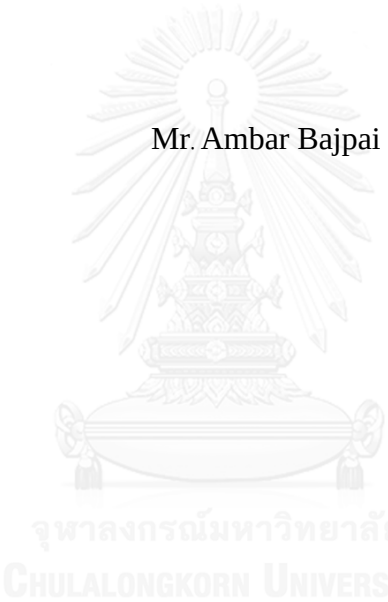
การออกแบบและการประเมินสมรรถนะวิธีเข้ารหัสความซับซ้อนต่ำสำหรับรหัสแอลดีพีซี

นายแอมบา บาจไพ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558

DESIGN AND PERFORMANCE EVALUATION OF LOW COMPLEXITY ENCODING METHODS FOR LDPC CODES

Mr. Ambar Bajpai



A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2015

Thesis Title	DESIGN AND PERFORMANCE EVALUATION OF LOW COMPLEXITY ENCODING METHODS FOR LDPC CODES
By	Mr. Ambar Bajpai
Field of Study	Electrical Engineering
Thesis Advisor	Associate Professor Lunchakorn Wuttisittikulkij, Ph.D.
Thesis Co-Advisor	Associate Professor Piya Kovintavewat, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

.....Dean of the Faculty of Engineering
(Associate Professor Supot Teachavorasinskun, Ph.D.)

THESIS COMMITTEE

.....Chairman
(Associate Professor Watit Benjapolakul, Ph.D.)

.....Thesis Advisor
(Associate Professor Lunchakorn Wuttisittikulkij, Ph.D.)

.....Thesis Co-Advisor
(Associate Professor Piya Kovintavewat, Ph.D.)

.....Examiner
(Assistant Professor Chaiyachet Saivichit, Ph.D.)

.....Examiner
(Lecturer Panuwat Janpugdee, Ph.D.)

.....External Examiner
(Lecturer Pisit Vanichchanunt, Ph.D.)

แอมบา บาจไพ : การออกแบบและการประเมินสมรรถนะวิธีเข้ารหัสความซับซ้อนต่ำสำหรับรหัสแอลดีพีซี (DESIGN AND PERFORMANCE EVALUATION OF LOW COMPLEXITY ENCODING METHODS FOR LDPC CODES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.ลัญฉกร วุฒิสิทธิกุลกิจ, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: รศ. ดร.ปิยะ โควินท์ทวีวัฒน์, 100 หน้า.

มีความก้าวหน้าครั้งสำคัญในศาสตร์ทางด้านเทคนิคการแก้ไขความผิดพลาดไปข้างหน้าที่เรียกว่า รหัสพาริตีเช็กความหนาแน่นต่ำ (แอลดีพีซี) รหัสประเภทนี้มีศักยภาพที่จะสามารถเข้าใกล้ขีดจำกัดทางทฤษฎีแชนนอนได้ในด้านความจุของช่องสัญญาณ ในช่วงที่ผ่านมานักวิจัยมีความสนใจอย่างมากในการพัฒนารหัสแอลดีพีซี เนื่องจากมีสมรรถนะที่ดีกว่ารหัสเทอร์โบ อย่างไรก็ตามรหัสแอลดีพีซีจะให้สมรรถนะที่ดีมาก เมื่อรหัสที่ใช้มีโครงสร้างไม่สม่ำเสมอ และใช้บล็อกที่มีขนาดใหญ่มาก อย่างไรก็ตามรหัสแอลดีพีซีที่ใช้บล็อกขนาดสั้นแม้จะมีประสิทธิภาพไม่ดีเท่า แต่รหัสเหล่านี้สามารถนำไปประยุกต์ใช้งานได้ง่ายในทางปฏิบัติ ดังนั้น การเข้ารหัสแอลดีพีซีเพื่อให้ได้รหัสที่มีอัตรารหัสและความยาวของรหัสที่หลากหลายจึงยังเป็นสิ่งที่จำเป็นอย่างยิ่งเพื่อให้สามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนารหัสแอลดีพีซีแบบควอสิ-ไซคลิก ที่มีน้ำหนักคอลัมน์เท่ากับ 3 โดยส่วนแรกของวิทยานิพนธ์จะนำเสนอขั้นตอนการสร้างแบบใหม่ของรหัสแอลดีพีซีแบบควอสิ-ไซคลิกที่มีความยาวรหัสขนาดกลางถึงขนาดใหญ่ ด้วยวิธีการรวมรหัสแอลดีพีซีแบบควอสิ-ไซคลิกที่มีขนาดเล็ก ในรูปของรหัสส่วนประกอบ ให้ได้เป็นรหัสที่มีขนาดใหญ่ขึ้นโดยใช้ทฤษฎีบทเศษของจีน รหัสส่วนประกอบดังกล่าวสร้างขึ้นโดยการสลับหรือเปลี่ยนแปลงแต่ละคอลัมน์ในบล็อกตามลำดับเพื่อให้บรรลุเกิร์ธที่ตั้งไว้ หลังจากการรวมรหัสส่วนประกอบทั้งหมดจะได้เป็นรหัสที่มีขนาดใหญ่ขึ้นตามต้องการ โดยเมทริกซ์พาริตีเช็กที่ได้มีเกิร์ธที่ใหญ่เท่ากับค่าเกิร์ธที่สูงสุดของรหัสส่วนประกอบ จากผลการทดสอบด้วยโปรแกรมคอมพิวเตอร์พบว่าวิธีการสร้างเมทริกซ์พาริตีเช็กที่นำเสนอให้ประสิทธิภาพที่ดีกว่าวิธีการอื่นๆ ที่มีอยู่อย่างมีนัยสำคัญ ในรูปของพื้นผิดพลาดต่ำ โครงสร้างที่เรียบง่าย มีประสิทธิภาพมากขึ้น และสามารถลดความซับซ้อนในขั้นตอนการเข้ารหัสได้

นอกจากนี้วิทยานิพนธ์ฉบับนี้ยังได้นำเสนอวิธีการสร้างรหัสแบบใหม่ของรหัสแอลดีพีซีควอสิ-ไซคลิก 2 แบบคือ วิธีฐานเมทริกซ์ซึ่งเป็นวิธีที่ง่ายและมีความซับซ้อนในการคำนวณต่ำสำหรับการสร้างองค์ประกอบเมทริกซ์ของรหัสแอลดีพีซี $(3, K)$ ที่มีเกิร์ธเท่ากับ 8, 10 และ 12 และวิธีที่สองคือวิธีการลบที่มีองค์ประกอบเมทริกซ์คล้ายกับวิธีฐานเมทริกซ์และมีเกิร์ธน้อยสุดเท่ากับ 8 จากการทดลองพบว่ารหัสแอลดีพีซีที่สร้างจากวิธีการทั้งสองวิธีนี้มีความยืดหยุ่นเหมาะสำหรับ K ใดๆ และมีประสิทธิภาพดีกว่าเมื่อเทียบกับวิธีการที่มีอยู่เดิม

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558 ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

5471468521 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS: GIRTH / LDPC CODES / QC-LDPC CODES / CHINESE REMAINDER THEOREM (CRT)

AMBAR BAJPAI: DESIGN AND PERFORMANCE EVALUATION OF LOW COMPLEXITY ENCODING METHODS FOR LDPC CODES. ADVISOR: ASSOC. PROF. LUNCHAKORN WUTTISITTIKULKIJ, Ph.D., CO-ADVISOR: ASSOC. PROF. PIYA KOVINTAVEWAT, Ph.D., 100 pp.

There is an important breakthrough in the field of forward error correction techniques, low-density parity-check (LDPC) codes. These codes have potential to approach the Shannon limit with reliable performance on a given channel, known as *the channel capacity*. Recently, researchers pay much more attention towards LDPC codes because its performance is better than Turbo codes. However, the LDPC code is the best when it is an irregular code and uses with a very large block-length. Although, the short block-length LDPC codes perform not so well, they are easy to implement in various practical applications. Therefore, there is still a demand of new development on the encoding side of LDPC codes for various ranges of code rate and code length so as to capable of using in many applications.

This thesis proposes the development of quasi-cyclic (QC) LDPC codes for column weight 3. In the first part of this thesis, we propose a new construction algorithm of QC-LDPC codes for medium to large block-length by combining QC-LDPC codes of small block-length as their component codes, via Chinese remainder theorem (CRT). Such component codes were constructed by permuting each column block sequentially to attain the desire local girth. After combining all component codes to generate an expanded parity-check matrix, the resulting girth is greater than or at least equal to the maximum girth of the component codes. Simulation results show that our proposed construction method of the parity-check matrix significantly outperforms the other well-known existing methods in terms of low error-floor, simple structure, high performance, and can reduce encoding complexity.

In addition, this thesis also proposes two new construction methods for QC-LDPC codes, namely a base matrix method and a subtraction based method. A base matrix based method is a simple, less computational complexity method for constructing the exponent matrix (3, K) of girth 8, 10 and 12 of QC-LDPC codes. Another method is a subtraction method, which has a similar exponent matrix as a base matrix method and has a girth at least 8. Results indicate that the LDPC codes constructed from these two methods have flexibility for arbitrary block-column length K and have a similar BER performance if compared to existing methods.

Department: Electrical Engineering
Field of Study: Electrical Engineering
Academic Year: 2015

Student's Signature .....
Advisor's Signature .....
Co-Advisor's Signature .....

ACKNOWLEDGEMENTS

Firstly I would like to thank my supervisor Assoc. Prof. Lunchakorn Wuttisittikulkij for giving me the opportunity to pursue my doctoral studies on a fascinating subject such as coding theory in wireless communications. I benefited tremendously from his vision and always feel motivated by his own dedication to research and hard work "No-pain No-gain" philosophy. Thanks for giving me the chance to visit many interesting places during the Ph.D. years and finally thanks for all the help including financial support and encouragement that I got during my Ph.D. studies. In addition, I would like to thank my co-advisor Assoc. Prof. Piya Kovintavewat, without his guidance and long constructive meetings, this work is very hard to complete.

