QUASI-CYCLIC LOW-DENSITY PARITY-CHECK (QC-LDPC) CODES WITH MAXIMIZED GIRTH PROPERTY BASED ON PROGRESSIVE EDGE GROWTH (PEG) ALGORITHM

PATANASAK PROMPAKDEE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN DATA STORAGE TECHNOLOGY

(INTERNATIONAL PROGRAM)

INTERNATIONAL COLLEGE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2012

KMITL-2012-IC-M-005-004

Thesis: Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) codes with maximized girth property based on progressive edge growth (PEG) algorithm

Student: Mr. Patanasak Prompakdee

Student ID: 52600626

Degree: Master of Engineering (M.Eng)

Program: Data Storage Technology

Year: 2011

Thesis Advisor: Assoc. Prof. Dr. Pornchai Supnithi

ABSTRACT

Low-density parity-check codes (LDPC) codes are linear block codes which approach the near capacity performance (Shannon's limit) and offer implementable decoders for practical application. These codes encode using the sparse parity-check matrix. We propose a method to construct the parity-check matrices of the quasi-cyclic (QC) LDPC codes with maximized girth property based on the progressive edge-growth (PEG) algorithm. Based on the PEG algorithm, these codes can eliminate short cycles, while the QC algorithm provides simple implementation. In this work, we illustrate that the codes constructed with the proposed algorithm have the superior performance to the previously proposed PEG-QC codes in an additive white Gaussian noise (AWGN) channel with binary inputs. At the bit error rate (BER) of 10^{-6}, the proposed code gains 0.05 dB compared with the previous code.

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to my advisor Assoc. Professor Dr. Pornchai Supnithi in valuable advice to me for this research work during the entire period of time my study. I would also like to thank Mr. Watid Phakphisut and team members for productive comments.

I also thank "Western Digital Thailand" for the great moral support in this data storage technology program. In addition, I would like to thank all my WD team members who supported my research.

Finally, I would like to thank my family who has given me much encouragement to do this Master Degree in Data Storage Technology at King Mongkut's instiute of technology Ladkrabang.

PATANASAK PROMPAKDEE

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF FIGURES

Fig **Page**

LIST OF FIGURES (Cont.)

Fig **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

INTRODUCTION

1.1 STATEMENT AND SIGNIFICANCE OF THE PROBLEM

Continuing rapid increase in data storage requirement towards terabytes or petabytes per device supports new applications such as a high-definition multimedia interface (HDMI) as well as social media revolution which need extremely high storage capacity. In addition, the continuing integration of computers, internet and electronic application require high-speed central processing unit (CPU) and high transfer rate from storage device. The hard disk drive (HDD), optical disk drive, magnetic tape drives and solid state drives are commonly used nowadays. Among them, the hard disk drives offer the best combination of capacity, speed and price. Consequently, it is used as the major storage device in the computer system.

A hard disk drive includes the magnetic disk, the read/write head, the actuator arm, the read/write channel and controller. The magnetic disk is coated with the magnetic material which stores the user data along concentric tracks and each track is divided into sectors. The write head applies the magnetic field to the disk to store data and the read head senses the magnetic field from the disk to translate into the electronic signal. Both heads are installed at the end of actuator arm that seeks to the required track to read/write data based on the controller command. The write channel converts the user data to the write current and bias to write heads. The read channel renders the read signal from read heads and recovers the user data. The controller interface with the host computer, controls actuator arm and performs the error correction codes (ECC) encoding and decoding process.

The signal processing on the hard disk drives is important to protect information data from random noise and other unpredictable disturbances. The signal processing schematic in Fig. 1.1 presents the encoder and the decoder which prevent the bit errors in a communication system. The massage bits are encoded and the parity bits are added (typically called "code word") before recording on the magnetic disk. The additional bits are used to prevent and correct the erroroneous bits due to communication systems. Then the modulator converts the code word into the write current waveform and transfers to the write head to record on the magnetic disk. The read head gets the read-

waveform on the magnetic disk then passes it to the demodulator. The demodulator interprets read-waveform to code word and transfers to the decoder then the code word is decoded, the error bits from noise are corrected at this process and the additional bits are removed.

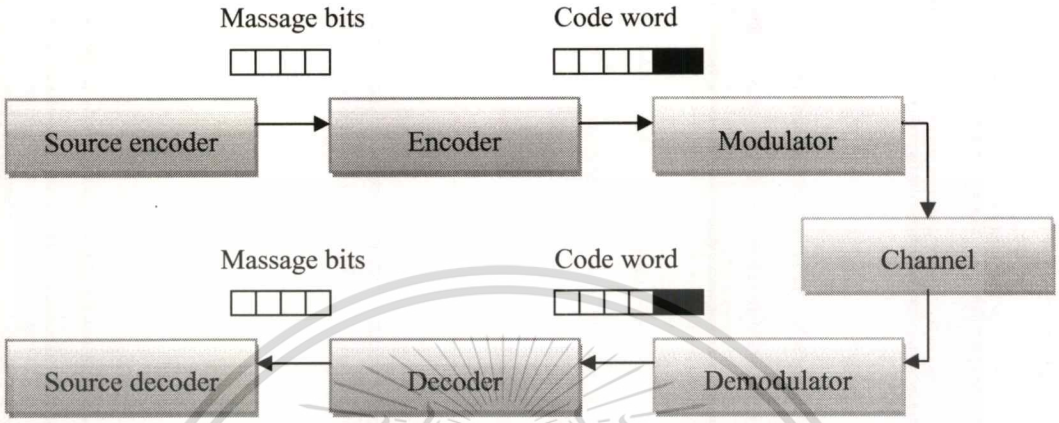


Figure 1.1 The signal processing schematics of a communication system

Given the high speed data transfer in a communication system, the main challenge on signal processing to maintain low bit error rate is noise. The noise on the play back signal arises from the electronics noise which contribute from the read head and the pre-amplifier, and the medium noise which leads to magnetization variation on medium and disk defect. The playback signal contains not only the noise, but also the inter-symbol interference (ISI) from adjacent bits. The advanced equalization, detection and coding techniques are commonly used to ensure reliable detection of the information.

Coding is usually used to achieve reliable transmission of information. Codes ensure higher noise tolerance at the receiver by adding the redundancy to the user data to achieve a better separation of data sequences. Typically, the codes are divided to two categories: convolution codes such a turbo codes and block codes such a low-density parity-check (LDPC) codes [1]. The original concept for turbo codes concatenate two or more constituent codes with random interleaver between concatenations; therefore, the codes nearly satisfied the random selection near-capacity performance, but can be practically decoded by decoding each constituent code separately and passing information between the decoders. The other code is LDPC code, instead of the random interleaving; the LDPC

codes approximate random code selection in that the code words use a large and randomly generated sparse parity-check matrix. The iteration technique on both turbo codes and LDPC codes present the best decoding performance and approach the Shannon's limit [2]. Thus, turbo codes and LDPC codes are of much interest in the recent years. There are many research works on both codes. However, the turbo codes are considered to have higher complexity in encoding and decoding than the LDPC codes. The LDPC codes are currently in the standards of wireless communication and data storage technology.

1.2 GOAL AND OBJECTIVES

An LDPC code is a class of linear block codes which provide a near-capacity performance on a large collection of data transmission and storage channels. Researchers are interested on LDPC code because its relatively simple implementation on the decoder compared with the turbo codes. The objectives of this research is to

1.2.1 study the fundamental technique for random LDPC codes with Progressive Edge Growth (PEG) algorithm and Quasi-Cyclic (QC) LDPC codes [3-8].

1.2.2 perform the feasibility to improve LDPC codes and QC LDPC codes based on PEG algorithm by optimizing candidate nodes selection to maximize local cycle during the construction of the parity-check matrix.

1.2.3 determine and optimize the decoding performance of developed codes versus the current LDPC codes in Perpendicular Magnetic Recording (PMR).

1.3 SCOPE OF THE STUDY

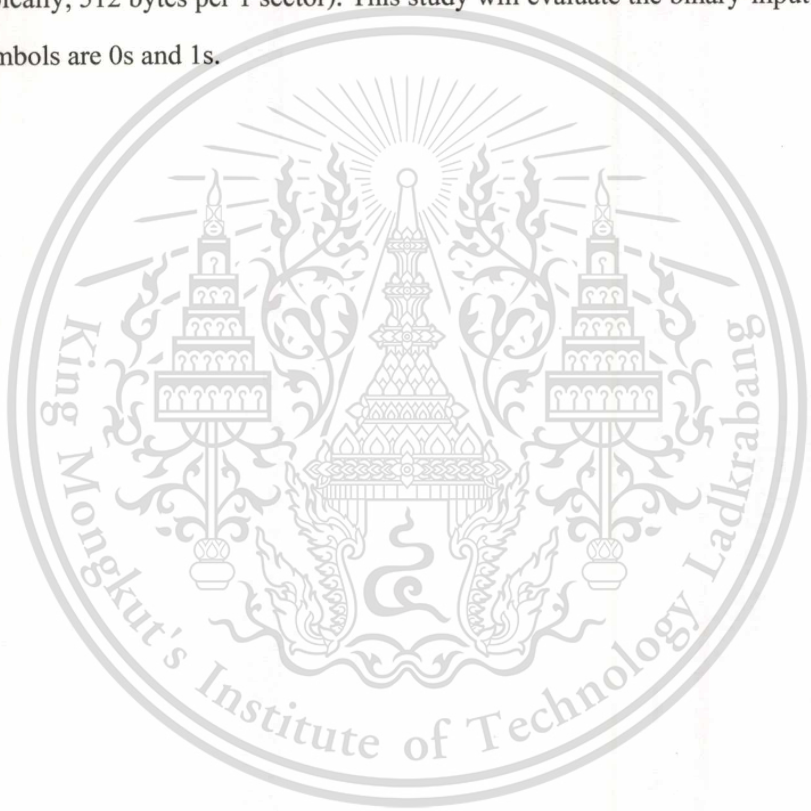
The decoding performance on signal processing is affected from various factors such error correction codes, equalizer, electronic noise, medium noise and others. We will focus on the decoding performance based on error correction and the electronic noise; the scopes of this study are as follow:

1.3.1 This study will determine the decoding performance for LDPC codes based on PEG and QC algorithm. This would help to determine the suitable method to construct LDPC codes for practical decoder.

1.3.2 This study will simulate LDPC codes in an addive white Gaussian noise (AWGN) channel and compare the current LDPC code design with proposed design.

1.4 LIMITATION OF THE STUDY

Since the commercial hard disk drives include error correction, equalizer, controller and others. In practice, it is difficult to control external factors and environment to study and evaluate the decoding performance for LDPC codes. Thus, the entire evaluations and the study are performed on the computer program. We will study the data size based on the data sector available on actual hard disk drives (typically, 512 bytes per 1 sector). This study will evaluate the binary-input system; that is information symbols are 0s and 1s.



CHAPTER 2

BASIC MAGNETIC RECORING

2.1 Magnetic recording structure

The basic structure of a magnetic recording system is shown in Fig. 2.1. It comprises the read/write head which is an electromagnet with ferrite core, the rotating magnetic disk with a ferromagnetic surface, the actuator arm which connects to read/write head known as head stack assembly, the voice coil motor that the control arm seeks to required track and the spindle motor controls rotating disk with a constant speed. The entire elements are contained in the drive with the top cover and vacuum internal environment to prevent the contaminations because they harm the read/write head while flying over the rotating media. Moreover, it protects the interruption on the play back signal from the external magnetic field. The printed circuit board assembly (PCBA) is attached in the drive; it includes the error correction, read channel, controller interface. Typically, the PCBA endures the contamination and external magnetic field.

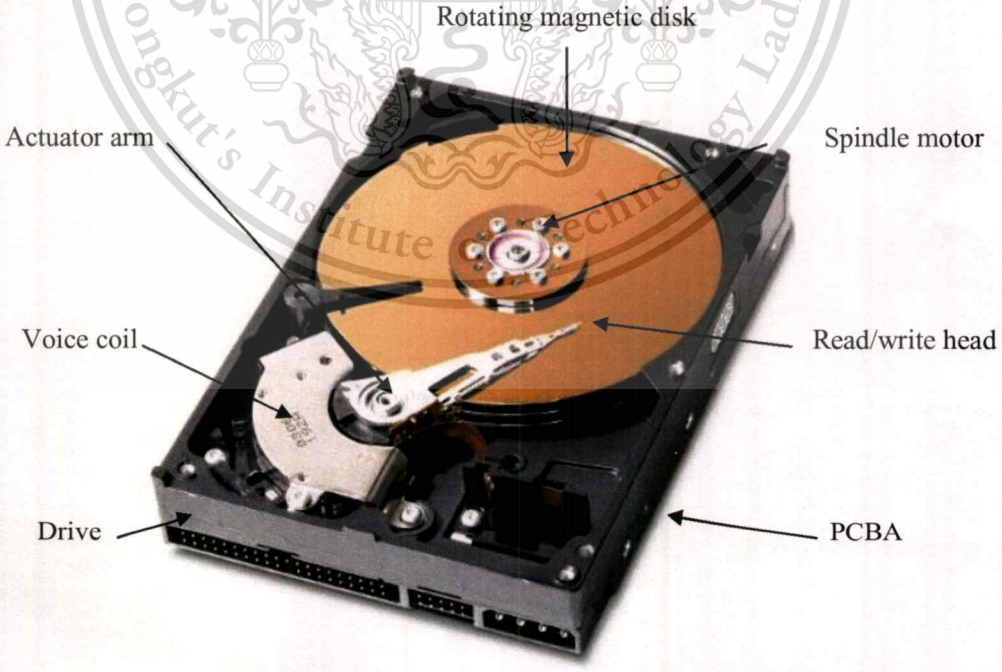


Figure 2.1 Basic structure of magnetic recording system

2.2 Magnetic recording technology

2.2.1 Longitudinal recording

In the longitudinal recording, the magnetizations of bits are aligned along the recording medium surface as presented in Fig. 2.2; this technology offers the high-density storage device for quite some time, but when designers need to increase the storage capacity further, they observed "super paramagnetic" issue. So, this is the reason to move to perpendicular recording.

