$ of length N_{LDPC}. Based upon this hard-decision sequence $\hat{\boldsymbol{z}}$, the BCH decoders then output the estimates $\hat{\boldsymbol{d}}$ of the information bits of BCH codes by using an algebraic decoding algorithm. Additionally, the results $\hat{\boldsymbol{d}}$ are double-checked to see whether they form a valid codeword of the SPC code. As a result, the set $\mathcal{S}$ is re-defined as the set of indices in an LDPC codeword of length N_{LDPC} that correspond to those SPC decoder inputs, out of which valid SPC codewords and also valid BCH codewords can be obtained. By the re-defined $\mathcal{S}$, the second outer iteration can be performed, following the steps described in Section 3.2.

3.3.2 Strategy 2: Explicit Feedback Condition on the Euclidean Distance

In this subsection, instead of adding an SPC code, we add an explicit condition under which the information bits in $\mathcal{S}$ can be feedbacked. Hence, the system structure in Figure 2.1 remains and no rate loss is resulted.

The decoding procedure is modified as follows. In the first outer iteration, the LDPC code decoder generates hard-decision sequence $\hat{\boldsymbol{z}}$. Based on this hard-decision sequence $\hat{\boldsymbol{z}}$, the BCH code decoders then output the estimates $\hat{\boldsymbol{d}}$ of the information bits of BCH codes by using an algebraic decoding algorithm. We then generate β BCH codewords with respect to the estimated information sequence $\hat{\boldsymbol{d}}$. Afterwards, for the ith BCH codeword just generated, where $1 \leq i \leq \beta$, we compute the Euclidean distance $u_{\text{BCH}}^{(i)}$ between this BCH codeword and the elements in its corresponding positions in channel output $\boldsymbol{y}$. We also compute the Euclidean distance $u_{\mathrm{LDPC}}^{(i)}$ between $\hat{\boldsymbol{z}}$ and $\boldsymbol{y}$ by taking into consideration only the elements corresponding to the the respective positions about the ith BCH codeword. Again, the correspondences between BCH codewords and $\boldsymbol{y}$ are well defined since the LDPC code is systematic. From our simulations, we notice that the portion of the decoded results $\hat{\boldsymbol{d}}$ that generates the ith BCH codeword is correct with higher probability if $u_{\mathrm{BCH}}^{(i)}$ is smaller than $u_{\mathrm{LDPC}}^{(i)}$. Therefore, we redefine the set $\mathcal{S}$ as follows: $\mathcal{S}$ is the set of indices in an LDPC codeword of length N_{LDPC} that correspond to those BCH decoder inputs, out of which valid BCH codewords can be obtained by the BCH decoder and also its respective $u_{\mathrm{BCH}}^{(i)}$ is smaller than $u_{\mathrm{LDPC}}^{(i)}$. By the re-defined $\mathcal{S}$, the second outer iteration can be performed, following the steps described in Section 3.2.

3.3.3 Strategy 3: Shortening the Outer BCH Code

It is obvious that if some of the information bits transmitted are known to the receiver, then the outer iterations can evidently improve the error performances by broadcasting these correct values. Hence, in this subsection, we propose to fix λ information bits as zeros for each BCH coder so that they can serve as the known information bits at the receiver to help improving the error performance.

Consider a concatenated coding scheme that comprises β identical $(N_{\mathrm{BCH}}, K_{\mathrm{BCH}})$ systematic BCH coders and an $(N_{\mathrm{LDPC}}, K_{\mathrm{LDPC}})$ systematic LDPC coder as shown in Figure 3.2. Among the K_{BCH} information bits feeding into each BCH encoder, λ of them are fixed as zeros. Hence, the total number of effective information bits for the entire system is reduced to $\beta K'_{\mathrm{BCH}}$, where $K'_{\mathrm{BCH}} = K_{\mathrm{BCH}} - \lambda$. Similar to what have been stated previously, these $\beta K'_{\mathrm{BCH}}$ bits are divided into β groups of size K'_{BCH}. After appending λ bits of zeros to each group, they are then fed into $(N_{\mathrm{BCH}}, K_{\mathrm{BCH}})$ BCH code encoders. This results in β BCH codewords. Afterwards, the β BCH codewords are aggregated as the information bits to the

LDPC code encoder. If necessary, q extra zeros will be appended to these $\beta \cdot N_{\text{BCH}}$ aggregated bits, where $K_{\text{LDPC}} = \beta N_{\text{BCH}} + q$. The LDPC code encoder then generates a length-N_{LDPC} LDPC codeword.

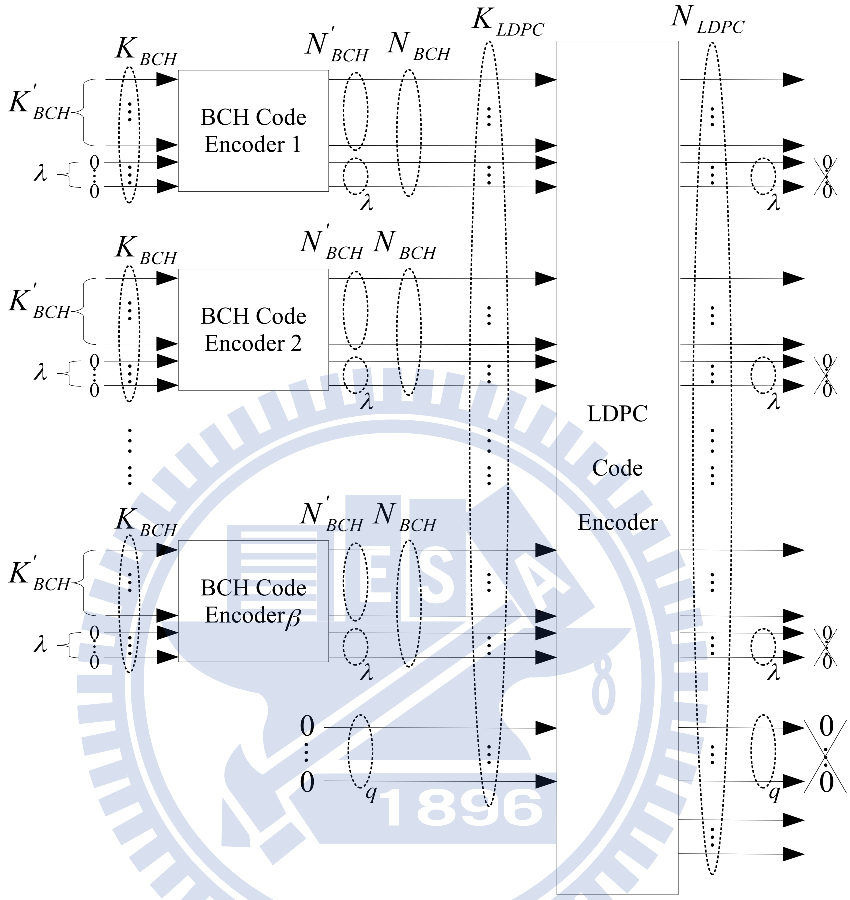

Figure 3.2: Block diagram of concatenated coding system with λ information bits being fixed as zeros.

Note that the $q + \beta\lambda$ extra zeros are not transmitted since the values of these bits are prior known at the receiver. Therefore, this strategy will only cause a small rate loss and the new code rate is given by:

$$\frac{\beta K'_{\text{BCH}}}{N_{\text{LDPC}} - \beta\lambda - q} = \frac{\beta K_{\text{BCH}} - \beta\lambda}{N_{\text{LDPC}} - \beta\lambda - q}.$$

The set $\mathcal{P}$ is redefined as those indices corresponding to these $\beta\lambda + q$ zeros. Moreover, these

λ zeros can be used to double-check whether the decoding outputs of the BCH decoder are valid BCH codewords or not. Notably, with this setting, a valid BCH codeword must be one, of which the component bits equal zeros in these λ specific positions. The detail of the decision feedback-aided iterative decoding procedure is then the same as that described in Section 3.2.

In comparison with Section 3.1, the strategy just proposed in this subsection increases the correcting capability of the outer BCH codes at a price of a little rate loss. Hence, in our simulations, a little performance loss can be observed at the waterfall region. More simulation results as well as discussions regarding them will be introduced in the next chapter.

Chapter 4

Simulation Results

In this chapter, simulation results are provided to demonstrate the effectiveness of our selection method for BCH codes as well as the proposed iterative decoding algorithm. Specifically, simulation results are summarized in Section 4.1, and discussions on them are given in Section 4.2.

4.1 System Settings

For all cases, the number of iterations for the inner LDPC decoder is 100. The maximum number of (outer) iterations between the inner and outer decoders is denoted by ξ. The condition to execute the outer iteration is that at least one of the BCH code decoders decodes successfully.