I was mostly associated with Telecommunication System Research Laboratory (TSRL) during my Ph.D. studies. Thanks to all the staff of the Electrical Engineering Department for all the help, assistance, and kindness. A special acknowledgement goes to the entire postgraduate students present and past for the technical and interpreting assistance, helpful discussions with Muhammad Saadi and the good time in TSRL Lab, Inthanin coffee shop, Rangnaam apartment, and faculty of arts canteen.

On a financial note, many thanks goes to Dr. Chaodit for allowing me to do TA work for Electrical Engineering department, Dr. Pisit for allowing me to teach in KMUTNB, Bangkok for part time, International School of Engineering (ISE) for TA work in various subjects related to communication systems. I would also acknowledge thanks to the graduate school for prestigious 90th year Chulalongkorn University scholarship and overseas publication presentation scholarship.

Above all, I would like to express appreciation to my ingenious wife Anshul Shukla, for her support, patience, counseling and encouragement in my most difficult times. She also worked as an international affair officer in ISE, Chulalongkorn University, Bangkok for almost three years to cultivate her career goals, generate funds for good life in Bangkok. We visited many places and experienced different culture in this beautiful world. Nevertheless, thanks to the almighty God and our many visits to wat kheak, Vishnu temple and lord Ganesha temple in the Central world. We are blessed with a beautiful daughter Prisha Arya Bajpai born in Chulalongkorn hospital Bangkok.

Finally, my gratitude goes to my loving parents, in-laws, my siblings Vasundhara and Ankur, and especially to my wife Anshul. I dedicate this work to them for their love and support that made it possible.

CONTENTS

Page

Page

Page

List of Figures

List of Tables

List of Abbreviations

APP	=	A Posteriori Probability
AWGN	=	Additive White Gaussian Noise
BER	=	Binary Error Rate
BP	=	Belief Propagation
BPSK	=	Binary Phase Shift Keyed
BSC	=	Binary Symmetric Channel
BHC	=	Bose-Chaudhuri-Hocquenghem
FEC	=	Forward Error Correction
FER	=	Frame/Block Error Rate
FFT	=	Fast Fourier Transform
GF	=	Galois Field
LDPC	=	Low Density Parity Check
LLR	=	Log Likelihood Ratio
MAP	=	Maximum A Posteriori
ML	=	Maximum Likelihood
MIMO	=	Multiple Input Multiple Output
PEG	=	Progressive Edge Growth
QC	=	Quasi Cyclic
QC-LDPC	=	Quasi-Cyclic Low-Density Parity-Check
SNR	=	Signal to Noise Ratio
SPA	=	Sum Product Algorithm

Chapter 1: Introduction

1.1 Overall scenario of communication systems and reliable transmission

Now a days Low-Density Parity-Check (LDPC) codes are one of the popular research topics. The milestones for LDPC research began in 1962, when R. Gallager introduced LDPC [1], in his PhD thesis. Since then, these codes were ignored for almost next 30 years due to complexity and less analytical tools available at that time. Later these codes were rediscovered by Mackay and Neal in 1996 [2]. Enormous potential research has been carried out on channel coding since Shannon's theory of mathematical constraints for channel capacity in 1948 [3]. It is now a well-known fact that highly random LDPC code construction methods can reach very close to Shannon limit [4]. Various standards such as IEEE 802.11n, Wi-MAX, DVB-S2, and so forth have adopted LDPC codes as channel codes. Today, LDPC codes are considered as the most eligible channel codes for future generation high data rate communication and various practical applications. The development of most optimized and efficient, constructed LDPC codes have also been also studied widely in the current decade. LDPC codes can provide lower error probabilities than equivalent conventional forward error correcting codes.

In practice, the performance of LDPC codes depends on various system parameters. It is very important to carefully design LDPC parity-check matrices, having sufficient iterations, ultra-sparse and low cycle presence, therefore, fulfilling these constraints leads to significantly optimized performance as stated by Chung *et al*. in 2001 [5]. Result approached to 0.0045 dB from Shannon's limit exists only for large block-length. Clearly, a large block-length results in a large parity-check matrix and hence a large generator matrix. Thus, LDPC codes are defined by a sparse parity matrix, in which most of the entries are zero and only few portions are nonzero values (the smaller the portion of nonzero entries, the less the encoding and decoding complexity). Generally, the complexity will increase by a factor of $\mathrm{O}\left(q^2\right)$, where q is

the order of Galois field GF (q). Large block length with GF (2) LDPC codes, although has good performance but increase scheduling time, hence if we increase the order of Galois field (i.e., increase q), the performance will significantly improve at an expense of the complexity factor $\mathrm{O}\left(q^2\right)$ of decoding.

1.2 A typical digital communication systems

Modern society is developed generation by generation, based on gaining knowledge from past and creating new for present and future. The amount of knowledge grows extra-ordinary and many people interact with each other very often. This all happen by technology revolution in digital communication systems. Almost all information, which we received in our day-to-day life, is digitalized. People are using various multimedia devices to communicate among themselves at work, offices, schools etc. Examples of popular digital communication systems are mobile telephony, satellite broadcasting, optical fiber communication and wireless or wired connection to the internet. The foundation stone of modern era communication systems following on Shannon's [6] work on information coding theory. Figure 1-1 shows the simplified block diagram of a typical digital communication system. The input data at the source may be analogue or a digital followed be a source encoder, which converts source data in to the sequence of binary bits which is called as message sequence. Furthermore channel coder adds some redundant bits named as parity bits on message sequence for forward error correction (FEC). This encoded message is termed as a codeword, which feds input to the modulator for high frequency transmission to the channel. When the signal propagates through the channel it gets corrupted by the noise. At the receiver end, demodulator again converts signal in to the digital information in the form of binary sequence. Channel decoder applies decoding algorithm for extracting original bits. These bits are further decompressed by source decoder to user readable format.

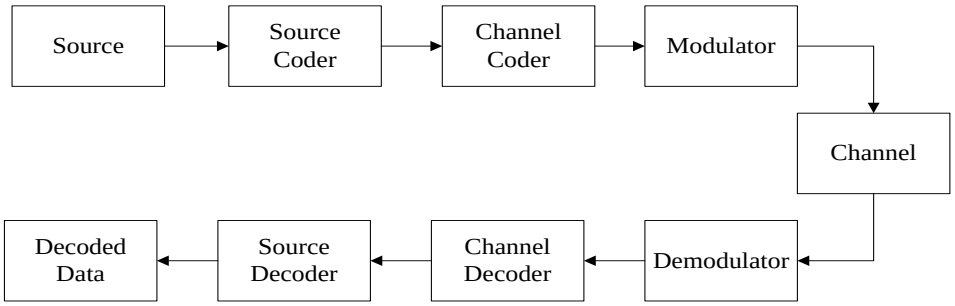


Figure 1-1: A typical digital communication system block diagram

1.3 Thesis motivation

The limitation associated with constructing a medium to large size block-length codeword from the **H** matrix is the requirement of large data storage and computation complexity. Large block-length and irregular LDPC codes give better performance but also leads to more complex construction of **H** matrix. To overcome the problem of memory required in data storage, another structured class of LDPC codes known as the Quasi-cyclic LDPC (QC-LDPC) has studied widely [7], due to its less memory requirement and less complexity encoding procedures. It suits best for many practical applications and adopted in various IEEE standards. QC-LDPC with circulant permutation matrix (CPM) proposed by [7-10]. Furthermore to enlarge parity-check matrix significant contribution proposed method are combining techniques using the Chinese remainder theorem as in [11]. Large block-length size and irregular LDPC codes performs near to Shannon limit but also leads to increased complexity to construct good parity check matrix. There is still a grave gap to reduce complexity in encoding of LDPC codes.

1.4 Need for the project

The challenge is to design practical, finite length codes which can achieve extremely low bit error rates at small SNR values. Such high performance is very useful in many practical applications for wireless communication. The goal of this research is to propose a novel scheme for **H** matrix with optimized lower bound with less time for computational complexity; the optimized lower bound may lead to less hardware requirement in terms of memory requirement hence more economical in practical applications.

1.5 Thesis Contribution

The main objective of this research to develop the **H** matrix for medium to large block-length structured QC-LDPC codes with high girth and good error correcting performance in order to assure less memory bits require for hardware implementation which leads to save costs and computation time of **H** matrix.

We propose a new construction algorithm of Quasi-cyclic low-density parity-check (QC-LDPC) codes of medium to large block-length by combining QC-LDPC codes of small length as their component codes, via the Chinese remainder theorem (CRT). We construct such component codes by permuting column block sequentially with maximizing the local girth for each column block. After combining all component codes to the parity-check matrix, **H**, the girth obtained will be greater than or equals to the highest girth of component codes. Our method provides efficient and fast encoding, have very simple structure and more economical in terms of hardware architecture.

In addition, we also propose two explicit methods to construct structured QC-LDPC codes based on a base matrix method and a subtraction method for column weight 3, which have comparable performance with existed work.

List of journal and conference papers published by the author can be found in Appendix.

1.6 Impact on Society and Scientific Community

LDPC channel codes are considered to be the best choice for FEC in communication systems. It has near Shannon capacity performance with low error floor, which make it best suited for many standards developed by an IEEE, ETSI DVB S2/T2 and Chinese organizations such as Advanced Broadcasting System-Satellite (ABS-S). The project output will benefit to a wide range of existing and future communication services including personal home area network, WLAN in public areas and wireless services to business and government units Table 1-1 illustrate standards using QC-LDPC codes as a FEC channel codes

Table 1-1: Standards using QC-LDPC codes as a FEC channel code

Standard's Name	Application and potential uses
IEEE 802.11n	Wireless networking standards uses MIMO, leads to significant improvement of throughput of maximum data rate from 54 Mbit/s to 600 Mbit/s.
Standard's Name	**Application and potential uses**
IEEE 802.16e	Mobile WiMAX or Wi-Bro in Korea, sometimes also branded 4 G and offers peak speed data rate of 128 Mbit/s.
IEEE 802.11ac	Upcoming brand name of Wi-Fi, provide high-throughput in 5 GHz band from at least 500 Mbit/s to 1 Gbits/s.
IEEE 802.20	Mobile Broadband Wireless Access (MBWA)
ETSI DVB-S2	Digital Video Broadcasting – Satellite 2nd Generation introduced HDTV and H.264 video Codecs
ETSI DVB-T2	Digital Video Broadcasting – Terrestrial 2nd Generation carrying HD signal in to terrestrial channels.
ABS-S China	(Advanced Broadcasting System-Satellite) China DTH competitive standard for DVB-S2

1.7 Brief organization of the thesis

The organization of thesis is as follows:

In chapter 2, the background knowledge of channel coding and a review of binary LDPC codes are given. A brief history about channel coding and linear block codes is described followed by the major milestones achieved in this area. After, review of binary LDPC codes, there is brief description of important parameters used in designing LDPC codes. In addition, decoding algorithms of LDPC codes are described. Various definitions used in designing of LDPC codes and parameters required for optimization of LDPC codes also enlightened. Lastly, the chapter concludes by summary.