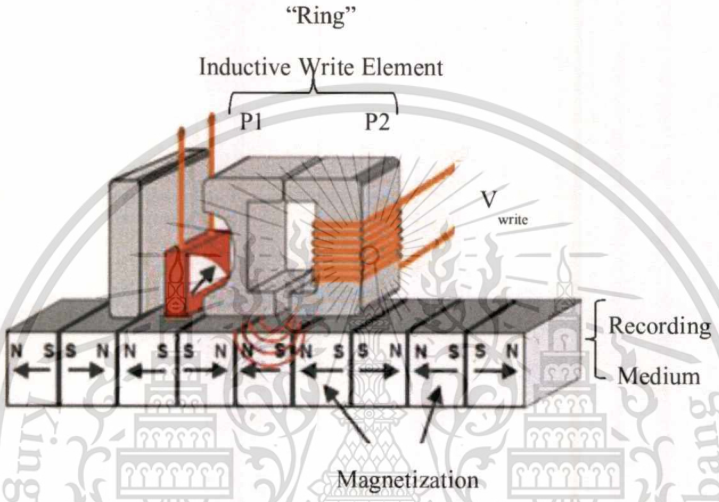


Figure 2.2 The illustration of longitudinal recording

2.2.2 Perpendicular recording

In perpendicular recording, the magnetizations of bits are aligned perpendicularly to the recording medium as presented in Fig. 2.3.

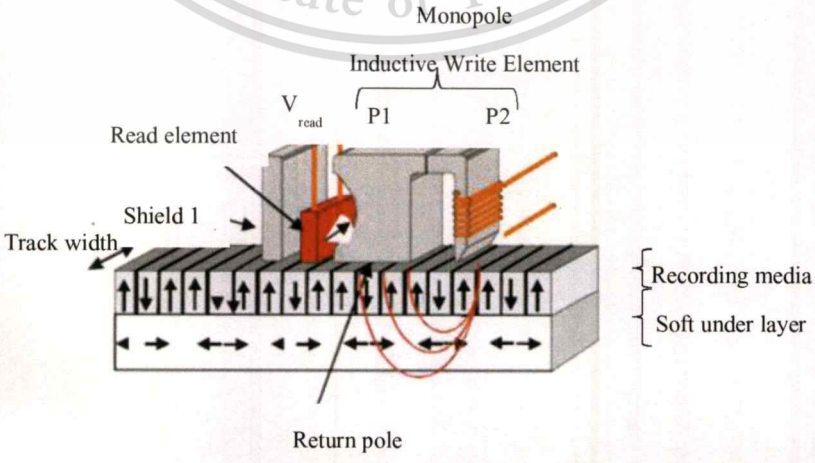


Figure 2.3 The illustration of perpendicular recording

Perpendicular recording provides several advantages over longitudinal recording at such a higher areal density because the effect of soft under layer, writing and reading process are independent on the medium thickness, then small grain diameter is achieved and maintain thermal stability. Another advantage is that the read-back signal on perpendicular recording is larger than longitudinal recoding and the higher writing field can be used to decrease transition jitter noise and DC particulate noise.

2.3 Writing process on magnetic recording

In the writing process, the massage bits are sent to the encoder for encoding. Then the encoded bits or code words pass through the modulator which convert bit stream into rectangular current waveform (see the Fig. 1.1) before being applied to the write head. Afterwards, the write head transforms the rectangular current waveform into the magnetic field to adjust the magnetization of magnetic medium. Finally, the magnetization on magnetic medium are aligned on the required direction. Typically, the applied magnetic field or the writing field is greater than demagnetization field on the medium to ensure the data are stored and the stored data is binary system corresponding to 0s and 1s bit.

The rectangular current waveform which is provided to the write head to convert into the magnetic field or writing field can be categorized into two groups.

2.3.1 Non return to zero (NRZ)

The binary data represent the amplitude of the rectangular waveform. The maximum amplitude corresponds to bit "1s" and the minimum amplitude corresponds to bit "0s".

2.3.2 Non return to zero interleave (NRZI)

The binary data represent the amplitude transition of the rectangular wave form. The transition of amplitude corresponds to bit "1s" and the non-transition of amplitude corresponds to bit "0s".

The rectangular wave form, NRZ and NRZI are presented on the Fig. 2.4, the characterization of wave form and the channel bit on NRZ is similar to NRZI. It is only different on the method to translate the bit stream.

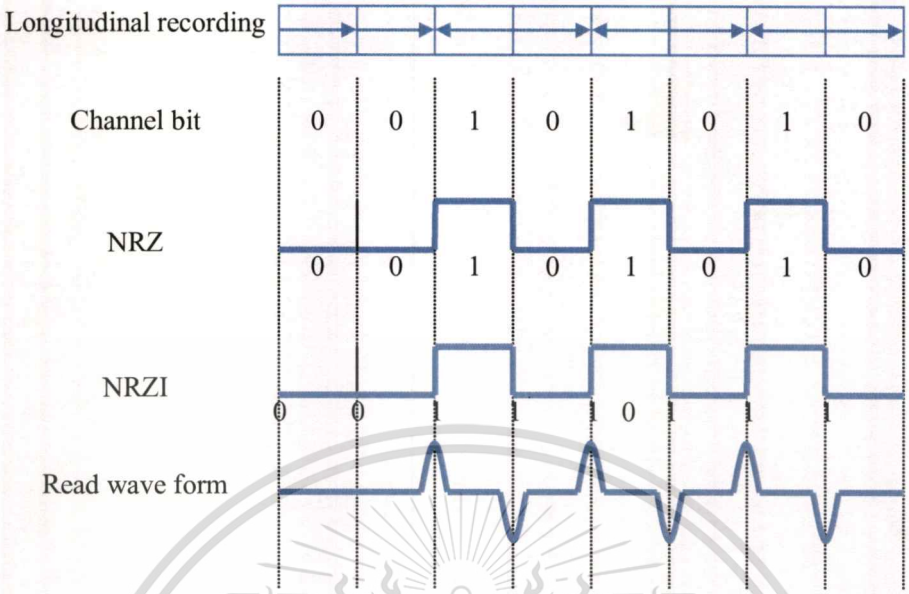


Figure 2.4 The characteristics of NRZ, NRZI and read waveform

A simple read waveform can be obtained based on the linear superposition of isolated transition responses. Fig. 2.5 shows the read waveform corresponding to the NRZ and NRZI on longitudinal recording. Letting a_k denote the input data of NRZ system and b_k denote the input data of NRZI system, the relation of NRZ and NRZI can be expressed as

$$a_k = \left(\frac{1}{1 \oplus D}\right) b_k \quad (2.1)$$

or

$$b_k = (1 \oplus D) a_k \quad (2.2)$$

where D is the delay operator and $\oplus$ is the XOR operator

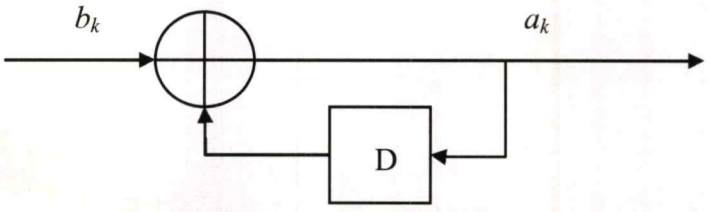


Figure 2.5 The relationship between NRZ and NRZI

2.4 Reading process on magnetic recording

During the reading process, the read head produces an output only when there is a magnetization flux change on the magnetic medium, the system is similar to a differential system. The read out process can be characterized by the transition response $g(t)$ which corresponds to a channel input of +1, that is, the positive transition from 0s to 1s and $-g(t)$ which corresponds to a channel input of -1, that is, the negative transition from 1s to 0s. For longitudinal recording, the transition response expressed as the Lorenzian pulse:

$$g(t) = \frac{1}{1 + \left(\frac{2t}{PW_{50}} \right)^2} \tag{2.3}$$

where PW_{50} is the width of $g(t)$ at half of its peak value. For perpendicular recording, the transition response $g(t)$ can be modeled as

$$g(t) = erf\left(\frac{2t\sqrt{\ln 2}}{PW_{50}} \right) \tag{2.4}$$

where *erf(x)* is the error function defined as

$$erf(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-x^2} dx \tag{2.5}$$

The recording linear density or normalized density is defined as

$$ND = \frac{PW_{50}}{T} \tag{2.6}$$

where T is the bit interval or bit cell. In practice, ND indicates the amount bits are recorded in the bit interval. Fig. 2.6 and 2.7 present the transition responses of longitudinal and perpendicular channel, respectively. The overlapping between the outputs from transitions increases the normalized density.

In contrast, the intersymbol interference (ISI) also increases. The ISI effect can be considered from the dibit response. The dibit response is the channel response of a positive transition followed immediately by a negative transition. Fig. 2.8 and 2.9 present the dibit responses for longitudinal and perpendicular recording channel respectively.

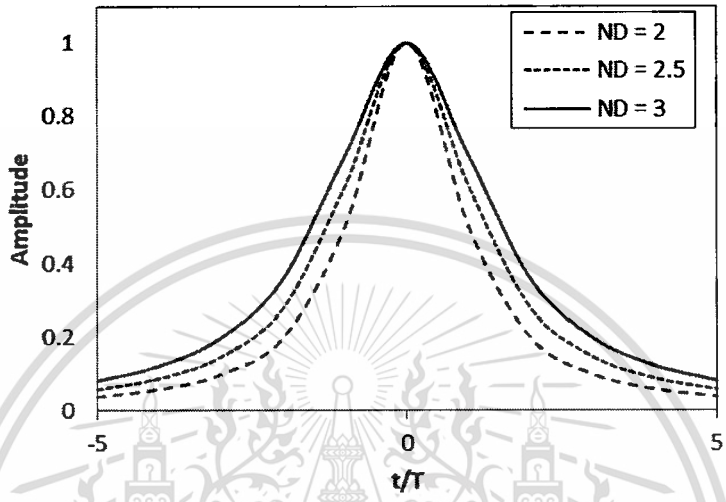


Figure 2.6 The transition response of a longitudinal channel

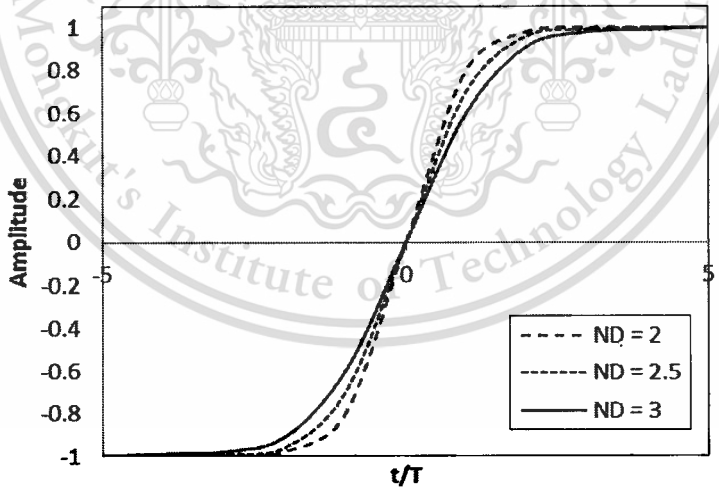


Figure 2.7 The transition response of a perpendicular channel

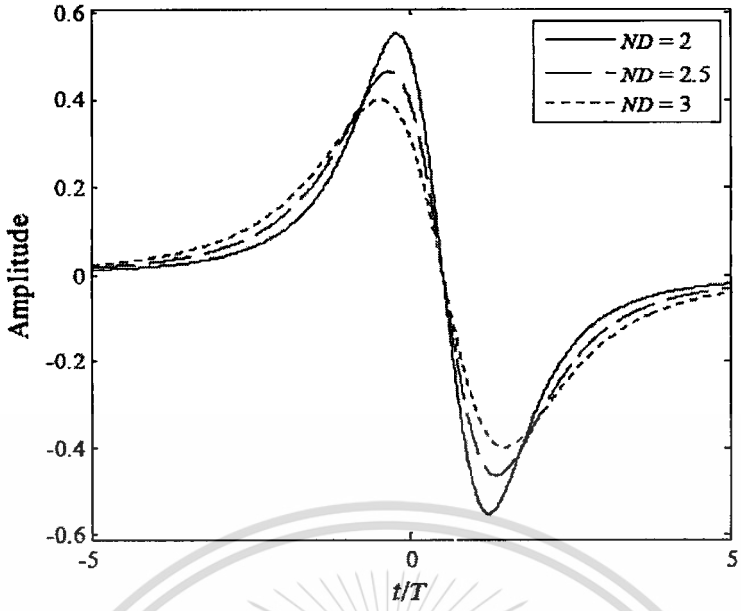


Figure 2.8 The dibit responses for longitudinal recording channel

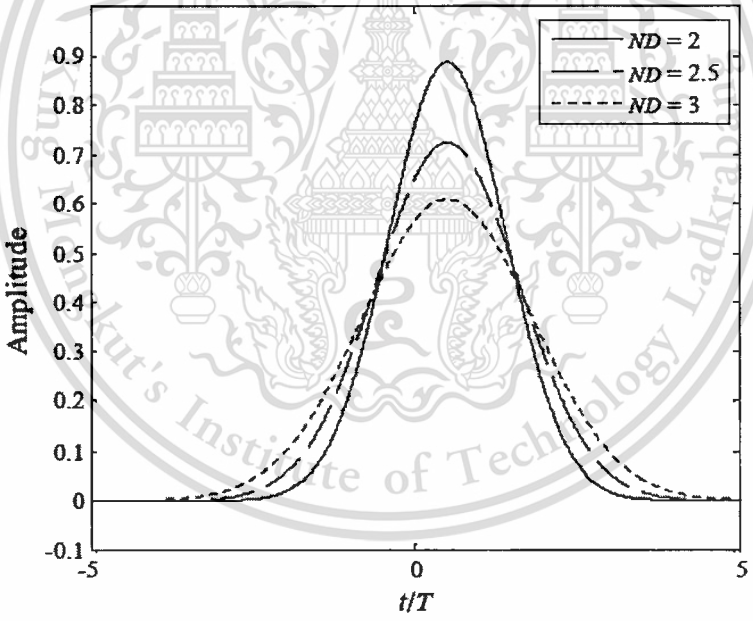


Figure 2.9 The dibit responses for perpendicular recording channel

The dibit pulse or dibit response of system can be expressed as

$$m(t) = g(t) - g(t-T) \tag{2.7}$$

The frequency response of dibit pulse is presented by the Fourier transformation on dibit pulse, in longitudinal channel can be expressed as

$$M(fT) = \exp(-\pi|fT|ND)\cdot(1-\exp(-j2\pi fT)) \tag{2.8}$$

The frequency response in perpendicular channel can be expressed as

$$M(fT) = \frac{T}{j\pi fT}\cdot\exp\left\{-\frac{\pi^2(fT)^2 ND^2}{\ln(16)}\right\}\cdot(1-\exp(-j2\pi fT)) \tag{2.9}$$

where fT is normalized frequency, f is frequency (Hz) and j is imaginary number. Fig. 2.10 and 2.11 present the frequency of dibit response on the longitudinal and perpendicular channels, respectively. When normalized density increases, the frequency dibit response approaches the low-frequency region. At the zero frequency or DC, the frequency dibit responses of longitudinal channel shows zero normalized magnitude, that is, longitudinal channel has no DC offset, but the perpendicular channel shows normalized magnitude equal to 1, that is, perpendicular channel has a DC offset.