In our simulations, the $(6350, 5878)$ regular QC-LDPC code $\mathcal{C}_1$ and $(4590, 3835)$ regular QC-LDPC code $\mathcal{C}_2$ [10] are adopted. For $\mathcal{C}_1$, the minimum value of a among its dominant trapping sets is 4; hence, we choose the $(255, 223, 4)$ and $(511, 475, 4)$ 4-errors correctable BCH codes as its outer codes. By performing similar trapping set search onto $\mathcal{C}_2$, the minimum value of a is again 4; hence, the $(255, 223, 4)$ and $(511, 475, 4)$ BCH code can also be used as the outer codes for $\mathcal{C}_2$.

We list the maximum number of outer iterations for each code combination as follows.

- The maximum number of outer iterations for the $(255, 223, 4)$ BCH code and $\mathcal{C}_1$ is $5878/255 \approx 23$. As can be seen from our computation, this number is exactly the number of BCH codes concatenated to the LDPC code. Since after each outer iteration, at least one additional BCH code decoder should claim to be successful in decoding, it is clear that at most 23 outer iterations are required to have all BCH code decoders obtaining valid codewords.
- The maximum number of outer iterations for the $(511, 475, 4)$ BCH code and $\mathcal{C}_1$ is $5878/511 \approx 11$.
- The maximum number of outer iterations for the $(255, 223, 4)$ BCH code and $\mathcal{C}_2$ is $3835/255 \approx 15$.
- The maximum number of outer iterations for the $(511, 475, 4)$ BCH code and $\mathcal{C}_2$ is $3835/511 \approx 7$.

Now we illustrate the details of all figures.

1. Figures 4.1-4.4 show the BER performances of feedback decoding introduced in Section 3.2.
   (a) Presented in Figure 4.1 are the performances of the LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the BFA.
   (b) The codes tested in Figure 4.2 are the same as those in Figure 4.1 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

(c) The codes tested in Figure 4.3 are the same as those in Figure 4.1 except that the SPA is employed instead of the BFA.

(d) The codes tested in Figure 4.4 are the same as those in Figure 4.3 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

2. Figures 4.5-4.8 show the BER performances of feedback decoding refined under Strategy 1 described in Section 3.3.1.

   (a) Presented in Figure 4.5 are the performances of the LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

   (b) The codes tested in Figure 4.6 are the same as those in Figure 4.5 except $(511, 475, 4)$ BCH codes are used in place of $(255, 223, 4)$ BCH codes.

   (c) The codes tested in Figure 4.7 are the same as those in Figure 4.5 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

   (d) The codes tested in Figure 4.8 are the same as those in Figure 4.6 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

3. Figures 4.9-4.12 show the BER performances of feedback decoding under Strategy 2 refinement introduced in Section 3.3.2.

   (a) Presented in Figure 4.9 are the performances of the LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Stratege-2-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with feedback loop. The LDPC code is decoded by the SPA.

(b) The codes tested in Figure 4.10 are the same as those in Figure 4.9 except $(511, 475, 4)$ BCH codes are used in place of $(255, 223, 4)$ BCH codes.

(c) The codes tested in Figure 4.11 are the same as those in Figure 4.9 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

(d) The codes tested in Figure 4.12 are the same as those in Figure 4.10 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

4. Figures 4.13-4.16 show the BER performances of feedback decoding under Strategy 3 refinement introduced in Section 3.3.3.

   (a) Presented in Figure 4.13 are the performances of the LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Stratege-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

   (b) The codes tested in Figure 4.14 are the same as those in Figure 4.13 except $(511, 475, 4)$ BCH codes are used in place of $(255, 223, 4)$ BCH codes.

   (c) The codes tested in Figure 4.15 are the same as those in Figure 4.13 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

   (d) The codes tested in Figure 4.16 are the same as those in Figure 4.14 except $\mathcal{C}_2$ is used in place of $\mathcal{C}_1$.

5. Figures 4.17-4.20 show the BER performances of feedback decoding under three different strategies.

   (a) Presented in Figure 4.17 are the performances of the LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

(b) Presented in Figure 4.18 are the performances of the LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

(c) Presented in Figure 4.19 are the performances of the LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

(d) Presented in Figure 4.20 are the performances of the LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

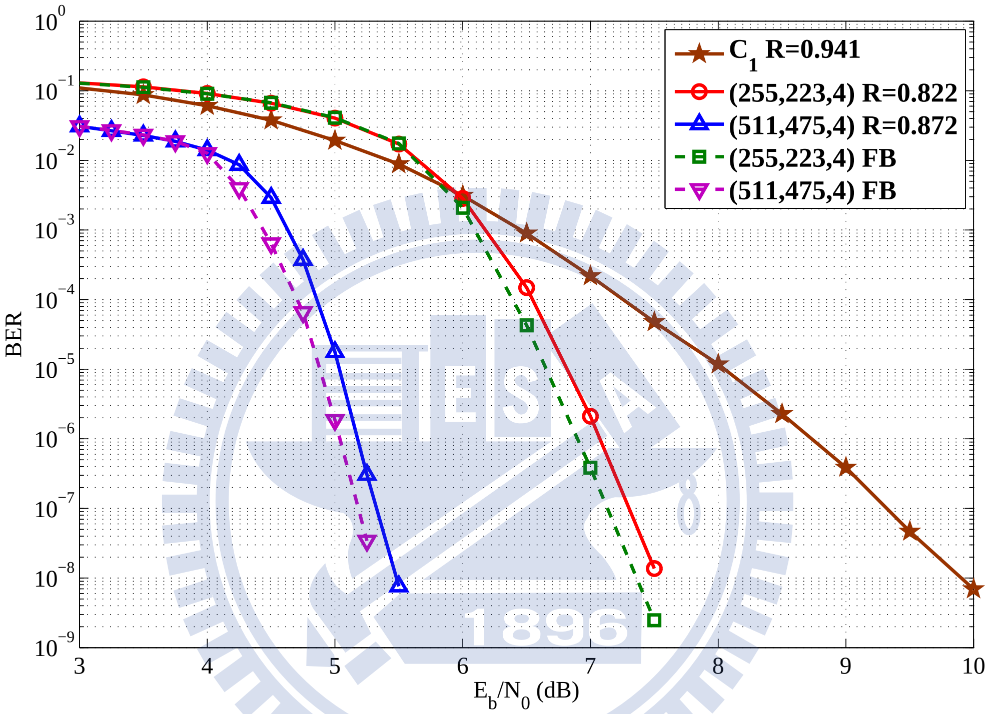

Figure 4.1: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the BFA.

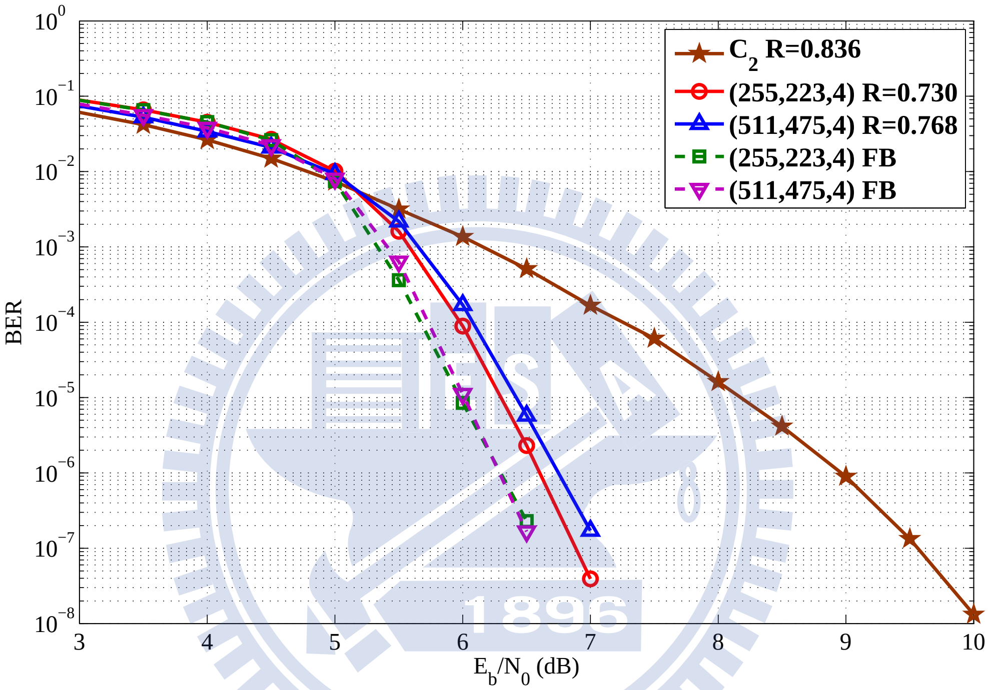

Figure 4.2: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the BFA.