Chapter 3 discusses the various LDPC code construction techniques which include brief review of randomly constructions and structured constructions techniques. We also describe why structured LDPC codes are more useful and many milestone developments discussed. Lastly, we focus on QC-LDPC code construction technique, its implementation and various practical benchmarked applications.

Chapter 4 has detailed discussion, analysis on proposed method of QC-LDPC codes and its expansion using CRT. In addition we generalize our codes for achieving high girth value with optimized lower bound and good BER performance. At the end, we discuss about complexity parameter and storage uses analysis of proposed codes. In addition, Chapter 5 presents two explicit proposed methods for QC-LDPC codes, and describes simulation results and analysis of the proposed codes.

In Chapter 6, conclusion has been drawn for this dissertation and the possible future directions of research work are discussed to implement same work for non-binary QC-LDPC codes.

Chapter 2: Background and definitions of LDPC codes

In this chapter, the author presents an overview of low-density parity-check (LDPC) codes. As the name depicts, these codes are in the category of block codes defined in the form of parity-check matrix with low density of number of 1's. They were first proposed by Gallager in 1962 [1]. These codes have iterative decoding scheme which have increased complexity as block-length increases. They were reinvented in 1996 by Mackay and Neal [2] and, since then, many researchers have contributed remarkable literature for practical wireless communication standards. These codes beat all other existing FEC codes for half rate and large block-length in terms of BER performance and decoding complexity. It is the world's best performing code and falling only 0.04 dB short of Shannon limit [5].

2.1 Milestones of coding theory

2.1.1 Shannon's Theorem

Year 1948 makes historic milestone for information theory, Claude E. Shannon published capacity approach paper for channel coding [3]. This theory applies limits to reliable transmission of data over unreliable channels and methods on how to target these limits. In addition, this mathematical relationship computes the amount of information, and establish bound for the maximum amount of information that can be transmitted over unreliable channels. Codes that can approach capacity are very good from a communication point of view, but Shannon's theorems are non-constructive and do not give an evidence on how to find such excellent codes. More importantly, even if an oracle gave us sequences of codes that achieve capacity for a certain rate, it is not clear how to encode and decode them efficiently. Design of codes with encoding and decoding algorithms which approach the capacity of the channel is the main topic of research since the race of high throughput wireless communication

begins. While the fundamental bounds have been known for many years, only significant recent works on turbo codes [12] and low-density parity-check (LDPC) codes [1] has resulted in practical codes that can approximately close to capacity. However, there are limitation on these codes as code length approaches infinity and valid for selected particular channels. So forth channel coding component still has wide room for research more specifically on finite length codes and decoding for various channels. Different scenario's presents different aspects to channel coding. Communication channel consists of two types:

- Binary Erasure Channel (BEC)
- Binary Symmetric Channel (BSC)

In both types of channel, input information is binary i.e., '1' or '0'; in the case of BEC output information consists of 0 and 1 along-with an additional element denoted as be called erasure.

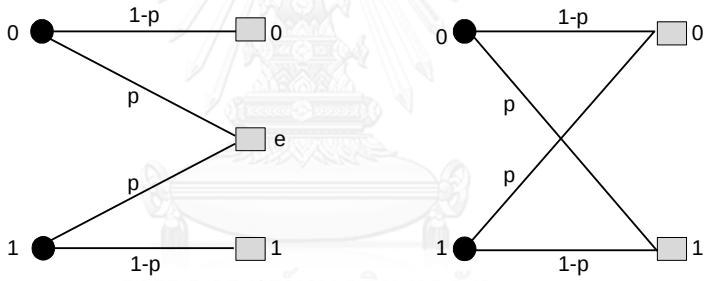


Figure 2-1: Examples of channels: (a) The Binary Erasure Channel (BEC) with erasure probability e, and (b) The Binary Symmetric Channel (BSC) with error probability p

Each bit is either transmitted correctly (with probability $1-p$), or it is erased (with probability p).

In the case of the BSC, each bit is either transmitted correctly with probability $1-p$, or it is flipped with probability p. This channel may seem simpler than the BEC at the first sight, but in fact it is much more complicated. The complication arises since it is not clear which bits are flipped. (In the case of the BEC it is clear which bits are erased.). The capacity of this channel is $1+p\log_2 p+(1-p)\log_2(1-p)$.

The challenge for channel coding is to provide high data rates with extremely low bit-error-rates at small SNR values, while still keeping the complexity low. Due to the complexity of the analysis, we will investigate performance on a simplify channel with binary erasure channel (BEC), where bits are either received correctly or are erased with probability *e*.

Conventionally, modern coding theories are divided into two categories namely algebraic coding theory and probabilistic coding theory. In the beginning, coding theory based on algebraic class mainly associated for linear block codes in a binary field. LDPC codes are also a sub class of linear block codes. These codes pursue advantage of having a strong algebraic structure which leads to allow studies of their mathematical properties having efficient encoding and decoding techniques. Till now, one of the major emphasis of codes based on algebraic theory, to achieve the maximum hamming distance i.e., the minimum number of positions where any two codewords can differ. This is because of motivation behind better error correction in presence of the unreliable channel since further apart codewords are less likely erroneous. Some of the renowned channel codes developed so far are Hamming codes [13], Golay codes [14], Reed-Muller [15, 16], Reed-Solomon codes [17], Bose-Chaudhari-Hocquenghem (BCH) codes [18] and algebraic geometry codes [19].

On the other hand, "Probabilistic coding theory" does not emphasis on the mathematical aspect of code and their minimum Hamming distance. This theory more concern with the searching of codes that have good performing codes and less encoding/decoding complexity. Usually, decoders for such codes use probabilistic information for decoding algorithms. Modern channel codes have superior performance and more close to Shannon capacity by using this branch of channel coding theory. Major adopted probabilistic codes are turbo codes [20] and LDPC codes [1]. It is important to mention turbo codes based on parallel working of two convolutional codes. Convolutional codes [21] developed by Elias having algebraic sympathetic. This structured based research work contributes one of the most influences in the modern information theory. Most of later work based on Elias studies ended with breakthrough algorithms such as Viterbi algorithm [22] and BCJR algorithm [23]. Another work proposed by Elias before his invention of convolutional codes was product codes [24], based on product of two linear block codes decoded,

sub-optimally, by decoding the two codes distinctly. Later, Forney [21] proposed concept of concatenated codes, where two smaller linear block codes, the outer and the inner codes are serially concatenating. Furthermore, concatenating has also been smeared to the convolutional codes. In addition, after invention of Turbo codes [20] in 1993, researchers and telecommunication industry reawakened the interest in modern coding theory and hence leads to renaissance of LDPC codes in the late 90's.

2.2 Linear Block Codes

FEC has always been a hot topic of research in recent years. In fixed block-length, linear block codes usually work for error correction. It has an extensive range of applications in wireless communication systems. In linear block codes, input bit streams are formed into message blocks of fixed size. A (*n*,*k*) block code defined as a binary code of length *n* with 2^k codewords and the entire block have *k* information bits. Hence, there are 2^k different messages, named as 2^k codewords. 2^k codewords will form a block code of size (*n*,*k*). Block encoder adds $n-k$ bits to the original message, which are named as parity bits. Parity bits are elaborated in identifying and correcting errors caused by the channel noise or interference. *k* bits added to each message by the channel encoder are called *redundant bits*. The function of these redundant bits is providing error detection and correction capability for the channel code [25]. Code rate is defined as $R = k/n$.

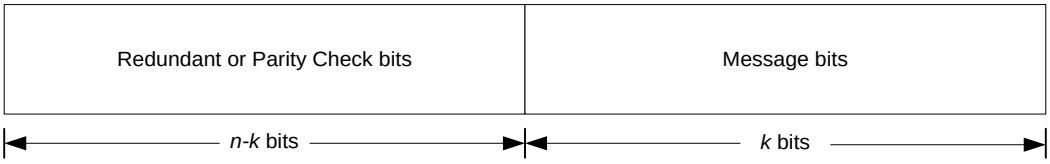


Figure 2-2: Systematic format of a codeword

Linear block code are those having a sum of two codewords is also a codeword with modulo-2 addition (for binary codes), and called basis of the space because it leads to span all code vectors in space. These codes reduce complexity of encoding by using a generator matrix (**G**) to transfer input

message bit stream into codewords. **G** matrix having linearly independent row vectors of size $n, r_1, r_2, \cdots r_k$, can be represented as-

$$\mathbf{G} = \begin{bmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_k \end{bmatrix}$$

At the encoder, codewords generates by multiplying the input vector with the generator matrix, $c = m\mathbf{G}$, where m is input message bit-streams and c denotes codeword. A simple encoding requires the encoder to store all combinations but linear block codes only store **G**, thus reducing the space complexity from $2^k \times n$ to size of **G** i.e. $k \times n$. A parity-check matrix **H** can be obtained from **G** matrix by using row-column operation in systematic form of $\mathbf{G} = [\mathbf{I}_k; \mathbf{H}^T] = 0$. These two matrices is related by

$$\mathbf{G}\mathbf{H}^T = 0$$

The decoder receives a sequence of bits incorporated with noise can be defined as *y*, valid codewords can be checked and passed to further decoding blocks after checking following condition

$$y\mathbf{H}^T = 0$$

2.3 Reviews of binary LDPC codes

In this section, the relevant background on LDPC codes will be provided, including LDPC code structure and representation and decoding algorithms.

2.3.1 Representation of LDPC Codes

LDPC codes can be represented in two basic forms. One of these is in the form matrix and another one illustrative way of presentation is graphical form and most commonly popular as Tanner graph.