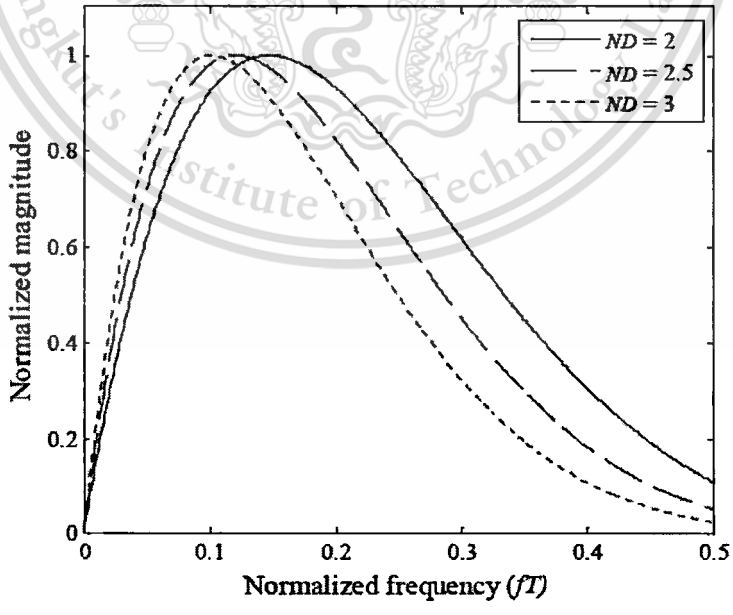


Figure 2.10 The frequency of dibit responses in a longitudinal channel

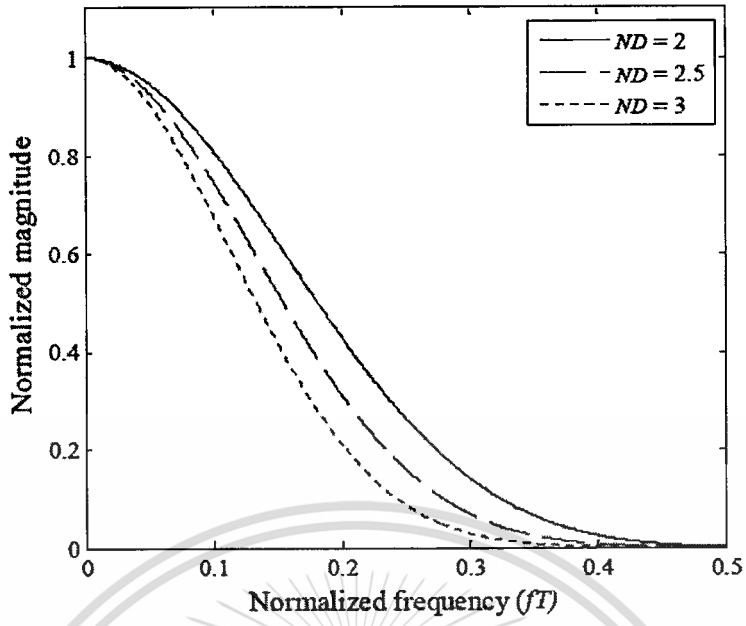


Figure 2.11 The frequency of dibit responses in a perpendicular channel

2.5 Magnetic recording channel model

Typically, the magnetic recording channel model divides to two categories; there are realistic channel model and ideal channel model.

2.5.1 Realistic channel model

Fig. 2.12 illustrates the realistic channel model, the binary input $a_k \in \{1, 0\}$ with bit interval T send to an ideal differentiator 1-D, where D is a delay operator. Then, binary input transforms to transition input $b_k \in \{-1, 0, 1\}$, where $b_k = 1$ represent the positive transition and negative transition, $b_k = 0$ represent no transition. The transition input (b_k) passes through the channel communication with transition response $g(t)$ and interrupted by white Gaussian noise $n(t)$. The read back $p(t)$ is filtered by low pass frequency (LPF) to eliminate the out-of-band noise then sampling by synchronize period with timing recovery. So, the sampled signal is transferred to the equalizer and the symbol detector, respectively, to determine the most likely input sequence $\hat{a}_k$.

The current symbol detector in magnetic recording system is Viterbi detector. However, the complexity of Viterbi detector is significantly increased by the amount of channel memory thus equalizer is necessary to adjust the signal to achieve requirement or known as target response. Moreover, it also decreases the complexity of Viterbi detector.

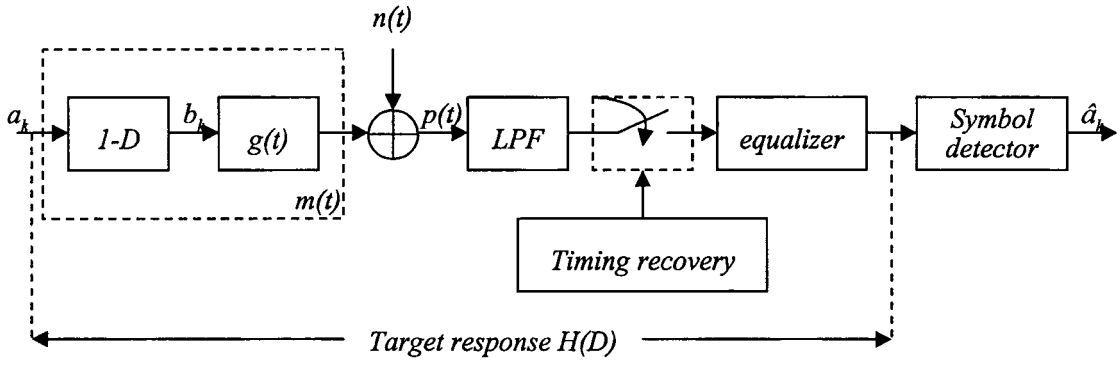


Figure 2.12 Realistic channel models

2.5.2 Ideal channel model

In an ideal channel model, the equalizer supposes to be the perfect equalization. The ideal channel model presented on Fig. 2.13, the binary input $a_k \in \{1, 0\}$ with bit interval T modulated by ideal Nyquist pulse $q(t) = sin(\pi t/T)/(\pi t/T)$ and interrupted by a white Gaussian noise $n(t)$. The read back $p(t)$ is filtered by low-pass frequency (LPF) to eliminate the out-of-band noise then sampling by synchronize period with timing recovery.

The target response on an ideal channel model is the partial-response target. For longitudinal recording can be expressed in (2.10) and for perpendicular recoding, it can be expressed in (2.11), i.e.,

$$H(D) = (1 - D)(1 + D)^n \tag{2.10}$$

$$H(D) = (1 + D)^n \tag{2.11}$$

where n is an integer number. Based on the above equation, the partial-response targets for perpendicular channels have no $(1\text{-}D)$ term. That means it composes a DC offset on the signal. Table 2.1 shows the common partial response target for longitudinal and perpendicular recording.

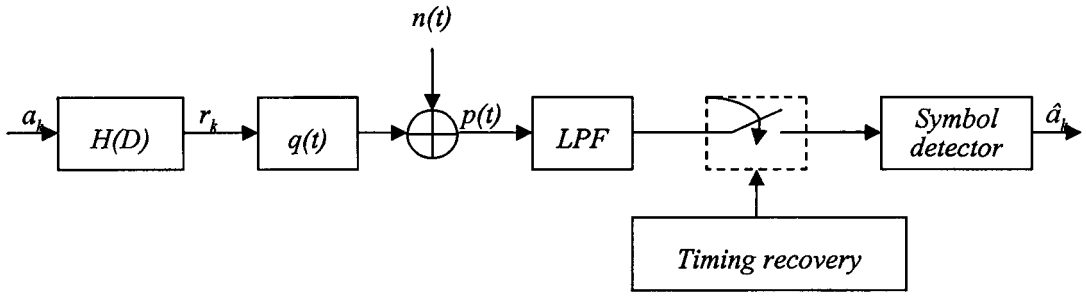


Figure 2.13 Ideal channel models

Table 2.1 Partial response targets on channel models

Recording channel	Partial responses target		
	n = 1	n = 2	n = 3
Longitudinal	PR4 [1 0 -1] $1 - D^2$	EPR4 [1 1 -1 -1] $1 + D - D - D^2$	EEPR4 [1 2 0 -2 -1] $1 + 2D - D^3 - D^4$
Perpendicular	PR1 [1 1] 1 + D	PR2 [1 2 1] 1 + 2D +D2	EPR2 [1 3 3 1] $1 + 3D + 3D^2 + D^3$

CHAPTER 3

LINEAR BLOCK CODES

Channel coding or error correction code is mostly used in data transmission or storage systems to recover the erroneous data. Typically, there are two types of error correction codes; convolution codes and block codes. However, in this study we focus on the block codes related to LDPC codes. The block code encodes the fixed number of bits (k bit), so the information string is divided into blocks of k bits before encoding. The block codes are divided to two groups; linear block codes and non-linear block codes. We suppose that information bits or symbols are binary system, which is 1s and 0s. The set $\{1, 0\}$ has a field structure denoted as the Galois field *GF(2)*, the Galois field of order 2.

The encoding procedure transforms k information bits into n information bits. Thus, we have 1-to-1 function f which represents the encoding procedure can be expressed as

$$f : GF(2)^k \rightarrow GF(2)^n \tag{3.1}$$

The set 2^k encode vectors of length n is called a code of length n and dimension k or an $[n, k]$ codes. The code word is the vector of length n and the code rate is the ratio k/n.

3.1 Hamming distance

The error correcting power of codes is denoted by the minimum Hamming distance of the code. The Hamming distance is the number of different notation in two given vectors with the same code length. The minimum Hamming distance is defined as

$$d = \min\{d_H(\mathbf{a},\mathbf{b}) : \mathbf{a} \neq \mathbf{b}, \mathbf{a}, \mathbf{b} \in \mathbf{C}\} \tag{3.2}$$

where d_H is the Hamming distance, **a, b** are the vectors of length n which is the member of code word **C**

The code word **C** is represented by an [*n, k, d*] codes. The axioms of the Hamming distance are

$$d_H(\mathbf{a},\mathbf{b}) = d_H(\mathbf{b},\mathbf{a}) \tag{3.3}$$
$$d_H(\mathbf{a},\mathbf{b}) = 0 \Leftrightarrow \mathbf{a} = \mathbf{b}$$
$$d_H(\mathbf{a},\mathbf{c}) \le d_H(\mathbf{a},\mathbf{b}) + d_H(\mathbf{b},\mathbf{c})$$

The maximum number of errors that an [*n, k, d*] coed can correct is (*d*-1)/2. Assume that the vector **a**, which is the member of code word **C**, was transmitted then the vector **r** was received and we assume that no more than (*d*-1)/2 errors have occurred. The decoder looks for the code word which has minimum Hamming distance to determine and correct the errors on the vector **r**. In contrast, the vector **r** contains the errors more than (*d*-1)/2 bits, the decoder only determines the error but cannot correct them. Consider the following 1-1 relationship between $GF(2)^2$ and $GF(2)^5$ defining the encoding:

$$\begin{aligned} 00 &\leftrightarrow 00000 \\ 10 &\leftrightarrow 00111 \\ 01 &\leftrightarrow 11100 \\ 11 &\leftrightarrow 11011 \end{aligned} \tag{3.4}$$

The four vectors in $GF(2)^5$ constitute a [5, 2, 3] code **C**. Then vector **r** = 10100 are received, the decoder is looking for these four vectors and finding that **r** belongs to 11100, the final output is the information block 01. Instead of correction of errors, the decoder can only determine the error on the code word. For example, 0100 is encoded as 01001 if an error occurs, these errors will be detected since the modulo sum of the received bits will be 1. But two errors will be undetected.

Another application of error correction code is erasure correction. An erased bit is a bit that cannot be read. So, the decoder has to determine this bit. Normally, the erasure bit is easy to correct because we already know the erasure bit location.

3.2 Linear block codes

Base on the previous topic, the binary codes of length *n* is the subset of $GF(2)^n$ thus, the linear block codes is also a subset of $GF(2)^n$. The axioms of linear block codes are

$$\begin{aligned} &\bar{\mathbf{0}} \in \mathbf{C} \\ &\forall \mathbf{a}, \mathbf{b} \in \mathbf{C}, \mathbf{a} \oplus \mathbf{b} \in \mathbf{C} \end{aligned} \tag{3.5}$$

The $\bar{\mathbf{0}}$ denotes the zero vectors. The linear codes are easier to encode and decode than non-linear codes and they are suitable for implementation in applications. The minimum distance of a linear code is represented by the minimum weight. The weight of a vector **u** is the distance between **u** and the zero vector or the weight of **u** denoted $w_H(\mathbf{u})$ which is the number of nonzero coordinates of the vector **u**.