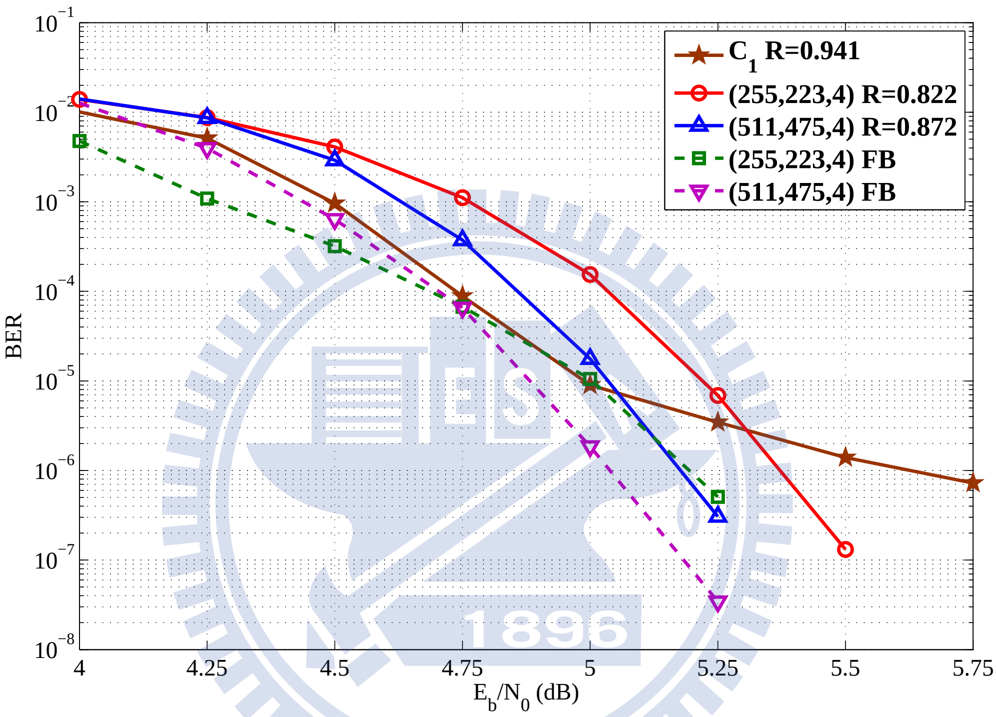

Figure 4.3: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

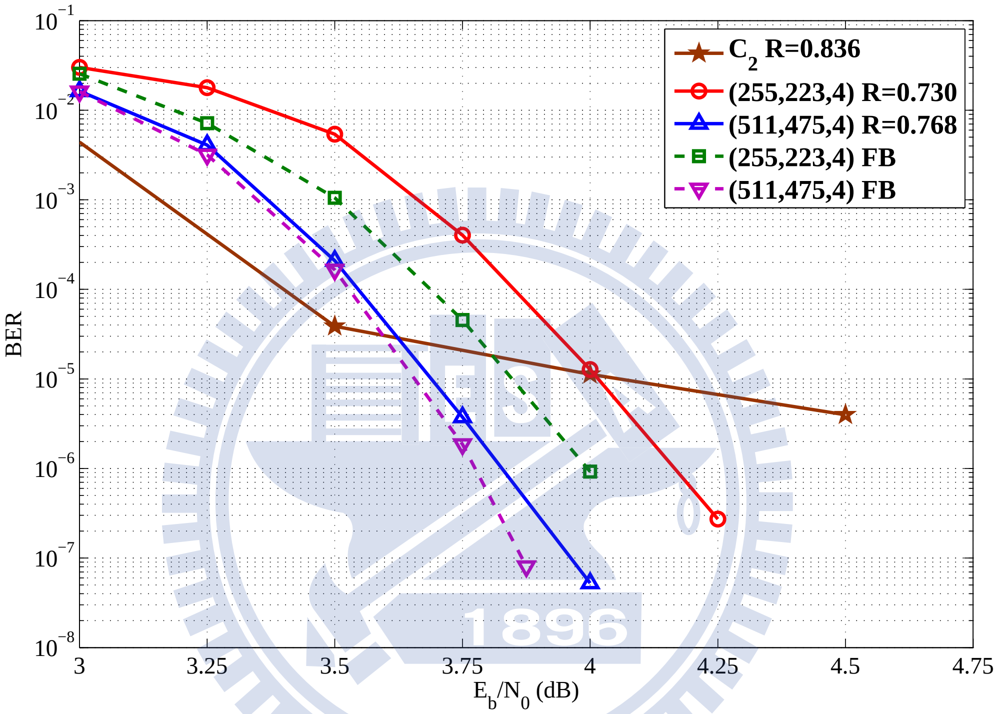

Figure 4.4: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

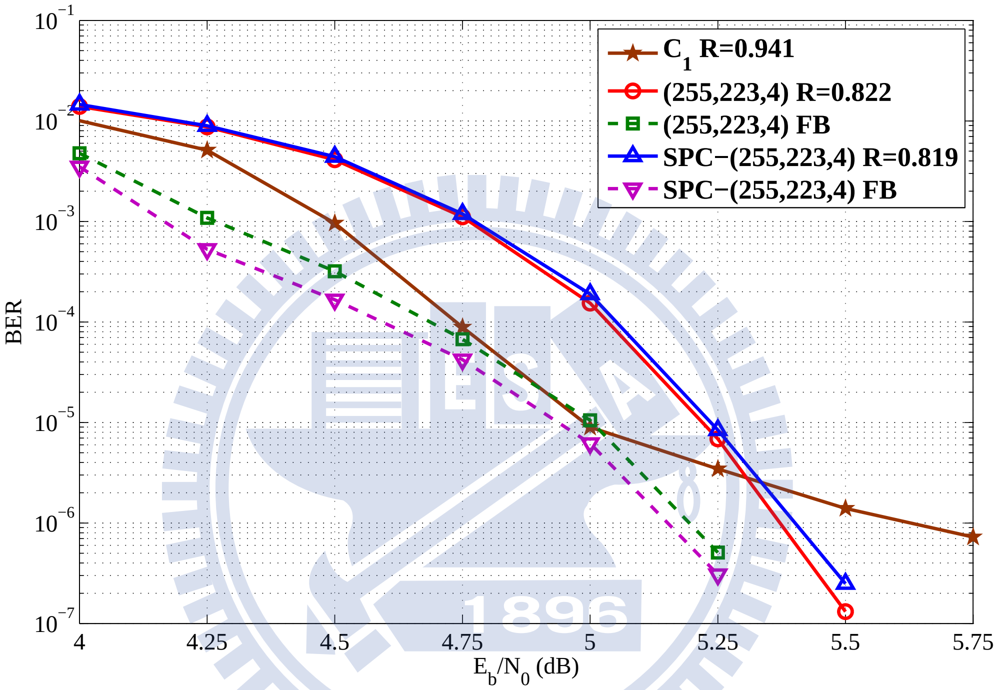

Figure 4.5: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

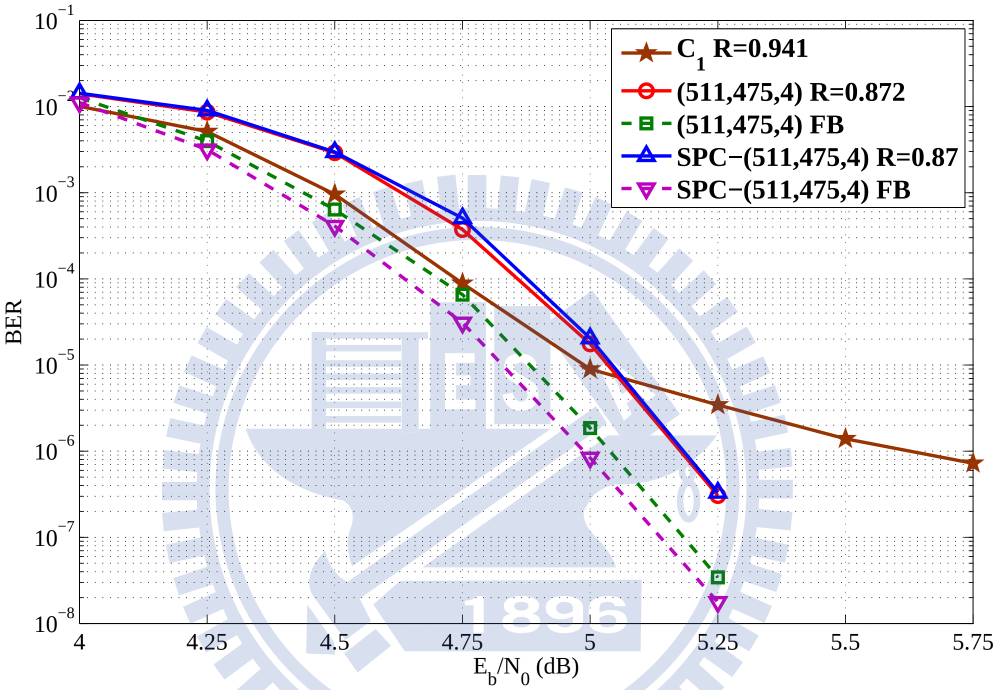

Figure 4.6: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

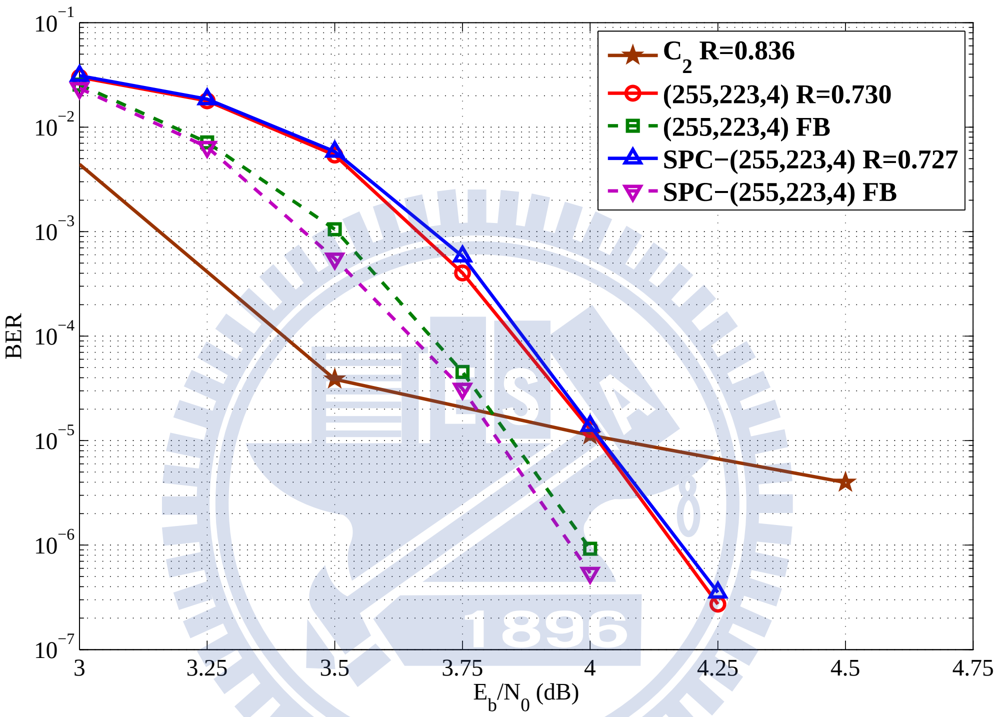