2.3.2 Code construction

A low-density parity-check (LDPC) code, a class of linear block codes is defined by a parity-check matrix (**H**) that is sparse. A regular (j,k), **H** matrix is a $M \times N$ binary matrix having *j* ones in each column and exactly *k* ones in each row, where $j < k$ and (j,k) are small as compared to *N* for sparse matrices. The code rate will be $R = 1-(M/N)$ which can equivalent then to $R = 1-(j/k)$, assuming the *M* rows are linearly independent.

An irregular LDPC matrix is also sparse, but in this case not all columns and rows have same number of one's as constant value. The matrix defined in example given in equation (2.1) is parity check matrix of dimension $M \times N$ for a (4, 8) code.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{2.1}$$

LDPC codes are linear codes obtained from sparse bipartite graphs. Suppose that G is a graph with *n* left nodes (called message nodes) and *r* right nodes (called check nodes). The graph gives rise to a linear code of block length *n* and dimension at least $n-r$ in the following way: The *n* coordinates of the codewords are associated with the *n* message nodes. The codewords are those vectors $c_1, c_2, \cdots, c_n$ such that for all check nodes the sum of the neighboring positions among the message nodes is zero. Figure 2-3 gives an example **H** matrix represented as graphical form more commonly known as Tanner graph. However, not every binary linear code has a representation by a sparse bipartite graph, if it does, then the code is called a low-density parity- check codes.

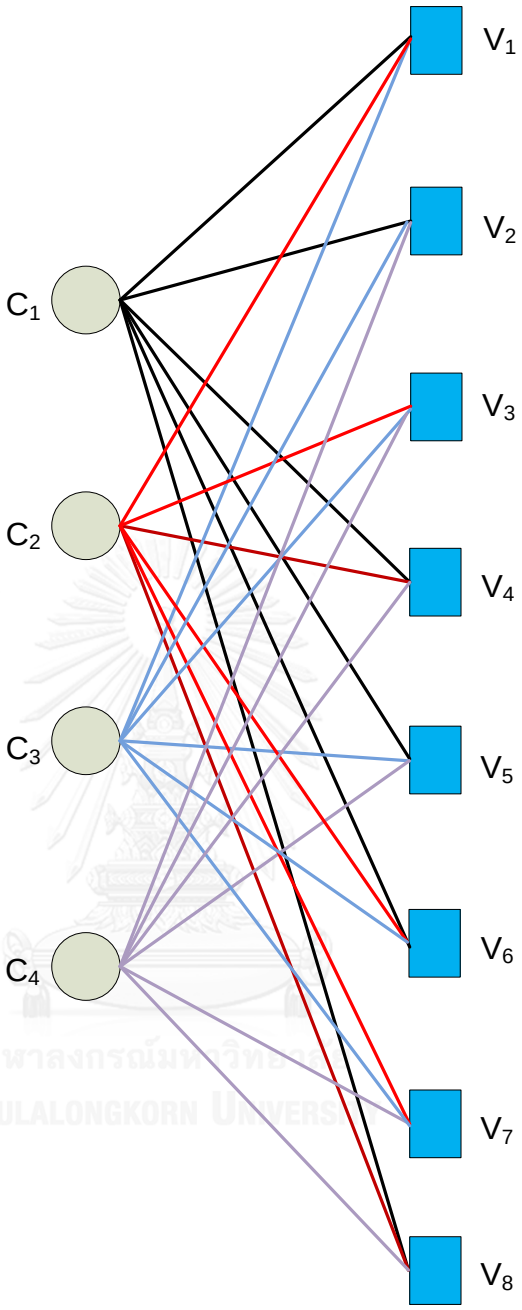


Figure 2-3: Bipartite tanner graph of **H** matrix

2.3.3 Tanner graphs

As mentioned in Section 2.2, a Tanner graph is used to represent the relationship between the codeword bits and parity check bits of a linear block code. Tanner graphs have been generalized to become factor graphs [26]. Variable nodes j and check nodes i connected by edges if the entry in $\mathbf{H}(i, j)$ is 1. Effetely, **H** matrix is also known as bi-adjacency matrix of Tanner graph.

2.4 LDPC code parameters

2.4.1 Cycle

A cycle in a Tanner graph is a sequence of connected symbol nodes and parity-check nodes that begin and end at the same node and no other nodes can appear in the order more than once.

2.4.2 Length

The length of a cycle is the number of edges reachable in the given cycle.

2.4.3 Girth

The girth of a Tanner graph is defined as the length of shortest cycle present in the Tanner graph.

2.4.4 Degree

The degree of a symbol node or check node in the Tanner graph is the number of edges linked to it.

2.5 Decoding Algorithms: Belief Propagation

We would like to start by describing a more common class of decoding algorithms for binary LDPC codes. In order to define two categories of decoding, one is hard decision decoding for only educational purpose to get an overview of an idea of LDPC decoder. Another and most common useful category is soft decision decoding algorithms. These algorithms are called message passing (MP) or sum-product iterative algorithms. The reason for iterative in nature is that at each stage of algorithms, messages are passed from the bit or variable nodes to the check nodes, and from check nodes to message nodes back and forth. The information from variable nodes to check nodes is computed based on the observed or received value of variable node and some of the check nodes associated with corresponding message

nodes. An important aspect is that the information sent from a variable node (v) to a check node (c) must not be taken account the information sent in previous rounds of iteration from c to v nodes, the same statement hold for information passed from check nodes to variable nodes.

Furthermore, one important subclass of message passing algorithms is the belief propagation algorithm. This work was proposed by Gallager's work [1], and it is also useful in various wireless applications [27, 28]. It is observed that iterative decoding algorithms of sparse codes perform very close to the optimal maximum likelihood decoder.

2.5.1 Hard Decision Decoding

Hard decision decoding also known as bit-filliping algorithm deals with flipping bits into 1 or 0. In this type of decoding, variable nodes sends message to its connected check nodes having value 1 or 0 and then check nodes give response to the variable nodes according to the **H** matrix by checks of parity bits equations in either 1 or 0. Let us consider BSC, where messages were passed in binary, in this case the lowest level of discretization is achieved. In this section, we would like to describe types of hard decision decoding based on the BSC, both given by Gallager work [1]. In both the cases, the message passed between the variable nodes and the check nodes consist of binary 0 and 1. In the Gallager algorithm, following are the steps of hard decision algorithm:

2.5.1.1 Steps of hard decision algorithm

Step 1 Check node update:

Variable nodes (v) send their received value to all their neighboring check nodes (c), this sent information belief to be correct even in case error caused due to transmission channel.

Step 2 Variable node update:

In the second step, every check node c calculates its response to connected variable nodes by modulo sum of information bits, incident from variable nodes (except form which the information belief to be sent) to check nodes c. The response

message contains the bit that check node *c* believes to be correct one for corresponding variable nodes *v*. Important point in this step might also be said as terminates. This case occurs if all check equations fulfilled.

Step 3 next phase:

Variable nodes receive the messages from check nodes and use this additional information to decide the originally receive bit from the channel is correct or not. A simple way to do this is a majority voting that consist of original bit received and number of suggestions depends on each variable node connect on how many check nodes. Now the *v* nodes can send another message to corresponding check nodes *c*.

Step 4 Go to step 2:

Iteration continues till decoding process not completed for certain number of fixed iterations, i.e. all parity-check equations and transmission's errors are not rectified.

Although the algorithm is very simple, the major drawback is that all the valuable information such as noise etc., is disregarded by the decoder. Hard decision decoding is used only for study point of view to understand the decoding process however practical implementation has soft decision decoding algorithms.

2.5.2 Soft decision decoding

Soft decision of LDPC codes based on belief propagation, also known as sum product algorithm (SPA) results in better decoding performance and most preferable method for decoding. Demonstrated idea is same as the hard decision decoding before discussing lets introduce some notations:

$\mathbf{H}$ = Sparse parity-check matrix

h_{ij} = Elements of $\mathbf{H}$ matrix

$\mathbf{H} = [\mathbf{P}^T; \mathbf{I}]$

$\mathbf{G} = [\mathbf{I}; \mathbf{P}]$

c_i = Elements in error free transmitted codeword

y_i = Elements of erroneous received codeword

$P_i = P_r(c_i = 1 \mid y_i)$

q_{ij} = Message sent by variable node v_i to check node c_j. Every message contains $q_{ij}(0)$ and $q_{ij}(1)$, which stand for the amount of belief that y_i is a "0" or a "1".

r_{ji} = Message sent by the check node c_j to the variable node v_i. Again there is $r_{ji}(0)$ and $r_{ji}(1)$ indicates the current belief of information "0" and "1" respectively.

The following rounds of the soft decision algorithm is based same as above discussed hard decision decoding algorithm.

2.5.2.1 Steps of soft decision algorithm

Step 1: check node update:

All variable nodes sent their q_{ij} messages to the corresponding check nodes. $q_{ij}(1) = p_i$ and $q_{ij}(0) = 1 - p_i$.

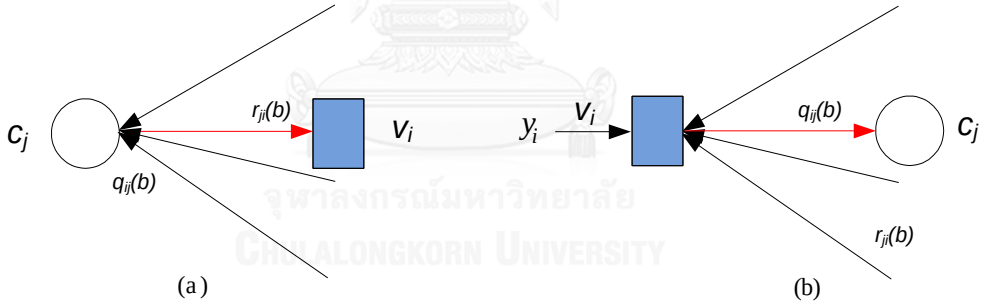


Figure 2-4: Illustrate the calculation of $r_{ji}(b)$ in (a) and in (b) calculation of $q_{ij}(b)$

Step 2: variable node update:

The check nodes calculate their information r_{ji} from the following equation

$$r_{ji}(0) = \frac{1}{2} + \frac{1}{2}\prod_{i' \in v_j \setminus i}(1 - 2q_{i'j}(1)) \qquad (2.2)$$

and

$$r_{ji}(1) = (1 - r_{ji}(0)) \qquad (2.3)$$

So this step calculates the probability of even number of one's sequence calculated by Gallager formula among the variable nodes except v_i. This probability denoted as $r_{ji}(0)$ that v_i belief to be zero.