There are two important matrices that define a linear block codes; generator matrix and parity check matrix. Since the code **C** is a subspace, the dimension k of **C** is the cardinality of a basis of **C**. Consider an $[n, k, d]$ codes, a $k \times n$ matrix **G** is a generator matrix of a code **C** if the row of **G** are a basis of **C**. The encoding process is simple given a generator matrix, let **u** be an information vector of length k and **G** is a $k \times n$ generator matrix, then **u** is encoded to vector **v** of length n by

$$\mathbf{v} = \mathbf{uG} \tag{3.6}$$

The block codes might have many generator matrices, but the encoding depends on the particular matrix chosen. **G** is the systematic generator matrix if **G** can be written as

$$\mathbf{G} = (\mathbf{I}_k \mid \mathbf{V}) \tag{3.7}$$

where $\mathbf{I}_k$ is the $k \times k$ matrix and **V** is the $k \times (n\text{-}k)$ matrix. When an information vector **u** of length k is encoded following (3.6), then the systematic generator matrix outputs a code word **(u|w)**, where **w** has length $n\text{-}k$. That means the systematic generator matrix adds the $n\text{-}k$ redundant bits to the k information bits. Thus, the redundant bits are separate from the information bits. This is easy for decoding because it is simple to determine the redundant bits after decoding. For this reason, the majority linear block codes use the systematic generator matrix.

The second matrix related to the linear block codes is parity-check matrix **H**. The parity matrix **H** has a size of $(n\text{-}k) \times n$. Generally, we call a parity-check matrix an $[n, k]$ code **C**. Let **v** is the vector and $\mathbf{v} \in \mathbf{C}$. The relation of parity-check matrix and **v** can expressed as

$$\mathbf{v}\mathbf{H}^{\mathbf{T}} = \mathbf{0} \tag{3.8}$$

where $\mathbf{H}^{\mathbf{T}}$ is the transpose of matrix and **0** is a zero vector of length $n\text{-}k$. The parity-check matrix is the systematic matrix if **H** can be written as

$$\mathbf{H} = \left(\mathbf{W} \mid \mathbf{I}_{\mathbf{n-k}}\right) \tag{3.9}$$

Where $\mathbf{I}_{\mathbf{n-k}}$ is the $(n\text{-}k) \times (n\text{-}k)$ identity matrix and **W** is the $(n\text{-}k) \times k$ matrix. The systematic parity check matrix **H** is easy to find by a systematic generator matrix **G** of a code **C**. It can be expressed as

$$\mathbf{G} \bullet \mathbf{H}^{\mathbf{T}} = \mathbf{0} \tag{3.10}$$

and

$$\mathbf{H} = (\mathbf{V}^{\mathbf{T}} \mid \mathbf{I}_{\mathbf{n-k}}) \tag{3.11}$$

3.3 Syndrome

Let **C** is an $[n, k, d]$ codes with parity-check matrix **H**. Let **u** be the transmitted vector and **r** is the received vector which consists the error pattern **e**.

$$\mathbf{r} = \mathbf{u} \oplus \mathbf{e} \tag{3.12}$$

The syndrome of vector **r** is represented by a vector **s** of length $n\text{-}k$ given by

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}} \tag{3.13}$$

If an error occurs, the syndrome of $\mathbf{r}$ is the zero vectors. Typically, the syndrome determines the error pattern on the information code. The proof of syndrome given by

$$\begin{aligned} \mathbf{s} &= \mathbf{r}\mathbf{H}^{\mathrm{T}} = (\mathbf{u} \oplus \mathbf{e})\mathbf{H}^{\mathrm{T}} \\ \mathbf{s} &= \mathbf{u}\mathbf{H}^{\mathrm{T}} \oplus \mathbf{e}\mathbf{H}^{\mathrm{T}} \\ \mathbf{s} &= \mathbf{e}\mathbf{H}^{\mathrm{T}} \end{aligned} \tag{3.14}$$

Based on (3.14), the syndrome does not depend on the received vector but on the error vector. Every error vector of weight $\leq (d\text{-}1)/2$ corresponds a unique syndrome. For example, let $\mathbf{e}_1$ and $\mathbf{e}_2$ be two error vectors with syndrome $\mathbf{s}_1 = \mathbf{e}_1\mathbf{H}^{\mathrm{T}}$ and $\mathbf{s}_2 = \mathbf{e}_2\mathbf{H}^{\mathrm{T}}$, respectively. If $\mathbf{s}_1 = \mathbf{s}_2$ then

$$\begin{aligned} \mathbf{s} &= (\mathbf{e}_1 \oplus \mathbf{e}_2)\mathbf{H}^{\mathrm{T}} \\ \mathbf{s} &= \mathbf{e}_1\mathbf{H}^{\mathrm{T}} \oplus \mathbf{e}_2\mathbf{H}^{\mathrm{T}} \\ \mathbf{s} &= \mathbf{s}_1 \oplus \mathbf{s}_2 \\ \mathbf{s} &= \mathbf{0} \end{aligned} \tag{3.15}$$

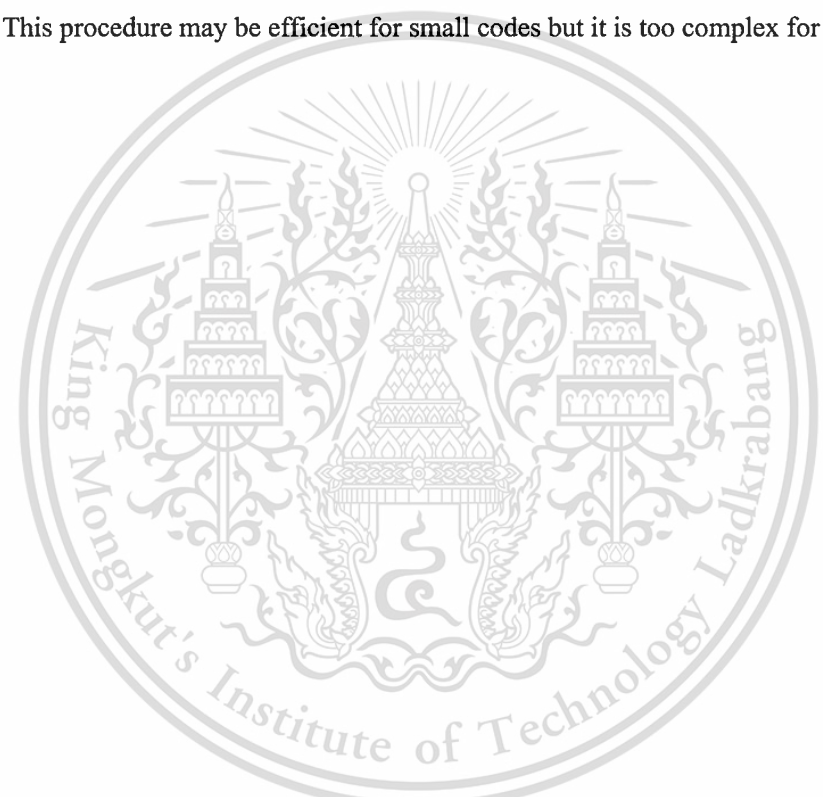
The advantage of syndrome is the corresponding with error vector. We can construct a table with the 1-1 correspondence between syndrome and error vector, during decoding we can look for the table to determine the error pattern. The procedure to apply syndrome on decoding is to determine the syndrome of the received vector then look for the error pattern on the table, finally we add the error pattern on the original received vector.

Consider a [5, 2, 3] code whose parity matrix $\mathbf{H}$ below. We have 6 error patterns of weight 1. And the correspondence between error vector and syndrome is as below.

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

$$
\begin{aligned}
00000 &\leftrightarrow 000 \\
10000 &\leftrightarrow 011 \\
01000 &\leftrightarrow 110 \\
00100 &\leftrightarrow 100 \\
00010 &\leftrightarrow 010 \\
00001 &\leftrightarrow 001
\end{aligned}
\tag{3.17}
$$

Assume that the received vector $\mathbf{r}$ = 10111. The syndrome of the received vector $\mathbf{r}$ is determined by $\mathbf{s} = \mathbf{eH}^T$ and $\mathbf{s}$ = 100. Then, we look at the table and the syndrome corresponds to error pattern 00100. After that, we add the error pattern to the original received vector $\mathbf{r}$ and we get the transitioned vector 10011. This procedure may be efficient for small codes but it is too complex for large codes.



CHAPTER 4

LOW-DENSITY PARITY-CHECK CODES (LDPC)

Low-density parity-check (LDPC) code is a type of linear block codes which approach near capacity performance (Shannon's limit) on a large collection of data transmission and storage channels simultaneously admitting implementable decoders. LDPC codes were invented by Robert Gallager [1] in the 1960 doctoral dissertation and forgotten for 35 years. In the 1990's, LDPC codes were rediscovered by MacKay and Neal [2] who, independently investigated the work of Gallager. The one important work is Tanner graph. The advantage of linear block codes is possessing sparse parity check matrix. Here, we consider only LDPC with binary system. Let **C** be the LDPC codes over $GF(2)$, **G** is the $(k \times n)$ generator matrix and **u** is the information vector. Then the encoded bit is written as

$$\begin{aligned} \mathbf{c} &= \mathbf{uG} \\ \mathbf{c} &= [u_0 + u_1 + u_2 + \ldots + u_{k-1}]\mathbf{G} \\ \mathbf{c} &= u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + u_2\mathbf{g}_3 + \ldots + u_{k-1}\mathbf{g}_{k-1} \end{aligned} \tag{4.1}$$

where $\mathbf{g}_i$ is the row vectors in generator matrix. The $(n\text{-}k)$ dimensional null space $\mathbf{C}^{\perp}$ of **G** and all **x** vectors which $\mathbf{x} \in \mathbf{C}$, $\mathbf{xG}^{\mathrm{T}} = 0$ and is explained by

$$\mathbf{H} = \{h_0, h_1, \ldots, h_{n-k-1}\} \tag{4.2}$$

H is called $(n\text{-}k) \times n$ parity-check matrix and $\mathbf{cH}^{\mathrm{T}} = 0$. The parity check matrix **H** is named because it performs $m = n\text{-}k$ separate parity checks on a received word.

A LDPC codes is a linear block code which the parity check matrix **H** has a low density of 1s. The regular LDPC code is a linear block code which parity checks matrix **H** contains constant column weight and constant low weight. In contrast, column weight and row weight are not constant number, it is irregular LDPC codes. The LDPC code rate equals to k/n.

4.1 Tanner graph

Tanner considers LDPC codes and showed how they may be represented by a bipartite graph, now called a Tanner graph. The Tanner graph of an LDPC code is analogous to the trellis of a convolutional code in that it provides a complete description of the decoding algorithm. The bipartite graph contains two types of nodes and edge connects only nodes which difference type. The nodes on Tanner graph are variable nodes and check nodes respectively. The Tanner graph is written following this method; it is the check node j connects to variable nodes i whenever h_{ji} in **H** is 1s. The variable nodes correspond to n columns on **H** matrix and the check nodes correspond to m rows on **H** matrix. The example relationship between a Tanner graph and a parity-check matrix can be expressed as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{4.3}$$

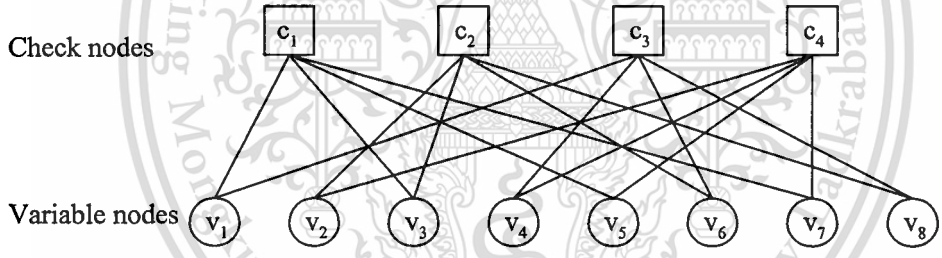


Figure 4.1 Tanner graph

In figure 4.1, the variable nodes v_1, v_3, v_5, v_7 connect to check node c_1 and represent the element of parity check matrix h_{11}, h_{13}, h_{15}, h_{17} which are equal to 1s. The sample Tanner graph is the regular graph because the column weight (w_c) is constant which equal to 2 and the row weight (w_r) is constant which equal to 4. For irregular graph, the column weight and row weight is a polynomial functions of column and row number called "degree distribution polynomial". The function of row weight $\lambda(x)$ can be expressed as

$$\lambda(x) = \sum_{d=1}^{d_{v\max}} \lambda_d x^{d-1} \tag{4.4}$$

where d_{vmax} is the maximum variable node degree and λ_d is the fraction of all edges connected to variable node which degree equal to *d*. The function of column weight $\rho(x)$ can be express as;

$$\rho(x) = \sum_{d=1}^{d_{c\max}} \rho_d x^{d-1} \tag{4.5}$$

where d_{cmax} is the maximum check node degree and ρ_d is the fraction of all edges connected to check node which degree equal to *d*.

A cycle or loop in Tanner graph is a part which contains the edges of the variable node with close back to itself. In Fig. 4.1, it is the Tanner graph with a cycle of length 4. Another important parameter on the Tanner graph is "girth"; the girth is the minimum cycle length of the graph. The shortest possible girth in a Tanner graph is girth-4. The particularly short cycles on Tanner graph degrade the performance of the iterative decoding for LDPC codes. That is, a short girth leads to a low decoding performance, while a large girth leads to a high decoding performance.