Figure 4.7: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

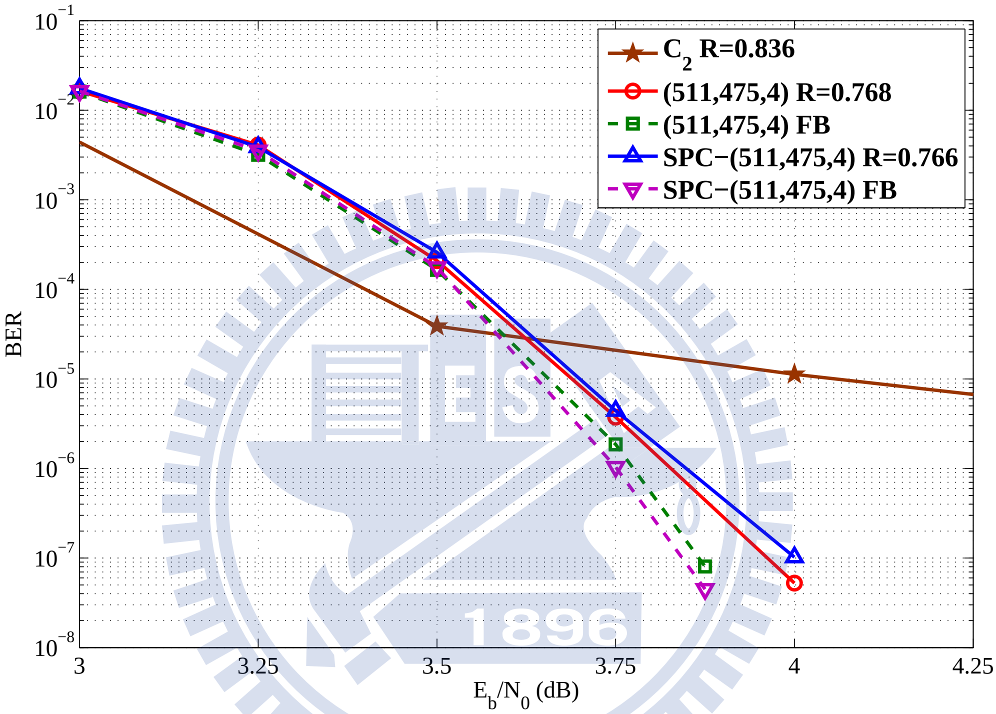

Figure 4.8: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The LDPC code is decoded by the SPA.

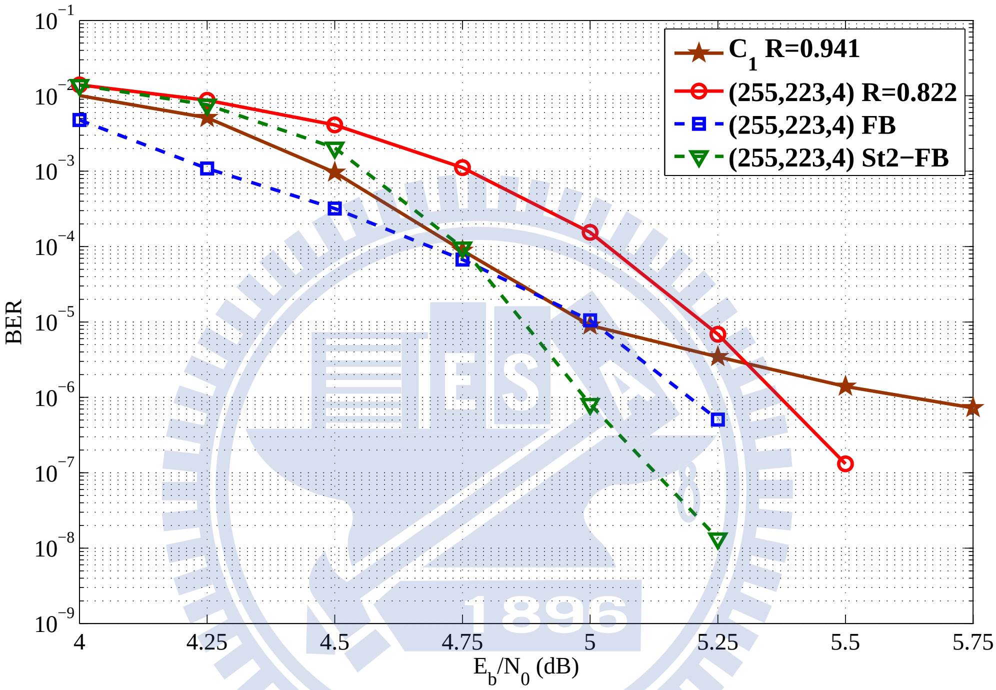

Figure 4.9: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with feedback loop. The LDPC code is decoded by the SPA.

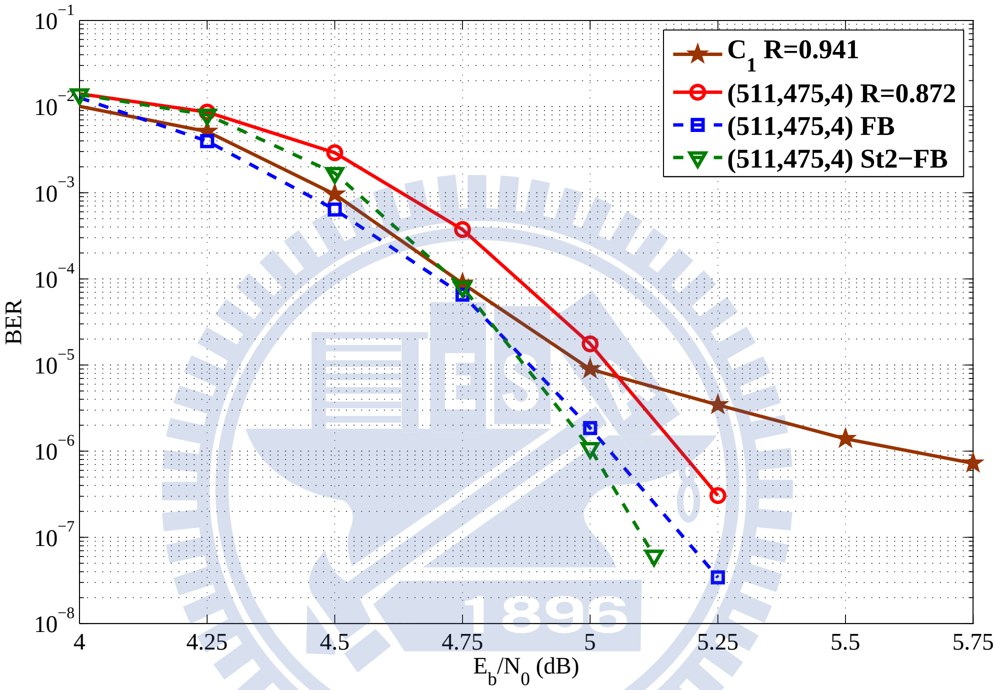

Figure 4.10: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with feedback loop. The LDPC code is decoded by the SPA.