Step 3: Check node update:

In this step, updated variable node sent their response to the check nodes. This will be done according to following equations

$$q_{ij}(0) = K_{ij}(1-p_i)\prod_{j'\in c_i\setminus j}(r_{i'j}(0)) \tag{2.4}$$

$$q_{ij}(1) = K_{ij}p_i\prod_{j'\in c_i\setminus j}(r_{i'j}(1)) \tag{2.5}$$

Where by constants K_{ij} are chosen in such a way to ensure that $q_{ij}(0)+q_{ij}(1)=1$ and $j'\in C_i\setminus j$ means all check nodes except c_j as scenario illustrates in Figure 2-4. At this stage variable node also update their current estimation $\hat{v}_i$ of their respective variable v_i. This is calculating by probabilities for 0 and 1 and voting for bigger estimates. The used equations are as follows are quite similar to those used for calculating $q_{ij}(\mathrm{b})$ but now the information of every check nodes is used.

$$Q_i(0)=K_i(1-p_i)\prod_{j\in c_i}r_{ji}(0) \tag{2.6}$$

and

$$Q_i(1)=K_i(p_i)\prod_{j\in c_i}r_{ji}(1) \tag{2.7}$$

$$\hat{v}_i=\begin{cases}1 & \text{if } Q_i(1)>Q_i(0),\\ 0 & \text{else}\end{cases} \tag{2.9}$$

If the current estimated codeword fulfills parity-check equations at this stage, the algorithm terminates else termination of algorithm would be ensured through a higher number of iterations.

Step 4: Go to *step 2* for next round of iteration

The above discussed soft decision algorithm is a very simple variant and could be modified for better performance. Besides performance issues there is numerical stability crisis due to many multiplications of probabilities. The result will close to zero if the large block length code is used.

2.6 Iterative Message Passing Algorithms

Iterative algorithms operate on Tanner graphs of LDPC codes. The algorithm computes the nodes of the graph and passes them along the edges to the adjacent

nodes. Messages are basically the probabilities or other values derived from probabilities.

In this thesis, our assumption is that a binary code word $(c_1, c_2, \cdots, c_n)$ is transmitted using the binary phase-shift keying (BPSK) modulation. The channel incorporated with the additive white Gaussian Noise (AWGN), and the received symbols from the channel are $(y_1, y_2, \cdots, y_n)$. It should be noted that throughout this thesis we used the notion for the average energy per information bit as (E_b) and one sided power spectral density (PSD) of the AWGN channel as N_0 respectively.

In addition, it is more convenient to use *log likelihood ratios* (LLR) to represent the messages, because by using LLRs, division and multiplication in probability domain becomes subtraction and addition in log domain. We use r_{ji} to denote the LLR message sent by the check node *j* to the variable node *i*, and q_{ij} to denote the LLR message sent by the variable node *i* to the check node *j*. There are two well-known message passing decoders namely sum-product decoder and min-sum decoder. In addition, there is extensive research devoted on behavior of message passing decoders. It is worth mentioning powerful decoders such as extrinsic information transfer (EXIT) charts [29] and density evolution [30] were proposed to examine the asymptotic performance of LDPC codes on a memory less channel. It was established that performance depends on the degree distribution of the given Tanner graph. It should be noted that the performance of SPA optimal for cycle free Tanner graph in terms of bit error rate probability and frame or block error probability. One more suggested method to analyze the performance of LDPC codes on the basis of computation trees was proposed by Wiberg [31]. However, range of computation trees increases while increasing the degree of the check nodes. Although, it is widely accepted that the performance of message passing decoding algorithm Here we summarizing the SPA and log domain algorithm in next sub section.

2.6.1 Message passing algorithm (MPA) or SPA using log domain

The message passing algorithm imposts the calculation of likelihoods by using the knowledge of before an event (intrinsic) and after an event (extrinsic). Let us define a variable *z*, there are different types of probabilities to express *z* relation with

an event *E*. The a-priori probability of *z* is equal to *a* with respect to the event *E* is defined as

$$P_E^{priori}(z = a) = P(z = a). \tag{2.10}$$

This probability called as a-priori because it indicates to what was known about *z* before observing the outcome of an event *E*. On contrast a posteriori probability of variable *z* with respect to the event *E* is the conditional probability of *z* given that event already occurred, and is represented by

$$P_E^{posteriori}(z = a) = P(z = a \mid E) \tag{2.11}$$

The above probability represents what is already known about the *z* after witnessing an event *E*. After applying Bayes theorem, the a posteriori probability can be written as

$$P(z = a \mid E) = \frac{1}{P(E)} P(E \mid z = a) P(z = a) \tag{2.12}$$

From above equation the extrinsic probability can be deprived as

$$P_E^{ext}(z = a) = dP(E \mid z = a), \tag{2.13}$$

Where *d* is a normalization constant such that $P_E^{ext} = 1$. Henceforth the relationship between P_E^{ext}, $P_E^{posteriori}$ and P_E^{priori} can be written as

$$P_E^{posteriori}(z = a) = P_E^{priori}(z = a) P_E^{ext}(z = a) \tag{2.14}$$

For the binary case, $z = [0,1]$, the probability of binary variable *z* can be expressed conveniently in term of real number called the log-likelihood ratio (LLR). If we assume $P(z = 1) = p$, then the LLR of *z* is expressed as

$$\mathrm{LLR}(z) = \log \frac{P(z = 1)}{P(z = 0)} = \log \frac{p}{1 - p} \tag{2.15}$$

If $p \geq 0.5$, $\mathrm{LLR}(z)$ is positive and $\mathrm{LLR}(z)$ is negative if $p \geq 0.5$.

$P(z = a \mid E) = \frac{1}{P(E)} P(E \mid z = a) P(z = a)$ can be re-written in terms of LLR as

$$\mathrm{LLR}_E^{posteriori}(z) = \mathrm{LLR}_E^{priori}(z) + \mathrm{LLR}_E^{ext}(z) \tag{2.16}$$

The extrinsic information signifies the incremental gain in evidence of a posteriori information over a priori information. The SPA depends upon priori, extrinsic and posteriori probabilities.

The channel and nodes are used to obtain priori information and extrinsic information respectively. Equations/steps for calculation the probabilities in log domain are mentioned below.

2.6.1.1 Steps of algorithm

Step 1 Initialization:

Each variable node *v* initialized with the received information y_i, from the channel. There-after each variable node *v* calculates the initial LLR, as follows:

$$L(z_n) = \ln\left\{\frac{P(z_n = 1 \mid y_n)}{P(z_n = 0 \mid y_n)}\right\}$$

We assume that channel is an AWGN,

$$L(z_n) = 2y_n / \sigma^2,$$

Where σ^2 is the variance of the AWGN and y_n is the information received from the channel. $L(z_n)$ is the log domain probability for the transmitted bit z_n is 1 or 0, given the received bit y_n. For each variable node, initial LLR $L(z_n)$ and messages along the edges in the Tanner graph both are set to be zero. Figure 2-4 shows the schematic diagram of message passing in SPA, same applies to the LLR domain version of SPA. Incoming and outgoing messages exchanged between check and variable nodes.

Step 2: Check node update

Calculate LLR for each check node, *c* and information sent from check nodes to the variable nodes based on the information coming from variable nodes. The check node LLR is given by

$$\lambda_c = \sum_{all-msgs} \ln\left\{\tanh\left(\frac{\text{abs}(q_{ij})}{2}\right)\right\} \tag{2.17}$$

The messages going out from check nodes to variable nodes are given by

$$\Lambda_{c,v} = 2 \times \text{atanh}\left\{\exp\left\{\ln(\lambda_c) - \ln(\tanh(\frac{q_{ij}}{2}))\right\}\right\} \tag{2.18}$$

λ_c is given by exclusive-or (XOR) of all the messages coming from variable nodes and $\Lambda_{c,v}$ is given by the AND operation of sign of λ_c and sign of the corresponding q_{ij} [32].

Step 3 Variable node update

For each variable node, *v*, calculate LLR information which will pass along its edges to the corresponding check nodes. The LLR is given by

$$\lambda_v = L(z_c) + \sum_{all-msgs} \Lambda_{c,v} , \tag{2.19}$$

where $\Lambda_{c,v}$ represents a message sent from check node to variable node. LLR is the sum of all incoming messages along with the initial value of the variable node. The messages moves to check nodes are given by

$$q_{ij} = \lambda_v - \Lambda_{c,v} \tag{2.20}$$

The departing message for each edge is given by the check node LLR minus the message obtain on that edge.

Step 4 Decision

Assign the LLR of variable nodes such that $\mathrm{LLR}_v = 0 \text{ if } \lambda_v < 0$ and $\mathrm{LLR}_v = 1$ if $\lambda_v \geq 1$. LLR provides the estimation of the codeword c_n, if $\mathrm{LLR} \times \mathbf{H}^T = 0$, then stop the algorithm with output same as LLR. Otherwise go to step 2 for next round of iteration. In case algorithm does not stop with in certain number of iterations, it is consider to be decoder failure.

The assumption laying this algorithm and justifying the equations is that the messages are statistically independent over the decoding process. Assumed that y_i*'s* are independent, former assumption holds true. For a given girth *g*, the independence assumption is only true up to the $g/2^{\text{th}}$ iterations, after which message starts to loop back. Numerical error is introduced in calculating a posteriori probability with the presence of cycle in the Tanner graph. An output of the decoder is hard decision for most likelihood transmitted codeword, so as long as error is introduced due to the presence of cycle, it can be ignored as final decision is based on the hard decision value of decoded output. It is assumed that if the good code is used the error's value is small. However, simulations and results also show that belief propagation algorithm is

considered to be the most efficient decoding algorithm in terms of BER and used widely in practical applications. All the steps required in this algorithm needs only local probabilities, which can be computed by each nodes locally, this helps to reduce the computational effort in to the small pieces that can be calculated in parallel.

2.7 Definitions used in designing of LDPC codes

LDPC code can be designed by determining the fundamental parameters of a code for example code rate and code size. These properties are often defined by considering the desired applications. Following are brief description of some of these parameters and their effect on performance and hardware implementation.