4.2 LDPC code design approaches

4.2.1 Gallager codes

The original LDPC codes [1] due to Gallager are regular codes with H matrix of the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_{wc} \end{bmatrix} \tag{4.6}$$

For any u and w_r are the integers and greater than 1, each sub matrix $\mathbf{H}_d$ is $u \times uw_r$ with row weight w_r and column weight 1. The sub matrix $\mathbf{H}_1$ has following specific form; for $i=1, 2, 3, \ldots, u$ the i^{th} row contains all of its w_r is in columns $(i\text{-}1)w_r\text{-}1$ to iw_r. The other sub matrices are simply column

permutations of $\mathbf{H}_1$ and $\mathbf{H}$ is regular with dimension $uw_c \times uw_r$. Base on this concept, there are not guarantee to eliminate cycle-4. The effective distance property are $w_c \geq 3$ and $w_r > w_c$.

4.2.2 MacKay codes

MacKay had independently discovered the benefits of the parity check matrix **H**. He provided the decoding performance near Shannon's limit with high number of LDPC codes [2]. The applications of his codes are storage devices and data communication. He introduced the semi-random structure on parity check matrix with regular codes. The concepts of MacKay codes are following;

H is created by randomly generating column weight with uniform row weight and no two columns having overlapped greater than one. Moreover short cycle avoiding is implementing and plus $\mathbf{H} = [\mathbf{H}_1\ \mathbf{H}_2]$ is constrained.

The issue for Mackay codes is high complexity encoding. The encoding is performed by generate **H** in form $[\mathbf{P}^T\ \mathbf{I}]$ via Gaussian elimination from generator matrix $\mathbf{G} = [\mathbf{I}\ \mathbf{P}]$ and **P** is not the sparse matrix so, encoding will take more complexity.

4.2.3 Array codes

The one class of LDPC codes is array codes. The array codes were introduced by J. Fan in 2000 [9]. This code decodes with a message passing algorithm which eliminate cycle length 4. The parity check matrixes **H** for array codes can be form follow this format;

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \mathbf{I} & \boldsymbol{\alpha} & \boldsymbol{\alpha}^2 & \cdots & \boldsymbol{\alpha}^{t-1} \\ \mathbf{I} & \boldsymbol{\alpha}^2 & \boldsymbol{\alpha}^4 & \cdots & \boldsymbol{\alpha}^{2(t-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \boldsymbol{\alpha}^{c-1} & \boldsymbol{\alpha}^{2(c-1)} & \cdots & \boldsymbol{\alpha}^{(c-1)(t-1)} \end{bmatrix} \tag{4.7}$$

Where k and j are two integers and $k, j \leq p$ (p denotes a prime number). **I** is the $p \times p$ identity matrix, **0** is the $p \times p$ null matrix and $\boldsymbol{\alpha}$ is the p × p permutation matrix representing a single cyclic shift from **I**. The example of permutation matrix is as;

$$\mathbf{I}=\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix} \qquad \boldsymbol{\alpha}=\begin{bmatrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\1&0&0&0&0\end{bmatrix}$$

$$\boldsymbol{\alpha}^2=\begin{bmatrix}0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\1&0&0&0&0\\0&1&0&0&0\end{bmatrix} \qquad \boldsymbol{\alpha}^3=\begin{bmatrix}0&0&0&1&0\\0&0&0&0&1\\1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\end{bmatrix}$$

$$\boldsymbol{\alpha}^4=\begin{bmatrix}0&0&0&0&1\\1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{bmatrix} \qquad \boldsymbol{\alpha}^5=\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$$

When the permutation matrix shifts with the number that equal to the dimension of identity matrix, the permutation matrix is the identity matrix.

4.2.4 Modify array codes

The modify array codes was provided in 2002 [10] by J. Fan. This code maintains the message passing algorithm on decoding and avoiding cycle of length 4. Moreover parity check matrices are in diagonal matrix form. The upper triangular of H guarantees encoding linear in the code word length. The parity check matrix of modify array codes are as

$$\mathbf{H}=\begin{bmatrix}\mathbf{I}&\mathbf{I}&\mathbf{I}&\cdots&\mathbf{I}&\mathbf{I}&\cdots&\mathbf{I}\\\mathbf{0}&\mathbf{I}&\boldsymbol{\alpha}&\cdots&\boldsymbol{\alpha}^{j-2}&\boldsymbol{\alpha}^{j-1}&\cdots&\boldsymbol{\alpha}^{k-2}\\\mathbf{0}&\mathbf{0}&\mathbf{I}&\cdots&\boldsymbol{\alpha}^{2(j-3)}&\boldsymbol{\alpha}^{2(j-2)}&\cdots&\boldsymbol{\alpha}^{2(k-3)}\\\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\cdots&\vdots\\\mathbf{0}&\mathbf{0}&\cdots&\mathbf{0}&\mathbf{I}&\boldsymbol{\alpha}^{j-1}&\cdots&\boldsymbol{\alpha}^{(j-1)(k-j)}\end{bmatrix} \tag{4.8}$$

4.2.5 Progressive edge growth algorithm (PEG)

In 2001, X. Y. Hu proposed a method to construct the Tanner graphs having a large girth by establishing edges or connections between symbol and check nodes in an edge-by-edge manner, called progressive edge-growth (PEG) algorithm [4]. The minimum distance of low-density parity-check (LDPC) codes are not available. Simple variations of the PEG algorithm can also be applied to generate linear-time encodable LDPC codes. Let n denotes the number of symbol nodes and m denotes the number of check nodes and d_v denotes the symbol–node–degree sequence. The PEG algorithm is started on the edge-selection and then places a new edge on the graph has as small an impact on the girth as possible. The underlying graph grows in an edge-by-edge manner, optimizing each local girth. The fundamental idea is to find the most distant check node and then to place a new edge connecting the symbol node and this most distant check node. The procedure of PEG algorithm illustrate as the figure 4.2.

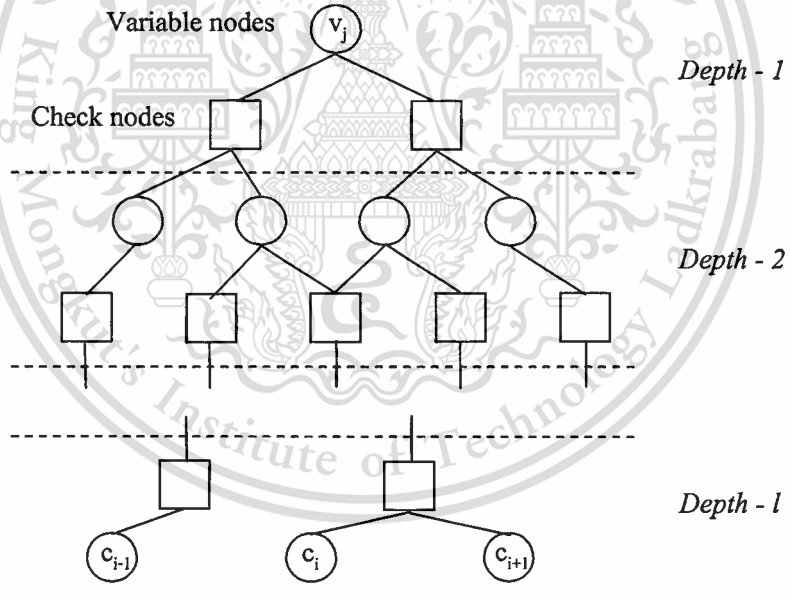


Figure 4.2 Tree diagram expanding from the symbol node with a depth-*l*.

The summary of PEG algorithm is as follows.

for $j = 0$ to $n - 1$ do

for $k = 0$ to $d_v - 1$ do

if $k = 0$

Select the check node which smallest row weight to establish an edge.

else

Expand a sub graph up to depth l under the current graph setting such that the $N_{v_j}^{l}$ stops increasing but is smaller than m or $\overline{N}_{v_j}^{l} \neq \varphi$ but $\overline{N}_{v_j}^{l+1} \neq \varphi$. In the first case, not all check nodes are reachable from v_j, so the PEG algorithm chooses the one that is not reachable. This often occurs in the initial phase of graph construction. In the second case, all check nodes are reachable from v_j, and chooses the one that is at the largest distance from v_j, say at depth $l+1$, so that the cycle created by establishing an edge is of the largest possible length $2(l + 2)$

end

end

end

4.2.6 PEG with quasi-cyclic (QC) algorithm

In 2004, L. Zongwang presents the algorithm with modifying the progressive-edge growth (PEG) graphs with a quasi-cyclic constraint [6]. This algorithm admits both regular and irregular LDPC codes and is simpler for implementation than the original PEG algorithm. Based on the original PEG algorithm, the tree is expanded from each variable node and the edges are added to the Tanner graph node by node. Then he divides all the variable nodes and the check nodes into small groups with each group having p nodes. Then he adds the edges to the Tanner graph group by group instead of node by node with circulant size $p \times p$. For each variable node group, he expands trees for the first variable node in the group to find its optimized edges. The edges of other variable nodes in the same group determine the circulant constraint automatically. The design of PEG QC algorithm is as seen in Fig. 4.3.

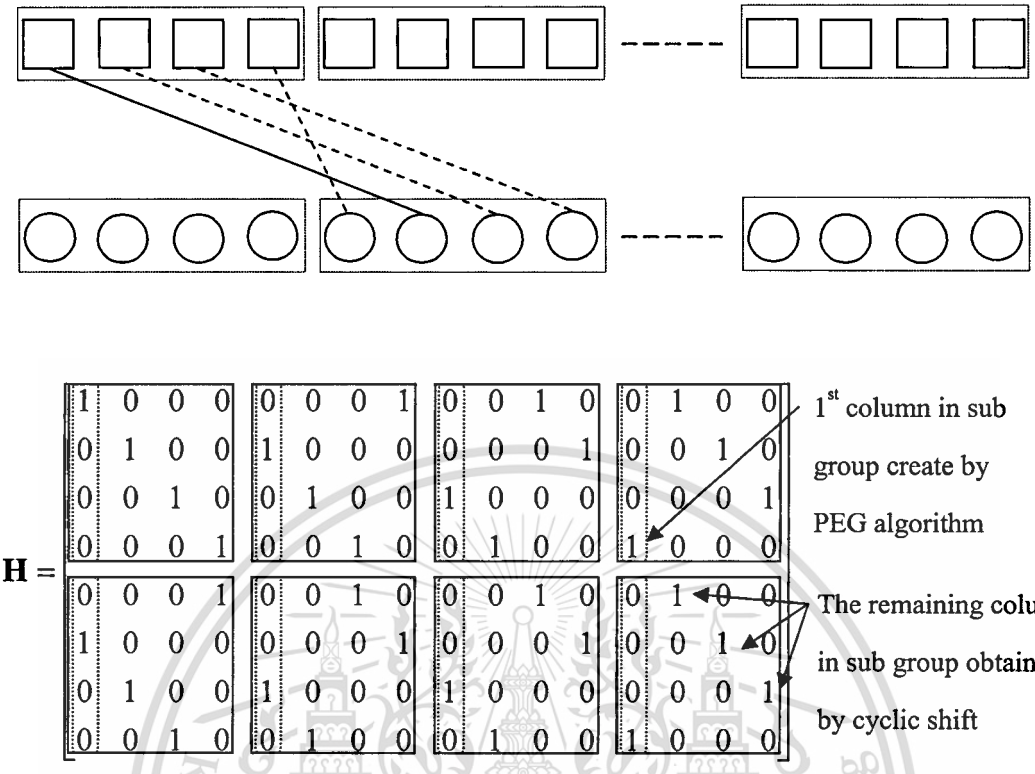


Figure 4.3 The design of PEG-QC algorithm and the sample H matrix

4.2.7 Extended PEG algorithm

In 2009, Z. Zhou and team provide an extend PEG algorithm for constructing LDPC codes with very high rate given the lower bound of girth [8].

Firstly, they construct the Tanner graph and then try to maximize the cardinality of symbol nodes set N by adding symbol node to current graph. In this additional procedure, they always check current graph g_r if $g_i < g$, they stop adding, where g is the lower bound of girth they defined which is an even integer bigger than 4.

Secondly, if $g_i < g_i$-1 they delete the edges incident to v_j and v_{j-1} then redo the connection procedure for these two nodes. Because the check nodes linked to variable modes v_j are always randomly selected, the new connections are different form old ones. This procedure repeats until $g_i = g_i$-1 or stops after *iter* iterations, where *iter* is an integer predefined. In this way, they can slow down the decreasing of current girth g_r

Finally, when choosing a check node from $\overline{N}_{v_j}^{l}$, they just randomly select one from $\overline{N}_{v_j}^{l}$, not from those having the lowest check node degree. This method gives more freedom for selecting check node, especially when g is not very large, this means more chances to change the check nodes of symbol node v_j.

4.3 Iterative decoding algorithms

The decoding algorithms perform iteratively on various variables in Tanner graph based on difference model. Typically there are multiple names for decoding algorithm such the sum product algorithm (SPA), the belief propagation algorithm (BPA) and the message passing algorithm (MPA).

4.3.1 Sum product algorithm (SPA)

Tanner graph is the key for decoding algorithm. The variable and check nodes operate like the processor that computes the inputs and pass through the neighboring nodes. First, we introduce the notation:

y_i is the received information

V_j is the set of variable nodes connected to check node c_j

$V_{j/i}$ is the set of variable nodes connected to check node c_j except variable node v_i

C_i is the set of check nodes connected to variable node v_i

$C_{i/j}$ is the set of check nodes connected to variable node v_i except check node c_j

$M_v(\sim i)$ is the set of message from all variable nodes except node c_i

$M_c(\sim j)$ is the set of message from all check nodes except node v_j

$P_i = \Pr(c_i = 1|y_i)$ is the possibility of check node c_i equal to 1 which received information y_i

S_i is the event that the check equations involving variable node v_i are satisfied.