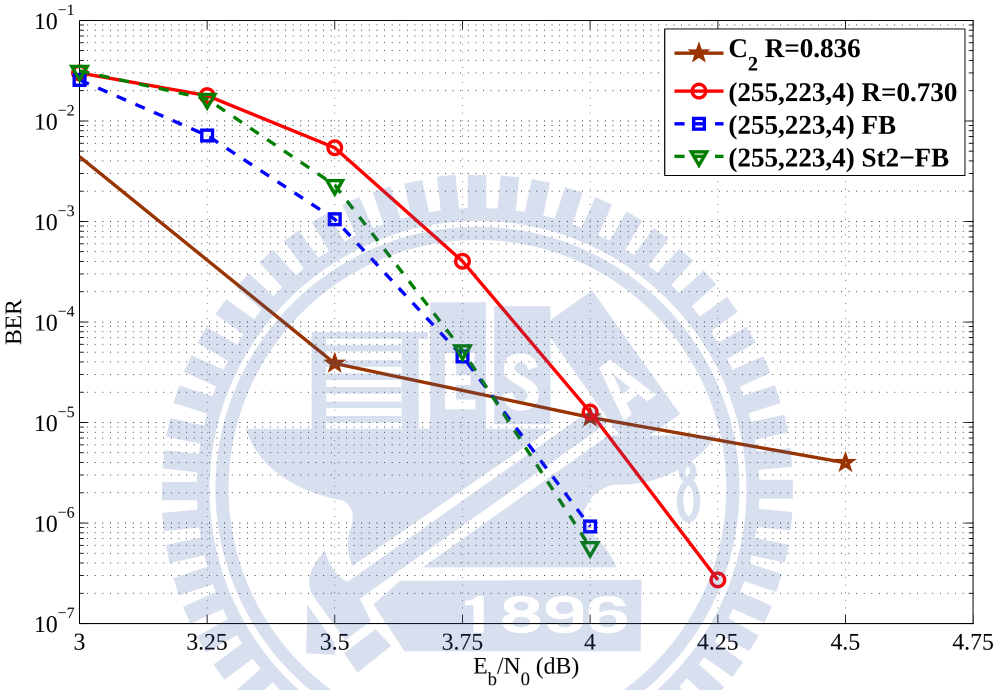

Figure 4.11: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with feedback loop. The LDPC code is decoded by the SPA.

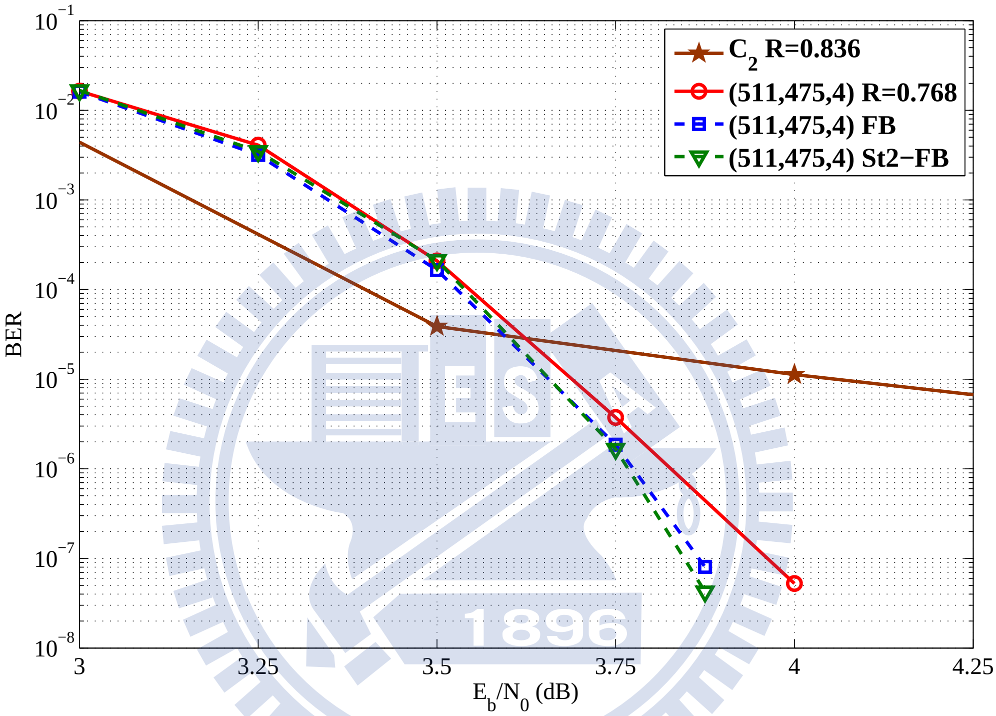

Figure 4.12: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with feedback loop. The LDPC code is decoded by the SPA.

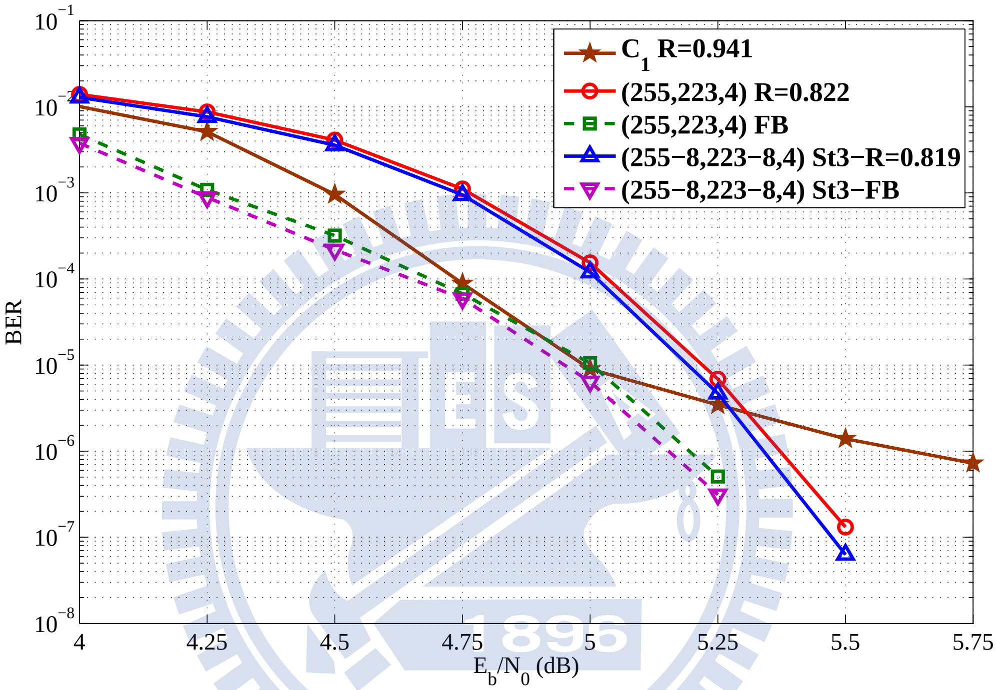

Figure 4.13: Performances of the rate-0.941 (6350, 5978) QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with (255, 223, 4) BCH codes decoded with and without feedback loop, and its corresponding Strategy-3-enhanced concatenated system with (255, 223, 4) BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 8, and the LDPC code is decoded by the SPA.