2.7.1 Code Size

The parity-check matrix of size $M \times N$ states the dimensions of the code size. Usually, code size or codeword is used to refer as *N*. Generally a code is represented by having its codeword size, row and column weights in the form of (N, j, k).*M* can be calculated from the other parameters (N, j, k). It has been shown that large block-length codewords are performed better than short length codewords [2, 5]. Therefore, to achieve good performance long codewords are chosen. Nevertheless, their hardware implementation requires more memory size and hence costly.

2.7.2 Code weight and Code rate

The rate of a codeword is denoted as *R*, is a number of information bits transmitted over the total number of codeword bits. It is expressed as $1-(j/k)$. As more number of row and column weights more will be its computation at each node because of increased number of incoming and outgoing messages. Nonetheless, if more nodes contribute in computation of the probabilities of a bit, the corresponding node converges faster in its outcome. Fewer redundant bits are used for high code rate in a given codeword, i.e. more information can be transmitted per block for higher throughput. However, low redundancy means less number of parity-bits and hence less decoding performance or high BER [32]. On the other hand, low code rates have more redundant bits with fewer throughputs. In this case better decoding performance

can be achieved due to large number of redundant bits. However, very low code rates also have poor performance with a small number of connections in the Tanner graph.

LDPC codes having column-weight of two have their minimum distance (described in the next section) increase in the logarithmic scale with code size as compare to a linear increase for codes with high column weights [1]. Consequently, a column-weight two codes perform below as compared to the higher column-weight codewords. Therefore, column-weight higher than two is generally used. In addition, carefully constructed irregular codes have better error correction performance as compared to the regular LDPC codes [33, 34].

2.7.3 Code configuration

The arrangements of the codes are determined by the connections between the rows and columns, i.e. check nodes and variable nodes. The connection pattern decides the complexity of the interconnection between check nodes and variable nodes i.e. in an encoder and decoder implementations there are two types of code constructions, random codes and structures codes. Random codes do not constitute any predefined connections between row-column of parity check matrix. On the other hand structure codes have predefined connections between row-column.

2.7.4 Number of Equations

The number of equations is defined as the number of times the received bits are estimated before a final hard decision is made by using the decoding algorithm. A large number of iterations may be required to converge the decoding algorithm, on the other hand, it may increase the delay time in decoding and power consumption. In simulation performance a large number of iterations may be used for more perfection. In general, for practical applications 10 to 40 equations are used [35].

2.8 Parameters of optimization of LDPC Codes

For improving the decoding performance of the LDPC codes, we need to optimize the several parameters. Mainly these parameters are girth, minimum distance. The improvement in performance also depends on the decoding technique

and on the **H** matrix. There are also performance measures to determine how good a code is in correcting errors with optimizing value. Below are some of the common techniques used for optimization found in the literature.

2.8.1 Minimum distance

The count of number of 1's in the codeword is known as the Hamming weight of a codeword. The hamming distance is defined as the number of bits with which the codeword defer from each other and the minimum distance of a code is the smallest hamming distance between the two codewords. The higher the hamming distance, the better the performance of a code. The better code can be determined by using a concept of minimum distance. A large block-length and high girth LDPC codes are likely to have larger minimum hamming distance [2]. In the case of randomly constructed codes, there cannot be any algorithm so far known to calculate the minimum distance accurately. This problem was investigated in [36]. On the other hand, for the structured codes, some researchers calculated minimum distance by using software tool such as MAGMA [37].

2.8.2 Girth

Girth will affect the decoding performance of a given code. Large girth improves the code performance whereas small ones especially of length four drastically degrades the decoding performance. Many researchers use the term "no-cycle" in the Tanner graph which means to avoid the cycle of four. With small cycles, a node gets a probability estimate including its own contribution after a few iterations, in case this node has received the wrong information so it will calculate and propagates a wrong probability estimates to the connecting nodes. In case of large girth, the estimates are less dependent on the node's connection for higher number of iterations, which is the assumption of the SPA algorithm. The most cited result for girth estimation and their performance comparison given by Sullivan [38]. It is shown that large girth codes perform better than shorter girth codes. In addition, distribution of local girth also contributes to the performance evaluation as in Mao [39], sometimes even more important than the overall girth estimates of the given code. A

code with a large average girth is likely to outperform a lower average girth of same girth code. In this thesis, we proposed new construction algorithms for obtaining large girth codes and discussed in Chapters 4 and 5.

2.8.3 Stopping sets

A stopping set *S* is a subset of *V*, the set of variable nodes, $S \in V$, such that all neighboring nodes of the variable nodes in *S* are connected to *S* at least twice. The size of *S* is defined as the cardinality of *S.* Prevention of small stopping sets has been proved to improve the BER performance and improving minimum distance as in [40].

2.8.4 Density Evolution

Density evolution is a kind of belief propagation algorithm with messages as probability density function, despite of LLR messages. This algorithm determines the probability density function of the messages through the graph node assuming that cycle free condition is verified [41]. The bit error probability can be made arbitrarily small as the code length size tends to very high, if the noise level is smaller than some constant threshold. In this, by noticing the density of messages between nodes, the performance of the codes can be estimated.

2.6.5 Bit-Error Rate

Above mentioned parameters could be used as a measure for decoding the performance, they do not show how much error can be corrected for the given code. As well as having a higher girth or average girth does not guarantee the better performance. Therefore, LDPC codes are generally evaluated using bit-error rate (BER) performance over a given channel and the type of modulation. In this thesis, all QC-LDPC code performance was performed by considering AWGN channel with binary phase shift keying (DPSK) modulation.

Channels are termed by a mathematical model making it easy to design appropriate modulations and coding schemes. The AWGN channel is the simplest channel model, having a vector of transmitted bits, *c*, to noise vector. The amount of noise at any given time instant can be defined by a normal distribution variable, *n*,

such that the channel bits are $y_i = c_i + n_i$. Uncertainty of the Gaussian noise has a one-sided power spectral density N_o, which depends on the variance σ^2, can be expressed as $N_o = 2\sigma^2$.

The BER measures how many number of errors in transmitted information bits and decoded bits found per iterations for a given signal-to-noise (SNR) ratio. The SNR ratio defined as the absolute power ratio of the signal (transmitted data) and the noise power spectral density. The higher the value of SNR, more signal power than the noise, on the other hand low SNR means to noise level approximately close to the signal. BER of the channel is expressed as follows

$$\text{BER} = \frac{\text{number of errors}}{\text{number of bits}}$$

The number of errors effectively decreases with increasing SNR. SNR can be expressed as $\text{SNR} = 10\log\frac{E_b}{N_o}$, where E_b is the signal energy. The simulation is performed several times for achieving considerably low BER for a given SNR. The BER curve shows the probability that a bit, after decoding will be in error at a given SNR. For example BER value of 10^{-6} means, there is a 1 bit error in 1000000 bits. Another measure for calculating error rate is Word error rate (WER) or Frame error rate (FER). WER is the number of decoded words (length of the codeword) that contains error as the fraction of the total number of words decoded; sometimes FER is preferred over BER.

2.9 Summary

Low-density parity-check (LDPC) code is an active area of research in the last decade and these codes have massive potential in the domain of wireless communication systems. The iterative decoding approach is already used in turbo codes, but the construction of the LDPC codes give even better results with less complex decoding algorithm. In numerous cases, LDPC codes allow a higher code rate and a lower error floor performance as well. Moreover, these codes make it possible to implement parallelizable decoders. The main difficulties are that the

encoders are somehow more complex and that the code length has to be rather long enough to yield better results.

Chapter 3: LDPC Code Constructions Techniques

3.1 Construction of LDPC Codes

In this section, we discuss various aspects of LDPC code construction techniques. Most challenging research incorporated to find a wide range of codes in block-length and code rate that have optimized performance and suits to practical aspects of hardware deployment. LDPC code constructions require the set patterns or connections between the check nodes and variable nodes in order to get short cycle for corresponding Tanner graph. Before constructing codes, it is essential to fix some important parameters such as row and column weights, code length, code rate and girth. Ultimate objective of the code construction to obtain good decoding performance with economical hardware implementation and less computational complexity. However, by considering constraint of low cost hardware deployment may leads to degrade in error performance. There are many ways of code constructions for a given code length and rate. Nevertheless, developed techniques often have certain limitations in order to meet desired constraints.

LDPC code construction methods can be either random or structured connections between check nodes and variable nodes. Codes based on random construction methods may be easy to design in terms of flexibility but at the same time lacking with uniformity of row and column connections which increase the decoder interconnections complexity. On the other hand, codes based on the structured constructions methods have set patterns of check nodes and variable nodes connection but also have limit of given code rate, length and girth. There is still a lot of scope in order to optimize the structured code construction techniques for good error correcting performance and remarkable cost cut for hardware implementation. Generally discourse, design techniques to construct parity-check matrices of LDPC codes fall into two main groups: computer-based and algebraic methods. The algebraic approach often involves finite mathematics, [38, 42, 43], or combinatorial

techniques, [11, 44-48], which are promising for industrial applications thanks to the simple encoding structures. On the other hand, computer-based techniques, including Gallager codes [1], MacKay codes [2] and density evolution (DE), [30] and Progressive edge growth (PEG) algorithm [49], are still predominant as those random constructions are highly flexible in their code design and can offer near-capacity performance with very large block lengths. In this section, we will present some of the most important computer- based techniques.

In this chapter, we review some construction methods of regular and irregular LDPC codes related to our proposed methods.

3.2 Randomly Constructions of LDPC Codes

3.2.1 Gallager Codes

Gallager first proposed regular LDPC codes in his doctoral dissertation thesis in 1962, [1] with three parameters (N, w_c , w_r) to denote the code length, the number of 1's in each column, and the number of 1's in each row, respectively. In his method, there are random connections between check and variable nodes of a LDPC code without any predefined arrangements. An **H** matrix for Gallager code is constructed by random column permutations, and has the following structure $\mathbf{H} = [\mathbf{H}_1 \, \mathbf{H}_2 \quad \cdots \quad \mathbf{H}_{w_c}]'$. The sub matrices formed by column permutations of $\mathbf{H_1}$ having constraint as in [1] based on no 4 cycle existence.