$q_{ij}(b) = \Pr(v_i = b|S_i, y_i, M_c(\sim j)$, where $b \in \{0, 1\}$, is the massage from variable node v_i to check node c_j

$r_{ji}(b) = \Pr(\text{check equation } c_j \text{ is satisfied}|\ v_i = b, M_v(\sim i))$, where $b \in \{0, 1\}$, is the massage from check node c_j to variable node v_i

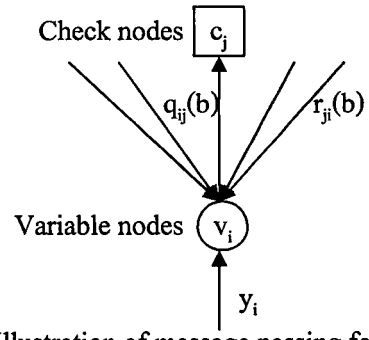


Figure 4.4 Illustration of message passing for computation of $q_{ij}(b)$

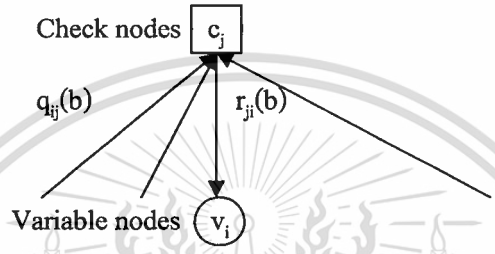


Figure 4.5 Illustration of message passing for computation of $r_{ji}(b)$

Consider the message $q_{ij}(0)$ which form as Fig. 4.4, we can express as:

$$\begin{aligned} q_{ij}(0) &= \Pr(v_i = 0 \mid y_i, S_i, M_c(\sim j)) \\ &= (1-P)\Pr(S_i \mid v_i = 0, y_i, M_c(\sim j))/\Pr(S_i) \\ &= K_{ij}(1-P)\prod_{j' \in C_{i \setminus j}} r_{j'i}(0) \end{aligned} \tag{4.9}$$

We use Baye's rule twice to obtain the second line and the independence assumption to obtain the third line. Then the $q_{ij}(1)$ can be obtained as;

$$q_{ij}(1) = K_{ij} P_i \prod_{j' \in C_{i \setminus j}} r_{j'i}(1) \tag{4.10}$$

Where the constant K_{ij} are chosen to ensure that $q_{ij}(0) + q_{ij}(1) = 1$. For $r_{ji}(b)$, we consider a sequence of M independent binary digit a_i which $\Pr(a_i = 1) = p_i$. Then the probability that $\{a_i\}_{\text{i=1 to M}}$ contains an even number of 1s is;

$$r_{ji}(b) = \frac{1}{2} + \frac{1}{2}\prod_{l=i}^{M}(1-2p_i) \tag{4.11}$$

For the correspondence p_i equivalent to $q_{ij}(1)$, we can rewrite the equation 4.11 as;

$$r_{ji}(0) = \frac{1}{2} + \frac{1}{2}\prod_{i' \in V_{jN}}(1-2q_{i'j}(1)) \tag{4.12}$$

Then the $r_{ji}(1)$ can be obtained as;

$$r_{ji}(1) = 1 - r_{ji}(0) \tag{4.13}$$

In binary system over add white Gaussian's noise system, the possibility function can express as equation 4.14. Let x_i = 1-2c_i is the i^{th} transmitted binary value (x_i = +/-1 when v_i = 0/1). Then the received information is $y_i = x_i + n_i$ where n_i is the Gaussian's noise with normal distribution η(o, σ^2).

$$\Pr(x_i = x \mid y_i) = \frac{1}{1 + e^{-2yx/\sigma^2}} \tag{4.14}$$

Summary of the probability domain SPA decoder

1. For i = 0, 1,…,n-1, set $P_i = \Pr(c_i = 1|y_i)$ then set $q_{ij}(0)$ = 1- P_i for all i, j which h_{ij} = 1
2. Update $r_{ji}(b)$ using equation 4.12 and 4.13
3. Update $q_{ij}(b)$ using equation 4.9 and 4.10
4. For i = 0, 1,…,n-1, computes

$$Q_i(0) = K_i(1-P_i)\prod_{j \in C_i} r_{ji}(0) \tag{4.15}$$

and

$$Q_i(1) = K_i P_i \prod_{j \in C_i} r_{ji}(1) \tag{4.16}$$

where K_i are the constant which ensure $Q_i(0) + Q_i(1) = 1$

5. For $i = 0, 1, \ldots, n-1$, set

$$\hat{c}_i = \begin{cases} 1, & \text{If } Q_i(0) > Q_i(1) \\ 0, & \text{else} \end{cases} \tag{4.17}$$

4.3.2 Logarithmic sum product algorithm

The probability sum product algorithm involves many multiplications of probability and could become numerically unstable. Thus the log-domain of the sum product is proposed. Then we define the Log Likelihood Ratio (LLRs) as;

$$L(c_i) = \log\left(\frac{\Pr(c_i = 0 \mid y_i)}{\Pr(c_i = 1 \mid y_i)}\right) \tag{4.18}$$

$$L(r_{ji}) = \log\left(\frac{r_{ji}(0)}{r_{ji}(1)}\right) \tag{4.19}$$

$$L(q_{ij}) = \log\left(\frac{q_{ij}(0)}{q_{ij}(1)}\right) \tag{4.20}$$

$$L(Q_i) = \log\left(\frac{Q_i(0)}{Q_i(1)}\right) \tag{4.21}$$

In binary system over add white Gaussian's noise system, the Logarithmic sum product function can express as;

$$L(q_{ij}) = L(c_i) = {}^{2y_i}\!/_{\sigma^2} \tag{4.22}$$

First step, we replace $r_{ji}(0)$ with $1 - r_{ji}(1)$ in equation 4.12 and rewrite it. Then we obtain;

$$1 + 2r_{ji}(1) = \prod_{i' \in V_{j \setminus i}} (1 - 2q_{i'j}(1))$$

Based on the principle, $\tanh[(1/2)\log(p_0/p_1)]= p_0-p_1 = 1 - 2p_1$, we can rearrange the equation as;

$$\tanh\left(\frac{1}{2}L(r_{ji})\right)=\prod_{i'\in V_{jN}}\tanh\left(\frac{1}{2}L(q_{i'j})\right) \tag{4.23}$$

This expression consists of the complex tanh function, then we separate $L(q_{ij})$ into its sign and magnitude for simply computes. So, $L(q_{ij})$ can obtain as;

$$\begin{aligned} L(q_{ij}) &= \alpha_{ij}\beta i_j \\ \alpha_{ij} &= sign\left[L(q_{ij})\right] \\ \beta_{ij} &= \left|L(q_{ij})\right| \end{aligned}$$

Then the equation 4.23 can rewrite as;

$$\tanh\left(\frac{1}{2}L(r_{ji})\right)=\prod_{i'\in V_{jN}}\alpha_{i'j}\prod_{i'\in V_{jN}}\tanh\left(\frac{1}{2}\beta_{i'j}\right) \tag{4.24}$$

$$\begin{aligned} L(r_{ji}) &= \prod_{i'}\alpha_{i'j}\cdot 2\tanh^{-1}\left(\prod_{i'}\tanh\left(\frac{1}{2}\beta_{i'j}\right)\right) \\ &= \prod_{i'}\alpha_{i'j}\cdot 2\tanh^{-1}\log^{-1}\log\left(\prod_{i'}\tanh\left(\frac{1}{2}\beta_{i'j}\right)\right) \\ &= \prod_{i'}\alpha_{i'j}\cdot 2\tanh^{-1}\log^{-1}\sum_{i'}\log\left(\prod_{i'}\tanh\left(\frac{1}{2}\beta_{i'j}\right)\right) \\ &= \prod_{i'\in V_{jN}}\alpha_{i'j}\cdot\varphi\left(\sum_{i'\in V_{jN}}\varphi(\beta_{i'j})\right) \end{aligned} \tag{4.25}$$

Where: $\varphi(x)$ define as equation 4.26 and $\varphi^{-1}(x)=\varphi(x)$ when $x > 0$. The function $\varphi(x)$ can express as;

$$\varphi(x)=-\log\left[\tanh\left(\frac{x}{2}\right)\right]=\log\left(\frac{e^x+1}{e^x-1}\right) \tag{4.26}$$

LDPC codes. It constructs the parity-check matrix **H** similar to the PEG-QC-LDPC codes to maintain a lower memory requirement. However, girth maximization method is added to enhance a local girth distance on the Tanner graph. Our proposed algorithm is as follows:

Initial parameters: m, n, d_{s_j} and p

for $j = 0$ to $n-1$ **do**

while new_checknode $\neq$ previous_checknode **do**

for $k = 0$ to $d_{s_j} - 1$ **do**

if $k = 0$

Select a check node c_i which has the lowest check node degree under the current graph and establish the edge to symbol node s_j, then create the circulant sub-matrix with size $p \times p$ based on the incident first edge and establish the edge to s_{j+p-1}.

else

Expand the tree diagram initiated from s_j under the current graph such that $\overline{N_{s_j}^{l}} \neq 0$ but $\overline{N_{s_j}^{l+1}} \neq \varphi$, or the cardinality of $N_{s_j}^{l}$ stops increasing but is smaller than the number of check node, m. Select a check node c_i from $\overline{N_{s_j}^{l}}$ which has the lowest check node degree then establish the edge to symbol node s_j, and create the circulant sub-matrix with size $p \times p$ based on the first edge incident and establish the edge to s_{j+p-1}. If all elements on the first column are zero, we place the null matrix.

end

end

Calculate the girth property of new check nodes and compare with collected check nodes.

if new and collected check nodes are similar **do**

Compare the girth property of collected check nodes and new check nodes and then select the check nodes with the maximum girth property to establish the edges.

else

Collect the edge link with s_{j-p} to s_{j-1} and s_j to s_{j+p-1} then delete these edges on the matrix **H**. Repeat the entire step again.

end

Second step, we simply equation 4.9 and 4.10 then take the logarithm of both sides to obtain;

$$L(q_{ij}) = L(c_i) + \sum_{i' \in C_i \setminus j} L(r_{j'i}) \tag{4.27}$$

Third step, return hard output follow this equation.

$$L(Q_i) = L(c_i) + \sum_{i' \in C_i} L(r_{ji}) \tag{4.28}$$

Summary of the Logarithmic SPA decoder

1. For $i = 0, 1, \ldots, n\text{-}1$, initialize $L(q_{ij})$ according to equation 4.22 for all i, j which $h_{ij} = 1$
2. Update $L(r_{ji})$ using equation 4.25
3. Update $L(q_{ij})$ using equation 4.27
4. Update $L(Q_i)$ using equation 4.28
5. For $i = 0, 1, \ldots, n\text{-}1$, set

$$\hat{c}_i = \begin{cases} 1, & \text{If } L(Q_i) < 0 \\ 0, & \text{else} \end{cases}$$

4.3.3 Min sum product algorithm

Consider on the equation 4.25, the shape of $\varphi(x)$ correspond to the smallest β_{ij} so, the equation 4.25 can rewrite as:

$$\begin{aligned} L(r_{ji}) &= \prod_{i' \in V_j \setminus i} \alpha_{i'j} \cdot \varphi\left(\sum_{i' \in V_j \setminus i} \varphi(\beta_{i'j}) \right) \\ &\cong \prod_{i' \in V_j \setminus i} \alpha_{i'j} \cdot \varphi\left(\varphi\left(\min_{i' \in V_j \setminus i} (\beta_{i'j}) \right) \right) \\ &\cong \prod_{i' \in V_j \setminus i} \alpha_{i'j} \cdot \min_{i' \in V_j \setminus i} (\beta_{i'j}) \end{aligned} \tag{4.29}$$

For a binary system over an additive white Gaussian noise channel, the noise power σ^2 can be eliminated, so the equation 4.22 can rewrite as:

$$L(q_{ij}) = L(c_i) = y_i \tag{4.30}$$

4.4 PEG-QC LDPC codes with maximized girth property

There are several new concepts of ECC present to improve wireless communication and future magnetic recording systems. An approach, for example to improve the design of the LDPC codes is a progressive edge-growth (PEG) algorithm presented. This method constructs a parity-check matrix on an edge-by-edge basis giving a large girth. Unfortunately, a relatively large memory is required for storing the parity-check matrices of the random LDPC codes. Therefore, quasi-cyclic (QC) LDPC codes have been proposed to reduce the memory requirement based on block matrices with the circulant architecture. The required memories can be reduced by a factor of $1/p$, where p is the dimension of the circulant matrices. However, QC-LDPC codes have a limitation of flexible block sizes and code rates. Consequently, the QC-LDPC codes based on PEG algorithm or PEG-QC- LDPC codes [6] have been introduced to improve the girth property and the memory requirement, designing the parity-check matrix based on the PEG algorithm by implementing circulant matrices instead of nodes. Recently, there are some methods which improve the random LDPC codes based on the PEG algorithm. An extended PEG algorithm [8] offers the large girth of random LDPC codes by considering the local girth on the symbol nodes before implementing the parity-check matrices. Motivated by this research, we propose the girth-maximized QC-LDPC codes based on the PEG algorithm. This method creates each element of the parity-check matrix based on the PEG algorithm and compares the girth property for every node. Then we select the nodes with the maximum local girth to create the circulant sub-matrix and insert it to the parity-check matrix. The advantages are a lower memory requirement due to circulant architecture and an improvement of PEG algorithm.

Although a PEG-QC-LDPC code can achieve less memory requirement due to the circulant architecture than original PEG LDPC codes. It lacks a large girth property due to circulant block size. Thus, we propose a modification on the PEG-QC-LDPC algorithm to maximize the girth of PEG-QC-

end

end

According to this algorithm, we always calculate the local girth for every symbol node and compare check node sequence with collected check nodes. If the existing check node sequence is similar to any collected check nodes, we compare the girth property for all collected check nodes and select the check nodes with the maximum girth property to establish in the matrix **H**. In contrast, if a different sequence is available, we store the check node set and the girth property, and then delete the check nodes set on the matrix **H** and repeat all steps again.

The required memory for storing our proposed codes can be reduced by a factor $1/p$ compared with the PEG LDPC codes. The parity-check of the matrix **H** can be expressed as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{0,0} & \mathbf{H}_{0,1} & \mathbf{H}_{0,2} & \cdots & \mathbf{H}_{0,t} \\ \mathbf{H}_{1,0} & \mathbf{H}_{1,1} & \mathbf{H}_{1,2} & \cdots & \mathbf{H}_{1,t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{c,0} & \mathbf{H}_{c,1} & \mathbf{H}_{c,2} & \cdots & \mathbf{H}_{c,t} \end{bmatrix}_{cxt}$$

where $\mathbf{H}_{i,j}$ is a $p \times p$ circulant or all-zero matrix, c and t are two positive integers with $c < t$.