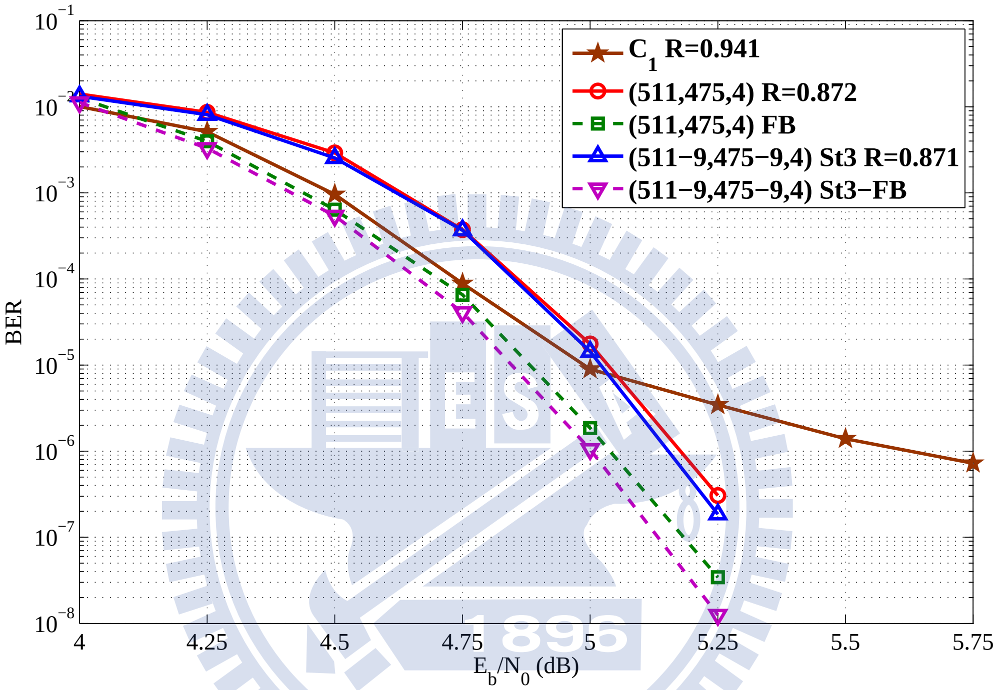

Figure 4.14: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 9, and the LDPC code is decoded by the SPA.

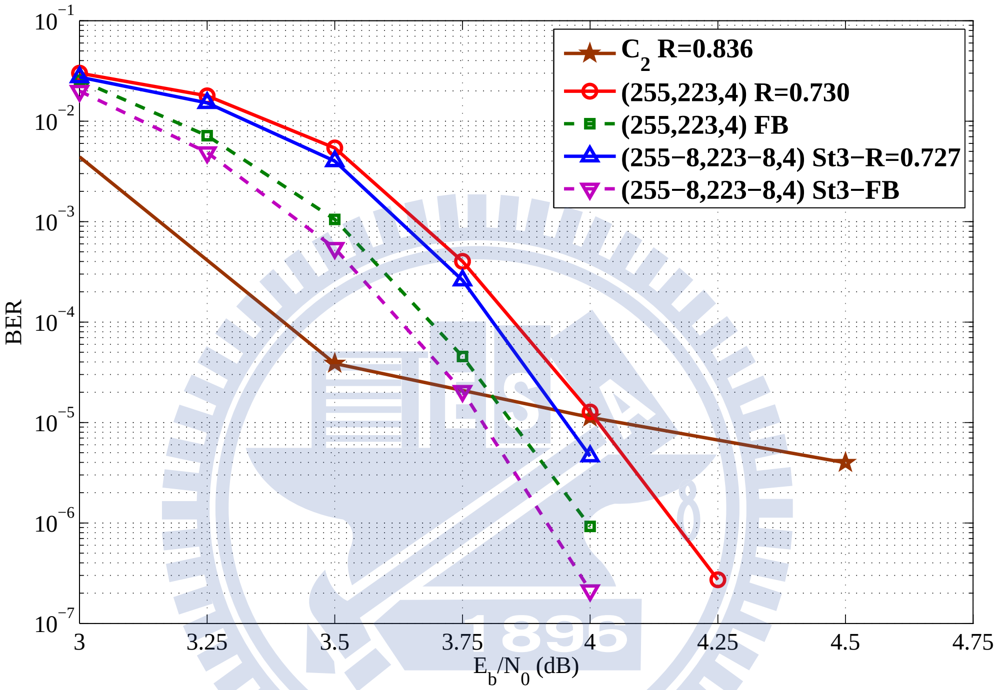

Figure 4.15: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 8, and the LDPC code is decoded by the SPA.

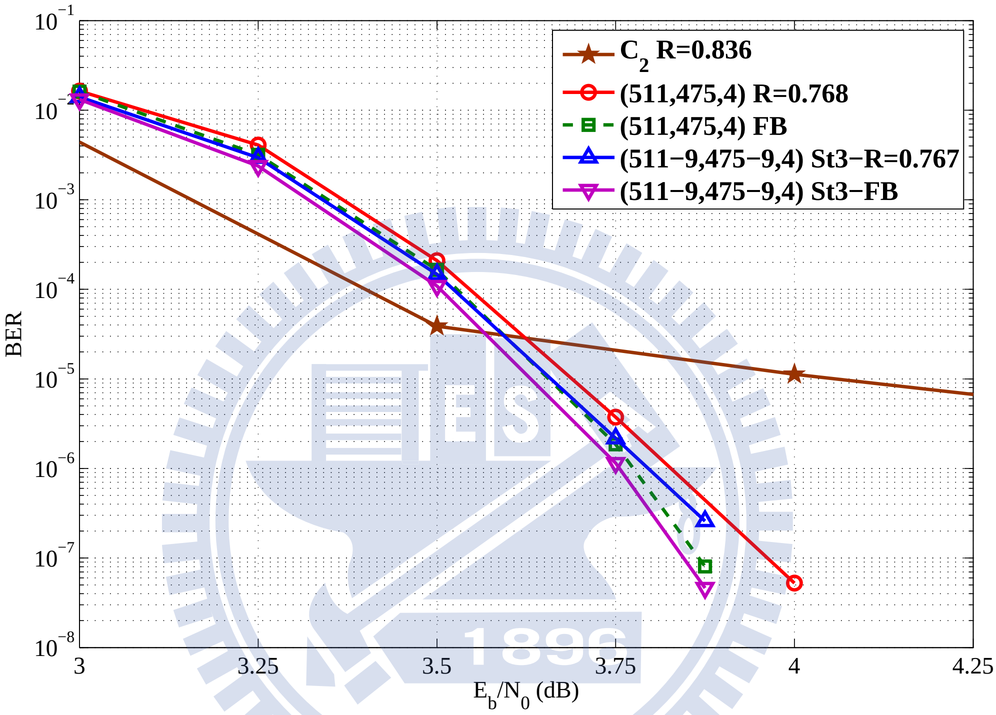

Figure 4.16: Performances of the rate-0.836 (4590, 3835) QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with (511, 475, 4) BCH codes decoded with and without feedback loop, and its corresponding Strategy-3-enhanced concatenated system with (511, 475, 4) BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 9, and the LDPC code is decoded by the SPA.

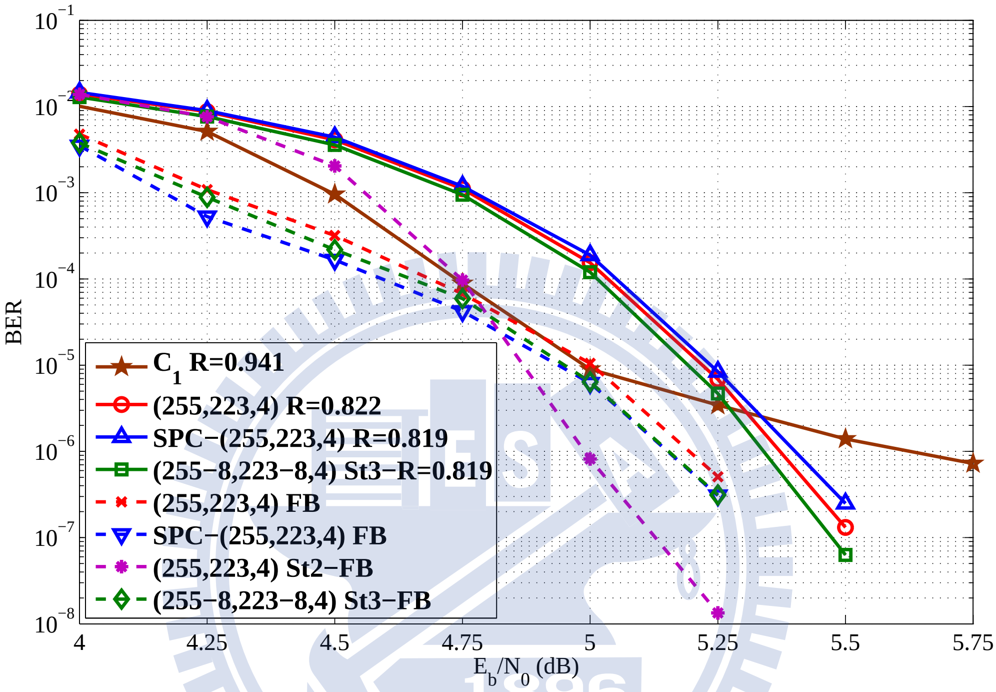

Figure 4.17: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 8, and the LDPC code is decoded by the SPA.