3.2.2 MacKay Codes

An alternate construction scheme for LDPC codes was devised by MacKay [2] while apparently not being conscious of Gallager codes. The method illustrates the benefits of designing codes with sparse **H** matrices, and interestingly for the very first time it demonstrate the capability of LDPC codes to perform near capacity limits [4]. Mackay developed few random constructed methods based on constraints to generate an **H** matrix of LDPC codes. Firstly, all zero **H** matrices generated with column weight 'w_c' and then randomly bit flipping in the matrix. On the foundation of Tanner graph, MacKay codes enforces an important fundamental property on **H** matrix that there should not be any two rows or two columns have

more than one position in common that contains a 1 element, that is referred to as the row-column constraint [44]. During iterative decoding, if two variable nodes share in two corrupt parity-check equations instantaneously, it is not possible to detect the corrupted bits and further can correct them. The row-column constraint eliminates short cycles of length 4 in Tanner graph since the existence of such cycles significantly degrades the performance of iterative decoding algorithms. Furthermore Mackay work used to find good error performing LDPC codes with different code length and rates. Some of the code words even enlist in World Wide Web database as in [50].

3.2.3 Progressive Edge Growth Algorithm

Progressive Edge Growth algorithm (PEG) was firstly presented by Hu *et al.* [49] that allows the construction of regular or irregular LDPC codes with high girth. It is the greediest algorithm to achieve high girth value locally at each edge placement. PEG algorithm works on variable node to variable node connections. The first edge connection at each variable node is made by considering that random selection of check node is done under the condition of minimum weight with the current draft setting. It is considered as most successful approaches for construction of finite length LDPC codes. In practice, a low cycle free Tanner graph provides optimum decoding and PEG try to maximize the girth cycle. The PEG algorithm can be summarized as follows.

An *M* by *N* parity-check **H** matrix (*M* rows and *N* columns) of an LDPC code can be represented by a Tanner graph [26], where *M* is the number of parity-check equations, *N* is the number of coded bits, and $K = N - M$ is the number of message bits. The Tanner graph is a bipartite graph, which composes of the set (V, E), where $V = V_c \cup V_s$, $V_c = \{c_0, c_1, ..., c_{M-1}\}$ is the set of check nodes, $V_s = \{s_0, s_1, ..., s_{N-1}\}$ is the set of bit nodes, and E is the set of edges, $(c_i, s_j) \in E$ corresponding to a nonzero element at the *i*-th row and the *j*-th column in the **H** matrix, where $0 \le i \le M-1$, and $0 \le j \le N-1$.

Additionally, let the degrees of check and bit nodes be define as $D_c = \{d_{c_0}, d_{c_1}, ..., d_{c_{M-1}}\}$ and $D_s = \{d_{s_0}, d_{s_1}, ..., d_{s_{N-1}}\}$, respectively, where $E_{s_j}^k$ denote the edges on s_j with $0 \le k \le d_{s_j} - 1$. Figure 3-1 shows the Tanner graph for D_s = {2, 2, 2, 2, 2, 2, 2, 2, 2, 2} and D_c = {4, 4, 4, 4, 4}, where the bit □ represents the check node c_i, and the o represents the bit node s_j.

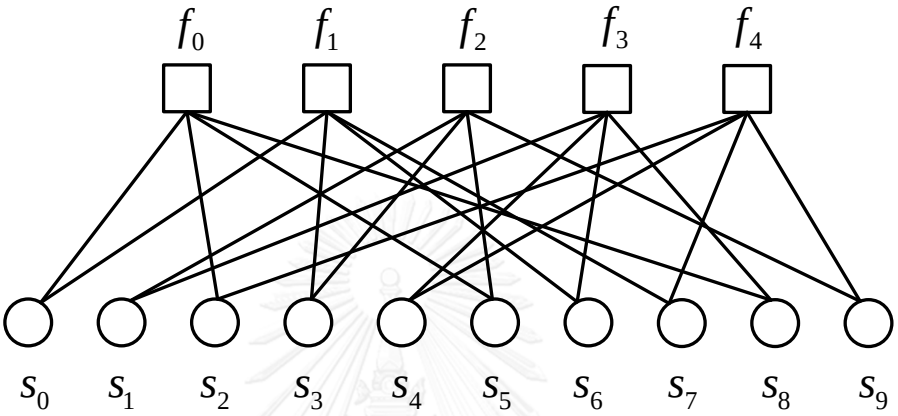


Figure 3-1: A Tanner graph for PEG based **H** matrix

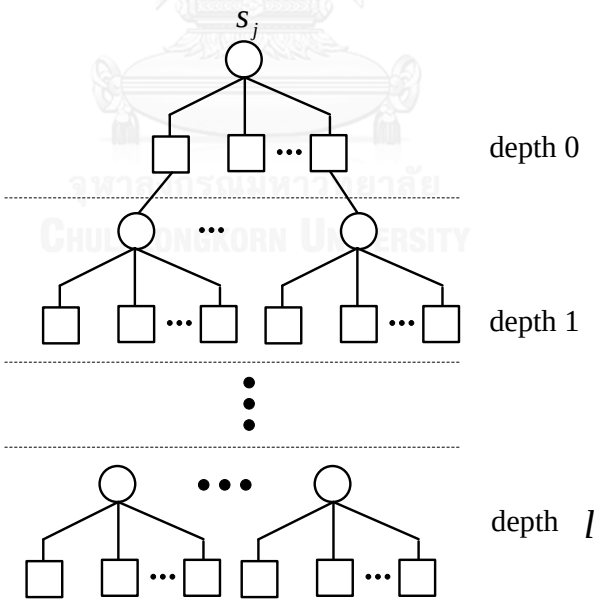


Figure 3-2: A sub-graph spreading from bit node s_j.

3.2.3.1 Parity-Check matrix constructions

This section briefly explains how the PEG algorithm works [49], whose process is to connect an edge between a check node and a bit node by using the

spreading of the sub-graph as depicted in Figure 3-2. For a given bit node s_j, we define its neighbourhood within depth l, $N_{s_j}^l$, as the set consisting of all check nodes reached by sub-graph spreading. Specifically, the first edge that wants to connect to a bit node s_j can be chosen randomly from a check node with the lowest degree. For the next edge, we must first spread the sub-graph from s_j and then select the check node from the lowest depth (at depth l) of this graph that has the lowest degree. However, if the sub-graph does not cover all check nodes, we must choose the check node with the lowest degree that is not within the sub-graph ($\bar{N}_{s_j}^l$). The PEG algorithm can be summarized as follows:

1. Assign the degree of a bit node s_j, e.g., $d_{s_0} = 2$, and the set of edges incident to this bit node as $E_{s_j} = \{E_{s_j}^0, E_{s_j}^1\}$.
2. Add an edge to this bit node s_j. First, if the edge that wants to be added to this bit node s_j is $E_{s_j}^0$, we can choose the set of V_c with the lowest degree randomly. If the edge is not $E_{s_j}^0$, we must expand the sub-graph up to depth l. Then, the two event a) and b) can happened:

   a) Given the set of V_c within depth l denoted as $N_{s_j}^l$, if the number of $N_{s_j}^l$ is less than M, we must choose the set of V_c that is not in depth l, denoted as $\bar{N}_{s_j}^l$, which has the lowest degree randomly; and

   b) If the cardinality of set $N_{s_j}^l$ is equal to M, we will choose the check node with the lowest degree at depth l. Repeat this step until the k-th equal to d_{s_j}
3. Go back to Step 2 for adding edges to the next bit node until $j = N-1$, where $0 \le j \le N-1$.

   Thus, the set of $N_{s_j}^l$ and $\bar{N}_{s_j}^l$ must satisfy $N_{s_j}^l \cup \bar{N}_{s_j}^l = V_c$, where $V_c \setminus N_{s_j}^l = \bar{N}_{s_j}^l$ and $V_c \setminus \bar{N}_{s_j}^l = N_{s_j}^l$. Finally, if we choose any check node at depth l, it can be shown that the number of girths from the bit node s_j will be equal to $2(l+2)$.

Below is a pseudo-code for the PEG algorithm.

For $j = 0$ to $n-1$

For $k = 0$ to $d_{s_j} - 1$

If $k = 0$

$E_{s_j}^0 \leftarrow$ Choose the check node with the degree randomly

Else

$E_{s_j}^k \leftarrow$ Use the above Step 2(a) if the number of

$N_{s_j}^l$ is less than M

Use the Step 2(b) if the number of $\bar{N}_{s_j}^l$ is

equal to M

End

End

End

3.3 Structured Constructions of LDPC Codes

Structured construction methods can be used to get a desired or pre-defined pattern by putting constraints on row-column connections. The main goals are to have a connection pattern that is easier for hardware implementation and to achieve good performance. There are many structured methods already developed that includes those based on graphs, algebra, combinatorial designs and heuristic searching techniques.

3.3.1 Combinatorial Designs

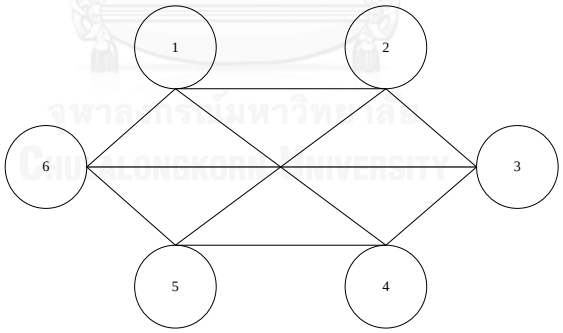
Structured LDPC codes can be generated by constructing parity check matrix using combinatorial mathematics. A combinatorial design is an arrangement of sets of v points into b subsets called blocks, according to some defined constraints.

The two basic constraints used are:

1- A pair of points appear together only in λ blocks for a defined value of λ.

2- The number of points in each block is given by γ, number of blocks in which a point appears is given by ρ [45].