CHAPTER 5

SIMULATION RESULTS

The girth-maximized QC-LDPC codes based on the PEG algorithm is the proposed code that eliminate the short cycle length or small girth size. Typically, the small girth size decreases the decoding performance. The proposed code improves the girth property by selects the nodes with the maximum local girth to create the parity-check matrix and improves the memory requirement due to circulant architecture.

The girth property is the key improvement for LDPC design. Regards to the code rate and the degree of each variable node affected to the girth property. Then we consider the girth property on different code rate. To reduce the other effect during evaluate the proposed LDPC codes; we fix the degree of each variable node, the block length and the dimension of the circulant matrix. Then the best LDPC design will select to do simulation to compare the decoding performance under additive white Gaussian noise (AWGN) channel.

The details of the methodology and the result of the girth property and the simulation are showed on the following chapter.

5.1 Girth property

The local girth property of PEG LDPC, PEG-QC LDPC and the proposed PEG-QC LDPC codes are compared with the dimension of the circulant matrix of 128 and the degree of each variable node of 3. The information block size is 4096 bits. Generally, the code rate for magnetic recording system is high code rate over 0.8. The sample girth properties of the LDPC codes of rate 0.89 and 0.86 are shown in the Figure 5.1 and 5.2.

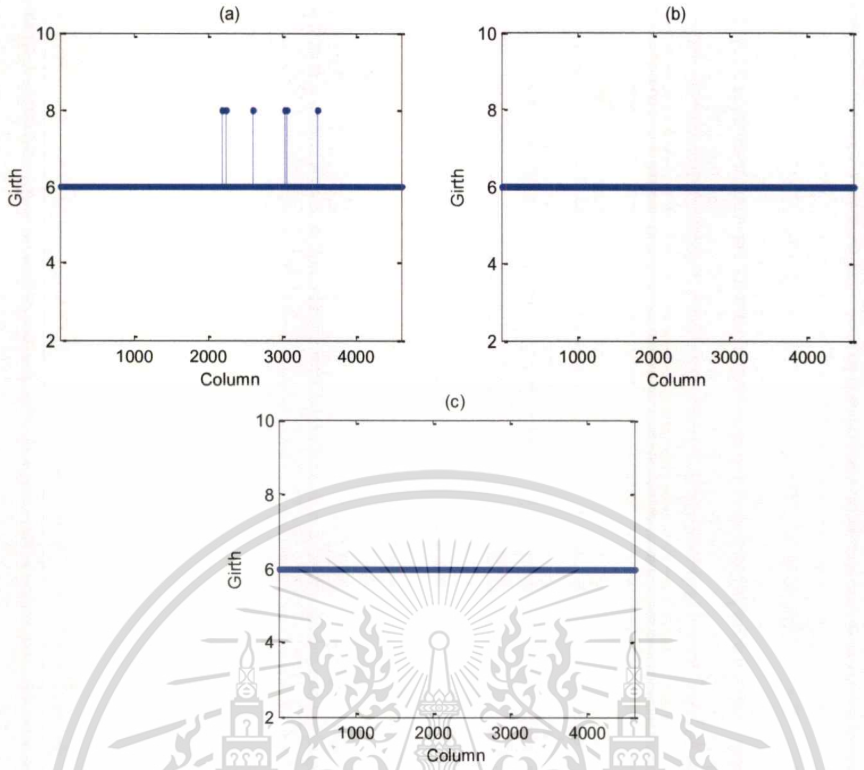


Figure 5.1 The girth profile of code rate 0.89 a) PEG, b) PEG QC and c) proposed PEG QC codes

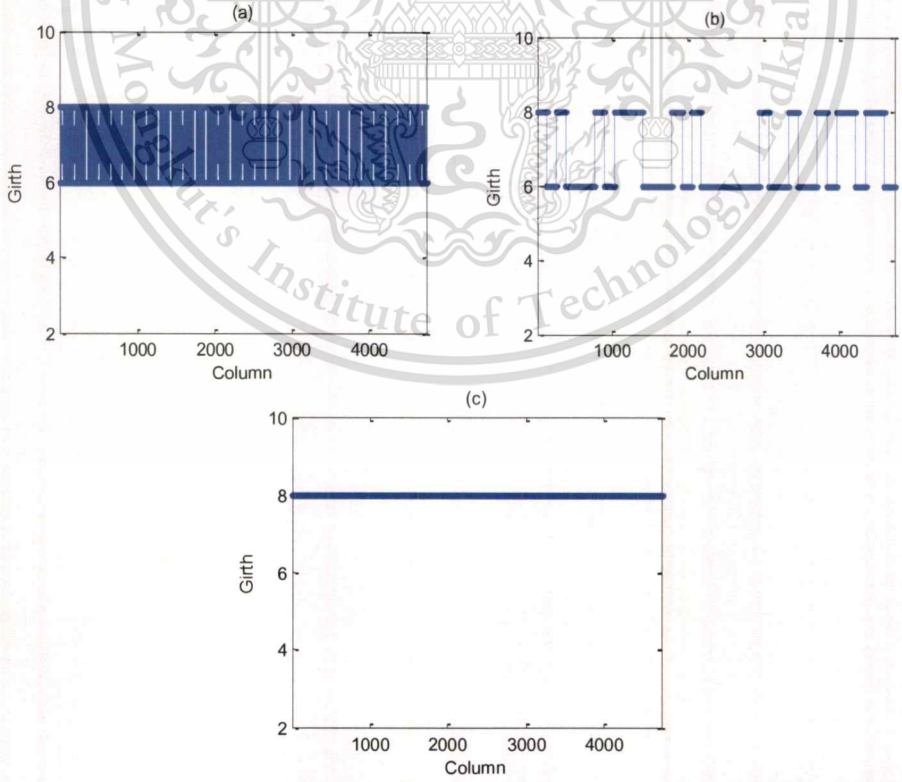


Figure 5.2 The girth profile of code rate 0.86 a) PEG, b) PEG QC and c) proposed PEG QC codes

The girth profile of the code rate of 0.89 for PEG-LDPC, PEG-QC-LDPC and proposed PEG-QC-LDPC codes are comparable. For the code rate of 0.86, PEG-LDPC and PEG-QC-LDPC codes are comparable, but the proposed PEG-QC-LDPC code offers a better girth profile. The girth length of proposed PEG-QC-LDPC is 8 for all symbol nodes. Then, we show the girth property of code rate 0.91, 0.89, 0.86, 0.84, 0.74, 0.64 and 0.50 respectively. The local girth property is shown in Table I by averaging the 100 samples **H** matrices of each code.

Table 5.1 Local girth property for PEG, PEG-QC and proposed PEG-QC codes with various code rates

Code rates	LDPC codes	Local Girths (%)				
		4	6	8	10	12
0.91	PEG	0.00	100.00	0.00	0.00	0.00
	PEG-QC	0.00	100.00	0.00	0.00	0.00
	Proposed PEG-QC	0.00	100.00	0.00	0.00	0.00
0.89	PEG	0.00	99.92	0.08	0.00	0.00
	PEG-QC	0.00	99.95	0.05	0.00	0.00
	Proposed PEG-QC	0.00	99.79	0.21	0.00	0.00
0.86	PEG	0.00	42.42	57.58	0.00	0.00
	PEG-QC	0.00	68.28	31.72	0.00	0.00
	Proposed PEG-QC	0.00	4.27	95.73	0.00	0.00
0.84	PEG	0.00	0.00	100.00	0.00	0.00
	PEG-QC	0.00	45.50	54.50	0.00	0.00
	Proposed PEG-QC	0.00	0.00	100.00	0.00	0.00
0.74	PEG	0.00	0.00	99.99	0.01	0.00
	PEG-QC	0.00	6.98	93.02	0.00	0.00
	Proposed PEG-QC	0.00	0.00	99.98	0.02	0.00
0.64	PEG	0.00	0.00	0.02	99.98	0.00
	PEG-QC	0.00	2.07	51.35	46.58	0.00

Table 5.1 Local girth property for PEG, PEG-QC and proposed PEG-QC codes with various code rates. (cont.)

Code rates	LDPC codes	Local Girths (%)				
		4	6	8	10	12
0.64	Proposed PEG-QC	0.00	0.00	0.00	100.00	0.00
0.50	PEG	0.00	0.00	0.01	6.15	93.84
	PEG-QC	0.00	1.19	16.14	51.72	30.95
	Proposed PEG-QC	0.00	0.00	0.00	64.29	35.71

For low and medium code rates, the PEG codes show a superior local girth property to the PEG-QC codes, that is, the code rate of 0.86, 0.84, 0.74, 0.64 and 0.50. However, for high code rates, both codes show the comparable local girth property, that is, the code rate of 0.89 and 0.91. Furthermore, our proposed codes can maximize a local girth property for every code rate. In the addition, our proposed PEG-QC codes are comparable to the original PEG codes and superior to the PEG-QC codes.

5.2 Decoding performance in an AWGN channel

The aim of the proposed codes is an improvement of PEG algorithm by maximizing the girth property and lower memory requirement due to circulant architecture. Based on Table 5.1, the local girth of the code rate of 0.84 of the proposed codes show 100% of the girth length 8, while the PEG-QC codes show 54.50% of the girth length 8 and 45.50% of the girth length 6. We simulate both codes over an addive white Gaussian noise (AWGN) channel with binary inputs to determine the bit error rate (BER) performance. The AWGN is often observed in electrical circuits in magnetic recording system due to the heat. The simulation diagram is shown in Figure 5.3.

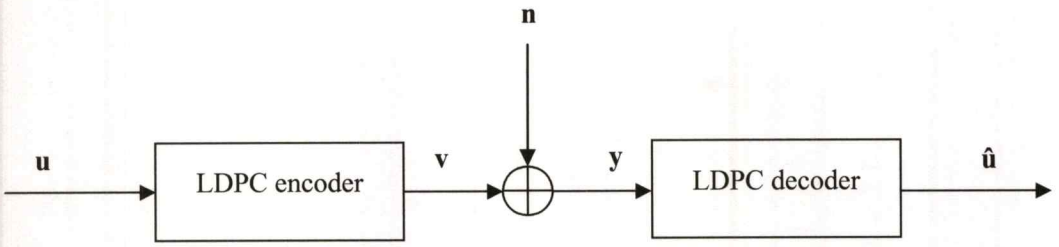


Figure 5.3 The simulation diagram for LDPC codes

where: **u** is the input massage vector,

v is the input code-word,

n is the white Gaussian's noise with power spectrum $N_0/2$ W/Hz,

y is the output code-word,

u is the output massage vector,

The input massage vector are encoded and the parity bits are added (typically call "code-word"). The additional bits are used to correct the error bits due to noise and distortions in communication systems. Then the white Gaussian noise with the power spectral density of $N_0/2$ W/Hz is added to the input code-word. The LDPC decoder processed the received code-words to correct the errornoeuos bits and then output the decided massage bits. The BER is computed from the differrence between input and output massage bits at each signal to noise ratio (SNR) level. The SNR definition follows the equation 5.1 and, in the simulation, we change the SNR level after the number of bit errors equal 300 bits.

$$\left(\frac{E_b}{N_0}\right)_{dB} = 10\log_{10}\left(\frac{1}{R}\cdot\frac{E_c}{N_0}\right) \tag{5.1}$$

where: E_b is average power of massage vetor,

E_c is average power of code word and

R is the code rate

The performance between the PEG-QC codes and the proposed PEG-QC codes are presented in Fig. 5.4, 5.5 and 5.6. The performance is plotted in terms of the bit error rate (BER) versus signal-to-noise ratio (SNR). In simulation, the code rate is 0.91, 0.84 and 0.64. All parameters are summarized in Table 5.2, 5.3 and 5.4.

Table 5.2 The simulation parameters to evaluate the decoding performance of PEG-QC-LDPC codes and our proposed codes over an AWGN channel at the code rate of 0.91.

LDPC codes	PEG-QC codes, Proposed codes
Parity check matrix size (N, K)	(4480, 4096)
Code rates	0.91
Circulant matrix size	128 x 128
Degree of variable nodes	3
SNR level	6.0 – 6.8 dB
Number of iterations	10

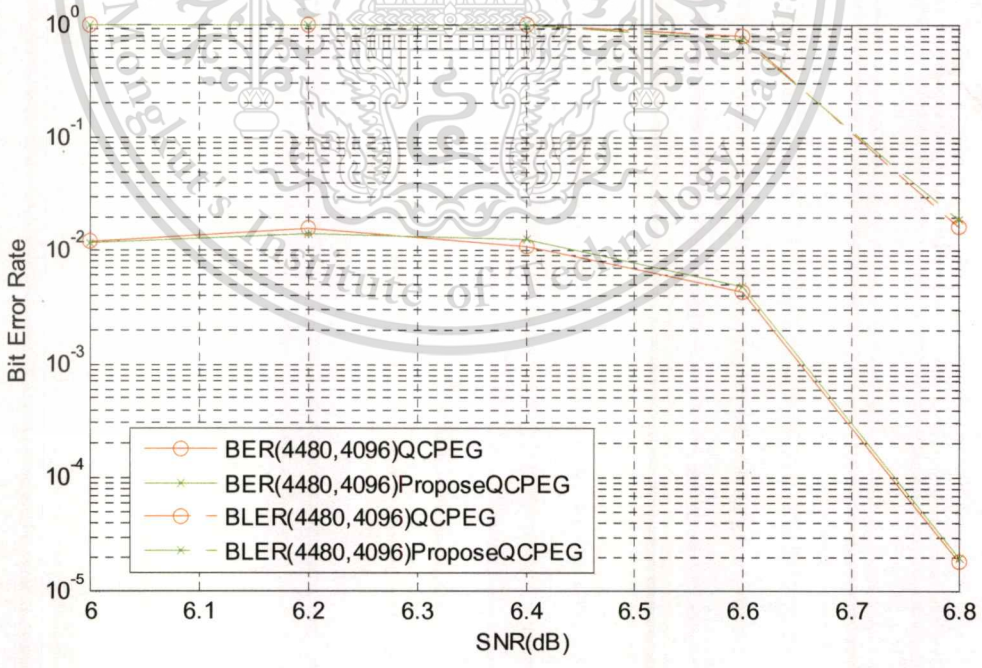


Figure 5.4 BER/BLER of PEG-QC codes and proposed PEG-QC codes (4480, 4096) with 10 iterations and degree of symbol nodes of 3

Table 5.3 The simulation parameters to evaluate the decoding performance of PEG-QC-LDPC codes and our proposed codes over an AWGN channel at the code rate of 0.84.