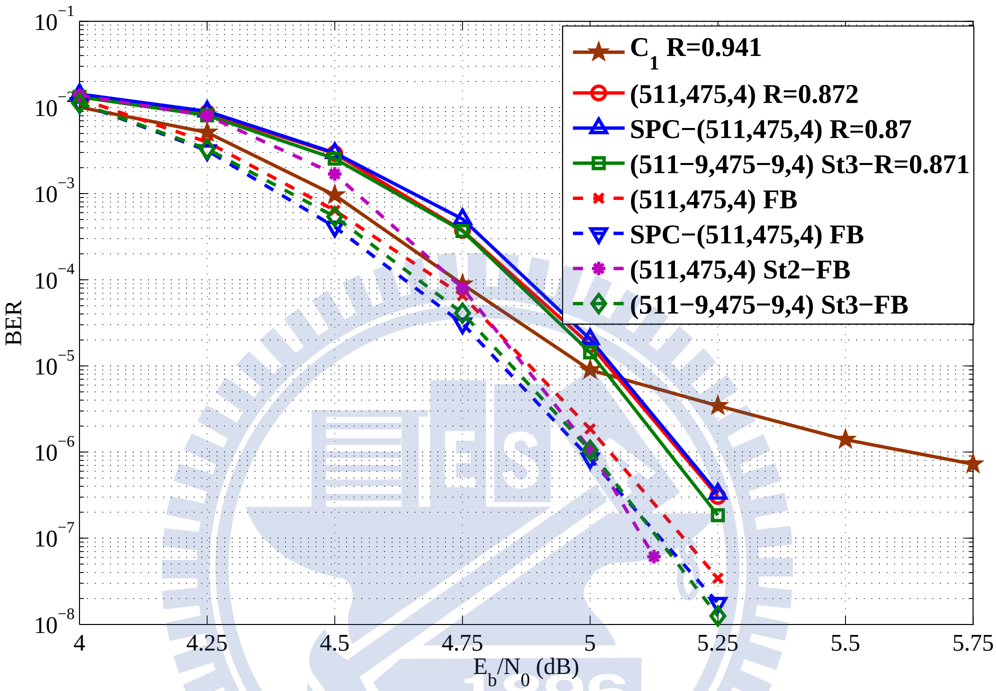

Figure 4.18: Performances of the rate-0.941 $(6350, 5978)$ QC-LDPC code $\mathcal{C}_1$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 9, and the LDPC code is decoded by the SPA.

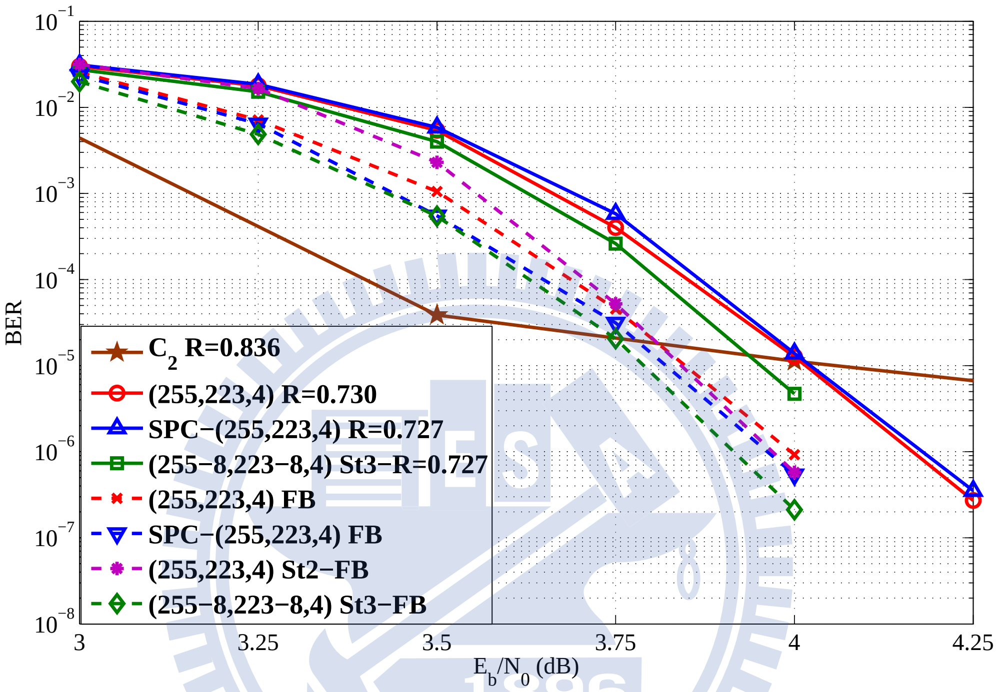

Figure 4.19: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(255, 223, 4)$ BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 8, and the LDPC code is decoded by the SPA.

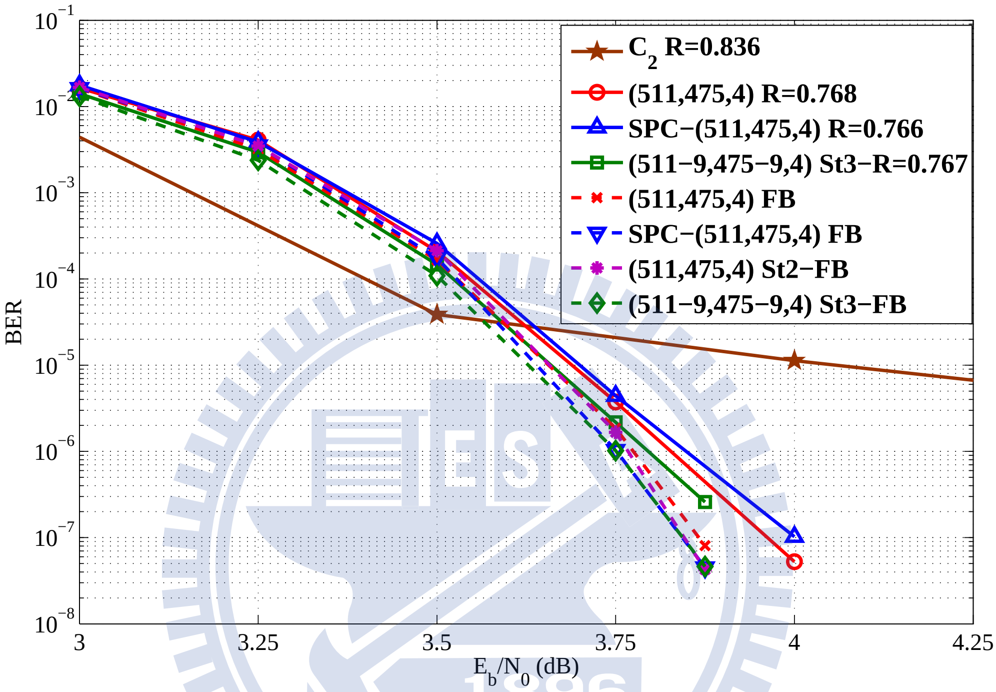

Figure 4.20: Performances of the rate-0.836 $(4590, 3835)$ QC-LDPC code $\mathcal{C}_2$, its corresponding concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding SPC-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop, and its corresponding Strategy-2-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with feedback loop, and its corresponding Strategy-3-enhanced concatenated system with $(511, 475, 4)$ BCH codes decoded with and without feedback loop. The parameter λ chosen in Strategy 3 is 9, and the LDPC code is decoded by the SPA.

4.2 Remarks

From Figures 4.1-4.4, it can be seen that the error floors are greatly improved by the concatenated coding scheme. Further improvement can be obtained by feedback iterative decoding. In some cases, however, the BER performances become worse in the waterfall region due to perhaps the rate loss. These figures also tell us that a larger BCH code may be favored; but the improvement from a larger BCH code is only limited. It is then reasonable to expect that further increasing the side of the BCH code may not be of visible help to the system performance. Note that we choose the BCH code that can resolve the dominant trapping set of the smallest size, i.e., 4. For a larger dominant trapping set, e.g., 6, in the Tanner graph, the errors will be fallen into one BCH code block with a higher probability if a larger BCH code is chosen. Thus they cannot be corrected by the joint decoding. In such case, feedback information is also less reliable and hence will not help the inner decoder to resolve the error trap.

Next, we examine the effectiveness of Strategy 1 for improving the performance of the concatenated coding system. The simulation results are summarized in Figure 4.5-4.8. From these figures, it can be observed that the overall BER performances can be a little improved by Strategy 1 in spite that further rate loss is resulted (e.g., 0.003 and 0.002 respectively in Figures 4.5 and 4.6). In addition, such improvement is more evident at the case when feedback iterative decoding is performed.

In Figures 4.9-4.12, Strategy 2 are used instead of Strategy 1. Different from Strategy 1, the BER performances are only a little improved at high SNR, while they become evidently worse at low to medium SNRs. This is perhaps because that the Euclidean distance between the channel outputs and decoded results are less reliable at low to medium SNRs.

Now we use Strategy 3 on the concatenated coding scheme in Figures 4.13-4.16. The values of λ adopted are indicated in the captions of these figures. By such choices of λ, the rate loss is only 0.003 (respectively, 0.001) in Figure 4.13 (respectively, Figure 4.14). In comparison with Strategy 1, Strategy 3 can achieve a smaller rate loss and maintain a better improvement by a proper choice of λ.