LDPC codes	PEG-QC codes, Proposed codes
Parity check matrix size (N, K)	(4864, 4096)
Code rates	0.84
Circulant matrix size	128 x 128
Degree of variable nodes	3
SNR level	2.7 – 3.9 dB
Number of iterations	50

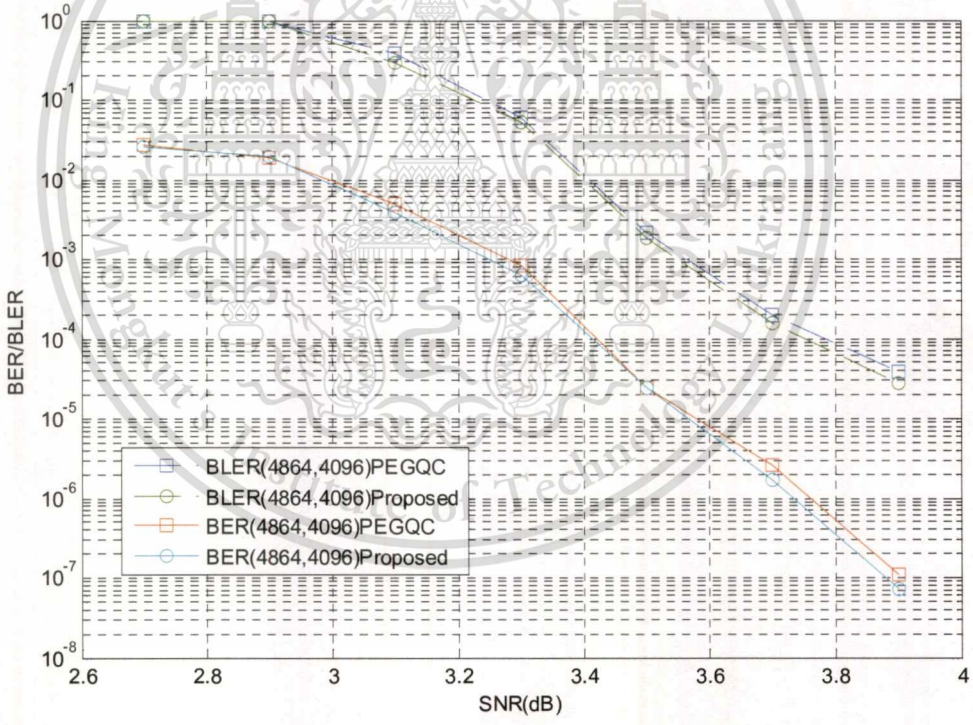


Figure 5.5 BER/BLER of PEG-QC codes and proposed PEG-QC codes (4864, 4096) with 50 iterations and degree of symbol nodes of 3

Table 5.4 The simulation parameters to evaluate the decoding performance of PEG-QC-LDPC codes and our proposed codes over an AWGN channel at the code rate of 0.64.

LDPC codes	PEG-QC codes, Proposed codes
Parity check matrix size (N, K)	(6400, 4096)
Code rates	0.64
Circulant matrix size	128 x 128
Degree of variable nodes	3
SNR level	1.8 – 2.6 dB
Number of iterations	10

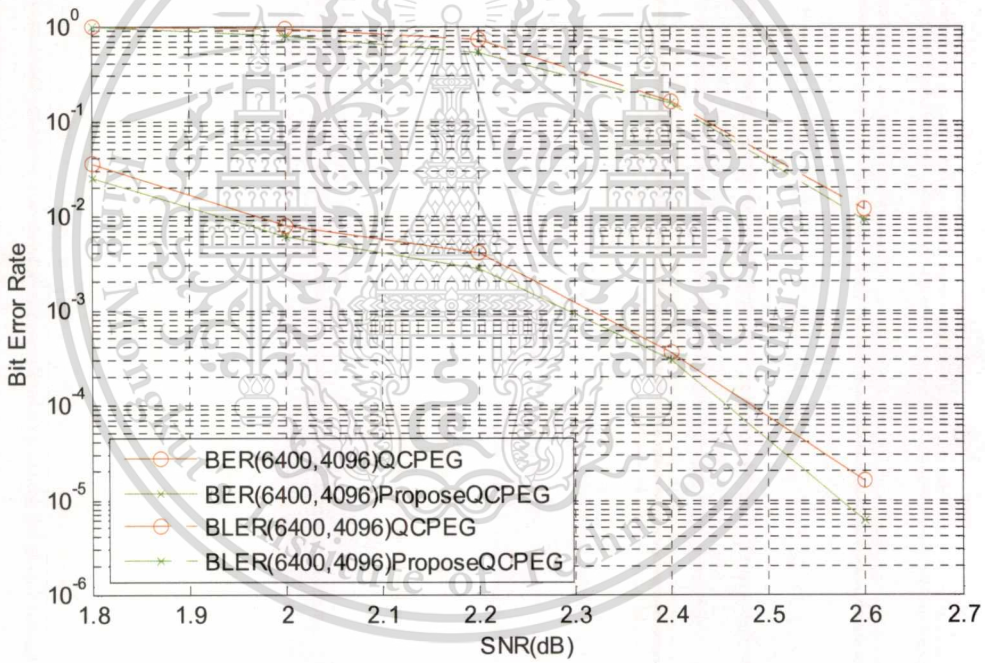


Figure 5.6 BER/BLER of PEG-QC codes and proposed PEG-QC codes (6400, 4096) with 10 iterations and degree of symbol nodes of 3

At the code rate of 0.91, the proposed PEG-QC codes and the PEG-QC codes present the comparable BER/BLER performance. At the codes rate of 0.84, the SNR gain of the proposed PEG-QC codes over the PEG-QC codes is about 0.03 dB at the BER of 10^{-7}. At the BLER of 10^{-4}, the proposed PEG-QC codes offer the SNR gain of about 0.05 dB compared with the PEG-QC-LDPC

codes. At the codes rate of 0.64, the SNR gain of the proposed PEG-QC codes over the PEG-QC codes is about 0.07 dB at the BER of 10^{-6}. At the BLER of 10^{-3}, the proposed PEG-QC codes offer the SNR gain of about 0.02 dB compared with the PEG-QC-LDPC codes.

We then consider the performance at various decoding iteration numbers. The proposed PEG-QC codes and the PEG-QC codes are simulated at the code rate of 0.84, block size of 4864 bits, SNR of 3.70 dB and the iteration numbers of 1 to 80. The performance results in terms of BER versus number of iterations are plotted in Figure 5.7 and all of the parameters are summarized in Table 5.5.

Table 5.5 The simulation parameters to evaluate the performance at various iteration numbers.

LDPC codes	PEG-QC codes, Proposed codes
Parity check matrix size (N, K)	(4864, 4096)
SNR level	3.7 dB
Number of iterations	1-80

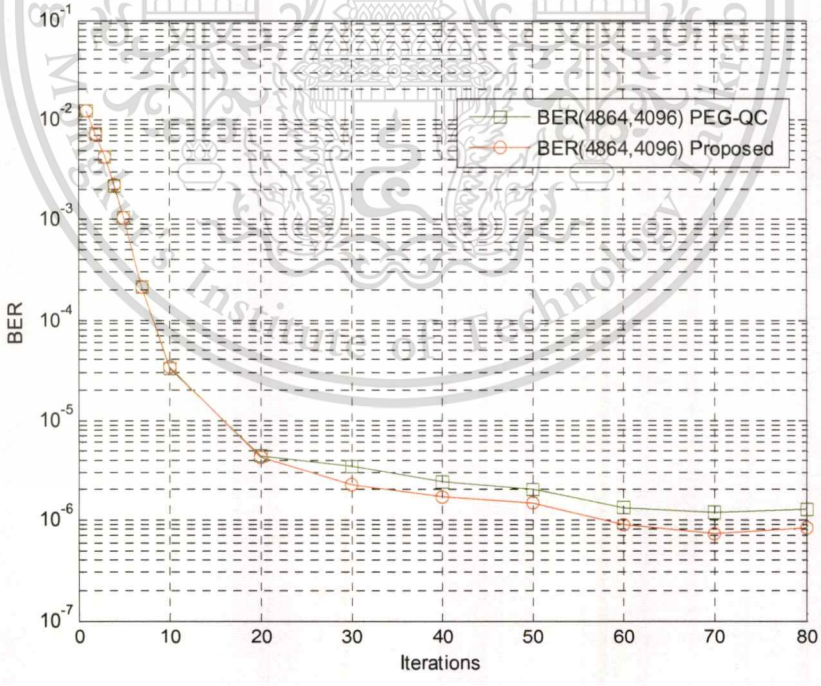


Figure 5.7 BER of PEG-QC codes and proposed PEG-QC codes with size (4864, 4096) at SNR 3.70 dB at various iteration numbers.

For the iteration numbers 1 to 20, the performance of proposed codes is similar to that of PEG-QC codes. But at the iteration numbers of 30 to 80, the proposed PEG-QC codes show the improved BERs compared with those of the PEG-QC codes.

CHAPTER 6

CONCLUSIONS

A low-density parity-check (LDPC) code is a linear error correcting code which approaches the Shannon's limit in an additive white Gaussian noise channel. Recently, there are several techniques to improve the performance of LDPC codes such as progressive-edge growth (PEG) algorithm and quasi cyclic (QC) LDPC codes. In this study, we improve the girth property of the parity-check matrix. We propose a local-girth optimized PEG algorithm for QC LDPC codes to construct the parity-check matrix with the maximum local-girth.

For low and medium code rates, the PEG codes show a superior local girth property to the PEG-QC-LDPC codes. At high code rates, both codes show a comparable local girth property. But our proposed method can maximize a local girth property for every code rate under consideration. Our proposed PEG-QC-LDPC codes are comparable to the original PEG-LDPC codes and superior to the PEG-QC-LDPC codes. The simulation results of the LDPC codes in an additive white Gaussian noise (AWGN) channel with binary inputs show that the proposed PEG-QC codes are superior in the BER/BLER performance to PEG-QC codes and show the SNR gains of about 0.07 dB at the BER of 10^{-6} at the code rate of 0.64 and the SNR gains of about 0.03 dB at the BER of 10^{-7} at the code rate of 0.84. Regarding to the quasi-cyclic architecture, our proposed PEG-QC codes improve the memory requirement by a factor of p, where p is the size of the circulant sub matrix, compared with original PEG codes which construct the parity check matrix via the random method.

The proposed PEG-QC codes have the parity-check matrix with improved memory requirement and superior decoding performance.

REFERENCES

[1] R. G. Gallager, "**Low-density parity-check codes**," *IRE Transactions on Information Theory*, vol. IT-18, pp. 21-28, Jan. 1962.

[2] D. J. C. MacKay and R. M. Neal, "**Near Shannon limit performance of low density parity check codes**," *Electronics Letters*, vol. 32, No. 18, pp. 1645-1646, 1996.

[3] T. J. Richardson, M. A. Shokrollahi, and R. Urbanke, "**Design of capacity-approaching irregular low-density parity-check codes**," *IEEE Transactions Information Theory*, vol. 47, pp. 619-637, Feb. 2001.

[4] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "**Regular and irregular progressive edge-growth tanner graphs,**" *IEEE Transactions Information Theory*, vol. 51, pp. 386-98, 2005.

[5] S. Lin, L. Chen, J. Xu and I. Djurdjevic, "**Near Shannon limit quasi-cyclic low-density parity-check codes**," in *Proc. IEEE Global Communications Conference*, Vol. 4, pp. 2030 – 2035, Dec. 2003.

[6] Z. Li and B. V. K. V. Kumar, "**A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph**," in *Proc. 38th Asilomar Conf. Signals, Systems and Computers, Monterey*, CA, Nov. 7–10, 2004, vol. 2, pp. 1990–1994.

[7] H. Xiao and A. H. Banihashemi, "**Improved progressive-edge-growth (PEG) construction of irregular LDPC codes,**" *IEEE Communication Letters*, vol. 8, no. 12, pp. 715–717, Dec. 2004.

[8] Z. Zhou, X.Li, D.Zheng, K.Chen and J. Li, "**Entended PEG Algorithm for high rate LDPC codes,**" in *Proc. IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 494-498, 2009.

[9] J. L. Fan, "**Array codes as low-density parity check codes**," in *Proc. 2nd International Symposium on Turbo Codes & Related Topics*, pp. 543-546, 2000.

[10] E. Eleftheriou and S. Olcer, "**Low-density parity-check codes for digital subscriber lines**," in *Proc. IEEE Int. Conf. Commun. (ICC), 2002*, pp. 1752-1757 vol.3, 2002.

APPENDIX

Publication

P. Prompakdee, W. Phakphisut and P. Supnithi, "**Quasi Cyclic-LDPC codes based on PEG algorithm with maximized girth property**," in *Intelligent Signal Processing and Communication Systems (ISPACS)*, Chiangmai, Dec. 7-9, 2011.

AUTHOR BIOGRAPHY

Name	Mr. Patanasak Prompakdee
Date of birth	29 August 1984
Address	155 Moo.12 Tambon Phichai Amphur Muang
	Lampang province 52000

Educational Background 2007 B.Sc. Physics, Thammasat University (Second honor)

Experience

2008-Present Electrostatic Discharge (ESD) Engineer of Western digital Thailand