For a better understanding of feedback iterative decoding, Tables 4.1-4.4 tell how much the outer iterations help improving the performance. As an example, in Table 4.1, the value located at (x, y) indicates the number of occurrences during our simulations for initial (outer) iteration to have y BCH successfully decoding blocks but the final outer iteration to improve up to x BCH successfully decoding blocks. Clearly, x should be no less than y; so the elements in lower triangle of the table should be ignored. These tables again confirmed that except for few cases, feedback iterative decoding can make successful the decoding of all BCH decoders.

Finally, Tables 4.5-4.8 list the statistics of the number of outer iterations performed in our simulations. Note that the outer iteration will have an early stop when all BCH code decoders decode successfully. From these tables, it can be easily seen that the complexity of feedback iterative decoding is prohibitively small. In most cases, only 2 outer iterations are sufficient.

Table 4.1: Effective indices of feedback deeding for the concatenation of $(255, 223, 4)$ BCH codes and the $(6350, 5978)$ QC-LDPC code. The index located at (x, y) indicates the number of occurrences during our simulations for initial (outer) iteration to have y BCH successfully decoding blocks but the final outer iteration to improve up to x BCH successfully decoding blocks. The SNR simulated is 5 dB.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0
12	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0
13	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	1
14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0
15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0
16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	6
17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	6	26
18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0	1	6	57
19	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	2	0	1	16	153
20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	2	58	423
21	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	8	85	1120
22	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	113	2962
23	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	47043

Table 4.2: Effective indices of feedback deeding for the concatenation of $(511, 475, 4)$ BCH codes and the $(6350, 5978)$ QC-LDPC code. The index located at (x, y) indicates the number of occurrences during our simulations for initial (outer) iteration to have y BCH successfully decoding blocks but the final outer iteration to improve up to x BCH successfully decoding blocks. The SNR simulated is 5 dB.

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	—	0	0	0	0	0	0	0	0	0	0	0
2	—	—	0	0	0	0	0	0	0	0	0	1
3	—	—	—	0	0	1	2	0	0	0	2	4
4	—	—	—	—	0	3	0	0	1	1	0	30
5	—	—	—	—	—	1	1	3	1	4	4	52
6	—	—	—	—	—	—	20	9	2	4	13	115
7	—	—	—	—	—	—	—	14	9	4	12	219
8	—	—	—	—	—	—	—	—	18	9	32	477
9	—	—	—	—	—	—	—	—	—	18	45	906
10	—	—	—	—	—	—	—	—	—	—	67	1410
11	—	—	—	—	—	—	—	—	—	—	—	430298

Table 4.3: Effective indices of feedback deeding for the concatenation of $(255, 223, 4)$ BCH codes and the $(4590, 3835)$ QC-LDPC code. The index located at (x, y) indicates the number of occurrences during our simulations for initial (outer) iteration to have y BCH successfully decoding blocks but the final outer iteration to improve up to x BCH successfully decoding blocks. The SNR simulated is 3.5 dB.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	—	7	0	1	1	1	0	1	0	0	0	0	0	2	5	8
2	—	—	0	4	1	2	1	1	0	0	1	0	0	2	4	21
3	—	—	—	11	3	2	3	2	1	1	0	0	0	2	10	40
4	—	—	—	—	4	1	0	1	1	0	0	2	1	2	16	48
5	—	—	—	—	—	2	1	2	0	1	0	0	0	3	18	55
6	—	—	—	—	—	—	4	0	0	0	0	0	0	2	30	78
7	—	—	—	—	—	—	—	0	2	0	0	1	1	2	29	80
8	—	—	—	—	—	—	—	—	0	0	0	1	1	2	23	97
9	—	—	—	—	—	—	—	—	—	0	2	1	0	1	18	78
10	—	—	—	—	—	—	—	—	—	—	0	0	0	1	15	80
11	—	—	—	—	—	—	—	—	—	—	—	0	0	0	16	84
12	—	—	—	—	—	—	—	—	—	—	—	—	0	0	10	73
13	—	—	—	—	—	—	—	—	—	—	—	—	—	0	6	79
14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	2	59
15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1556

Table 4.4: Effective indices of feedback deeding for the concatenation of $(511, 475, 4)$ BCH codes and the $(4590, 3835)$ QC-LDPC code. The index located at (x, y) indicates the number of occurrences during our simulations for initial (outer) iteration to have y BCH successfully decoding blocks but the final outer iteration to improve up to x BCH successfully decoding blocks. The SNR simulated is 3.5 dB.

	0	1	2	3	4	5	6	7
0	245	0	0	0	0	0	0	0
1	—	23	2	1	0	0	0	86
2	—	—	9	0	1	1	2	64
3	—	—	—	2	0	0	3	41
4	—	—	—	—	0	0	2	24
5	—	—	—	—	—	0	2	12
6	—	—	—	—	—	—	3	23
7	—	—	—	—	—	—	—	35007

Table 4.5: Statistics of the number of outer iterations required in our simulations. The concatenated system consists of $(255, 223, 4)$ BCH codes and an $(6350, 5978)$ LDPC code. The SNR simulated is 5 dB.

Number of outer iterations	1	2	3	4	5	6	7
Number of occurrences	47157	4919	14	0	1	0	0

Table 4.6: Statistics of the number of outer iterations required in our simulations. The concatenated system consists of $(511, 475, 4)$ BCH codes and an $(6350, 5978)$ LDPC code. The SNR simulated is 5 dB.

Number of outer iterations	1	2	3	4	5	6	7
Number of occurrences	430365	3355	69	20	3	0	0

Table 4.7: Statistics of the number of outer iterations required in our simulations. The concatenated system consists of $(255, 223, 4)$ BCH codes and an $(4590, 3835)$ LDPC code. The SNR simulated is 3.5 dB.

Number of outer iterations	1	2	3	4	5	6	7
Number of occurrences	1558	969	130	35	30	8	6

Table 4.8: Statistics of the number of outer iterations required in our simulations. The concatenated system consists of $(511, 475, 4)$ BCH codes and an $(4590, 3835)$ LDPC code. The SNR simulated is 3.5 dB.

Number of outer iterations	1	2	3	4	5	6	7
Number of occurrences	35009	511	30	6	1	0	0

Chapter 5

Conclusion and Future Work

In this thesis, for a given inner LDPC code and a given codeword length of outer BCH codes, we suggest a method to determine the error correcting capability of the outer BCH codes based on the knowledge of the dominant trapping sets of the inner LDPC code. In addition, we present a feedback-based iterative decoding scheme for the BCH and LDPC concatenated coding system. Due to the algebraic decoding characteristic of the outer BCH code decoders, generating soft outputs for the outer code decoders to fit the need of the iterative decoding between inner and outer codes is unnatural. Thus, the feedback-based iterative decoding scheme requires an elaborate design. In the end, we additionally propose three strategies to improve the reliability of the feedback information and hence the performance of the feedback-based iterative decoding can be further enhanced. All the ideas we propose are then verified by simulations.

At the current stage, only the information of BCH code decoders is fed back to the inner decoder, but the inner decoder outputs have little effect on the outer decoders. Thus, if the BCH code decoders could make use of the structural characteristic of the QC-LDPC inner code so as to form a positive interactive cycle between the inner and outer decoders, the system performance may be further improved. This could be an interesting future work for this subject.

Bibliography

[1] F. Sun, K. Rose, and T. Zhang, "On the use of strong BCH codes for improving multi-level NAND flash memory storage capacity", in *Proc. IEEE Workshop on SiPS*, Banff, Canada, October 2006.

[2] W. Liu, J. Rho, and W. Sung, "Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories", in *Proc. IEEE Workshop on SiPS*, Banff, Canada, October 2006, pp. 303-308.

[3] X. Wang *et al.* (2011). Error Correction Codes and Signal Processing in Flash Memory [Online]. Available: http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/

[4] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed., New Jersey:Prentice Hall, 2004.

[5] L. Lan, L. Zeng, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: A finite field approach", *IEEE Trans. Inform. Theory*, vol. 53, no. 7, pp. 2429-2458, July 2007.

[6] L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes", *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71-81, January 2006.

[7] Y. Dai, N. Chen, and Z. Yan, "Memory efficient decoder architectures for quasi-cyclic LDPC codes", *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 9, pp. 2898-2911, October 2008.

[8] R. Gallager, "Low-density arity-check codes", MIT Press, Cambridge, MA, 1963.

[9] T. J. Richardson, "Error floors of LDPC codes", in *Proc. 41st Annual Allerton Conf. on Commun., Control and Computing*, Monticello, IL, USA, October 2003, pp. 1426-1435.

[10] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "A class of low-density parity-check codes constructed based on Reed-Solomon codes with two information symbols", *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 317-319, July 2003.