A balanced design can be constructed if the covalency λ, is the same for all pair of points, from the definition, the dimensions of the code is given by $(v \times b)$. We can denote column weight (w_c) and row weight (w_r) as γ and ρ respectively. The design is consistent if all blocks have same number of points and it appears same number of times. The obtained codes will not have a four cycles as the design ensures this by using covalency of 1 in the first constraint. As per matrix formation, a four-cycle is formed by having a pair of rows connected to the same columns or having a pair of columns connected to the same rows more than once. The first constraint i.e. $\lambda = 1$ breaks this condition as two points appears in the same block only once. This constraint is known as row-column (RC) constraint [9]. RC constraint is used in many methods in different forms to avoid girth of four. Figure 3-3 and Table 3-1 shows combinatorial design with row weight three and column weight two. Part (a) is a subset of division of points. The graph beside shows edges as blocks and points as vertices. The resulted matrix forms a LDPC code where columns and rows are connected in case they do not belong to the same block. Part (b) of the figure shows the resulting matrix from the graph. Adjacency matrix shows connection of



Points {1, 2, 3, 4, 5, 6}
Blocks [1, 2], [1, 4], [1, 6]
[3, 2], [3, 4], [3, 6]
[5, 2] [5, 4], [5, 6]

Figure 3-3: (a) Combinatorial design graph (b) Points and subset arrangements using design graph as in (a)

Table 3-1: Incidence matrix

Points	*Connections to points based on design graph*								
1	1	1	1	0	0	0	0	0	0
2	1	0	0	1	0	0	1	0	0
3	0	0	0	1	1	1	0	0	0
4	0	1	0	0	1	0	0	1	0
5	0	0	0	0	0	0	1	1	1
6	0	0	1	0	0	1	0	0	1

vertex in given graph. This method was used in [51] to construct column weight two codes having girth eight. The given size of code was $2k \times k^2$, where k is same as ρ. To obtain higher column weight combinatorial design, the same method with different polygons can be applied to construct different codes as explained in [9, 45]. It should be noted that codes obtained from above method having RC constraints for only girth four. So in these codes there is girth of six present which will limit the performance of the code. For the medium to large block length codes decoding performance can be enhanced by increasing the girth of the codes. Even though a wide range of code rates and length can be obtained by using combinatorial design by using RC constraint.

3.3.2 Finite Geometry Method

Finite geometries are other approach which can be used for designing structured LDPC Codes. Using Euclidean and projective geometries over finite fields, four groups of LDPC codes are constructed. Their performance is stated to be good with iterative decoding. Moreover, they can have cyclic or "quasi-cyclic" form, therefore they can be encoded easily using simple feedback shift registers.

The four classes of LDPC codes based on Euclidean and Projective geometry are:

Type-I Euclidean geometry (EG) LDPC codes

Type-II Euclidean geometry (EG) LDPC codes

Type-I Projective geometry (PG) LDPC codes

Type-II projective geometry (PG) LDPC codes

Finite geometry is generally defined by n points and J lines with the following properties [52]:

1. Every line contains ρ points.
2. Any two points are connected by only one line.
3. Every point lies on γ lines.
4. Two lines intersect at only one point or they are parallel.

Figure 3-4 represents a finite geometry with $n = 4,\ J = 6,\ \gamma = 3,\ \rho = 2$. The resulted matrix derived from the geometry is also shown in the Table 3-1. Rows represent lines and columns represent points. The intersection of line and point is represented by '1' in the matrix. The incidence matrix can be regarded as a low-density parity-check matrix when $\rho << n$ and $\gamma << J$ as the number of '1' entries will be very small as compared to number of '0' entries. The example matrix with *n* as the block length is called type-I geometry LDPC codes (Euclidean geometry or EG-LDPC codes). The transpose of this matrix will be the matrix with length *J* and is referred as type-II geometry LDPC code (projective geometry or PG-LDPC codes) γ and ρ are column and row weights in type-I codes and row and column weights in type-II codes respectively.

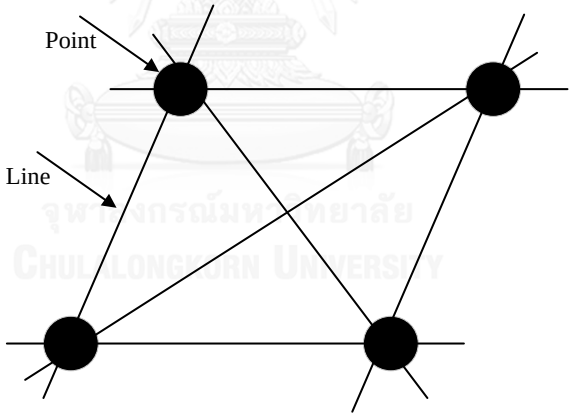


Figure 3-4: Finite geometry with $\gamma = 3$ and $\rho = 2$

Table 3-2: Corresponding incidence matrix

Points	*Connections using finite geometry*			
1	1	1	0	0
2	1	0	1	0
3	1	0	0	1
4	0	1	1	0
5	0	0	1	0

The codes are made regular as per properties 1) and 3) of finite geometry as lines have the same number of points and points are crossed by same number of lines. Obtained codes have a girth of at least six is ensured by property 2 of finite geometry. That means, four-cycles are avoided. However, it was found that these codes will always have a maximum girth of six [53] which is the demerit of this method in case of improving performance by larger girths. In addition, one more drawback of this type of construction is that the resultant **H** matrix will be square matrix of size $N \times N$, therefore, we have to consider $(N-K)$ rows to decode, which will have degraded performance of the code as well as the complexity of the decoding process due to relatively large row and column weights with less flexible structure of the codes. So these designs do not obtain wide range of code lengths and code rates, henceforth, not practically suited. In [52, 53] finite geometry of both the types were constructed and analyzed. Type-I codes have minimum distance of $\gamma+1$ and type-II codes have distance of at least $\rho+1$. In this type of codes the length of the code is calculated by its row weight. It is given as $a(p^{ms}-1)$ with a given row weight of ap^s, where *a, m* and *s* are positive integers and p is a prime number. In addition, he row weight of PG-LDPC codes is $a(p^s+1)$ and code length of $\frac{a(p^{(m+1)s}-1)}{p^s-1}$ as in [53].

3.3.4 Algebraic Methods

Algebraic codes were proposed for the first time in 2001 [53]. These codes can be constructed using finite geometries such as Euclidian and projective geometries over finite fields. In general, these codes results in cyclic or QC LDPC codes. The main advantage of these codes is that it can be easily encoded using shift registers [9]. Much research concludes that algebraic cyclic and QC-LDPC codes can attain excellent overall performance in terms of error floor, computation time complexity and iterative decoding convergence (i.e. average number of iterations needed to converge decoding algorithm). Besides many merits mentioned above these codes do have some major disadvantage, that is the large hardware decoder employment complexity in terms of the number of message processing units, the number of wires connected to corresponding processing units and the amount of memory required for

messages storage. The large decoder hardware employment complexity is primarily caused due to comparatively high density of number of 1's and due to the large of number of redundant rows in **H** matrix of an algebraic cyclic or QC-LDPC codes. This complexity is a critical issue for realizing practical implementation of the applications in the wireless communication system [54].

To obtain these codes with a defined structure, parity-check matrix connections can be algebraically constrained. Constraints can even be used to get any required rate, girth or length. Fossorier [7] offers algebraic constraints to get quasi-cyclic codes of desired girth. The code matrix is divided into sub-matrices of same sizes. The structure of the matrix is given by-

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{a_{11}} & \mathbf{I}_{a_{12}} & \cdots & \mathbf{I}_{a_{1k}} \\ \mathbf{I}_{a_{21}} & \mathbf{I}_{a_{22}} & \cdots & \mathbf{I}_{a_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{a_{j1}} & \mathbf{I}_{a_{j2}} & \cdots & \mathbf{I}_{a_{jk}} \end{bmatrix}, \tag{3.1}$$

Fossorier shows conditions under which cycles are present in a code. Thereafter, the algebraic conditions are used to find codes with girths from six to twelve. To obtain a girth of at least six, a crucial condition is that $\mathbf{I}_{a_{Jjk_1}} \neq \mathbf{I}_{a_{j_2k_1}}$ for $j_1 \neq j_2$ and $\mathbf{I}_{a_{j_1k_1}} \neq \mathbf{I}_{a_{j_1k_2}}$ for $k_1 \neq k_2$. Any four-cycles in the matrix, breaks this condition. A four-cycle matrix contains "1" entries forming a rectangle or a square in a matrix. The author also shows the girth six codes have a length *N* of at least k^2 in case *k* is odd and $k(k+1)$ in case *k* is even. In addition, Fossorier also proves that QC-LDPC codes can achieve up to maximum girth of twelve for any *j* and *k* rows and columns of sub-matrices in **H** matrix. To avoid cycles of given length in a matrix above mathematical constraints can be used. However, the conditions fails to determine the size of sub-matrices or values of shifts in sub-matrices that will work for a given *N*, *j*, and *k*. In order to reduce code search, structured construction method is suggested. The method puts more constraints on the girth conditions so that there is more space to search. The main advantage of methods as in [38, 55] is fewer restrictions are present for the size of sub-matrix group. There are various methods for constructing structured LDPC code in literature. These codes also apply the row-

column constraint to avoid girth of four. Most of the codes are as good as other codes; however, they too have limitations in some or another way.

3.4 Protograph based LDPC codes

The construction of LDPC codes based on protograph begins by having a relatively small bipartite graph from which a large graph can be obtained by copy-and-permute method to obtain a single large graph [56]. This concept has been further studied to build **H** matrices with high girth [57, 58]. Furthermore, the protograpoh LDPC codes have many tactics to optimize the construction of LDPC codes [59]. Protograph based code was studied with the possibility of designing parameters of the code constructed on small base codes. In this process, protograph is copied several times and then the edges are copied and connected for the large single graph based on the same restrictions as on small protograph. After obtaining the single Tanner graph, we can obtain the **H** matrix. If a variable node *V* is connected to a check node *C* in the given protograph, then after copying protograph, variable node *V* can only be connected to one check node *C* of the replicas. Any parallel edges in the main protograph are eliminated in the construction, so as to obtain a single Tanner graph suitable for the constraints of parity-check matrix.

3.5 Benefits of Structured code constructions

LDPC codes based on structured connections usually reduce hardware complexity and minimize cost of encoders and decoders [59, 60]. In structured codes, **H** matrix can be divided into sub-matrices which are mapped onto decoder side with the number of processing same as the number of row and column sub-matrices. The number of processing nodes and interconnections is reduced since there are few sub-matrices compared to rows/columns of **H** matrix. The row-column constraints of structured codes usually have similar behavior. Hardware deployment of these codes can be simplified, that is a group of messages between variable nodes and check nodes can be directed using single row-column constraint. By using the connection rule the source and destination addresses can be calculated in case if messages are stored in memory blocks.

An additional advantage of structured codes is that they could be considered by their performance superiority. The family of structured codes based on construction constraints can be usually obtained. In addition, important parameters such as girth, code rate and minimum hamming distance may be applied to all codes in the family.

However, the performance of structured codes is degraded compared to random codes in the case of comparatively large block-length. Row-column constraints may limit the cycle length, code rate and minimum hamming distance of codeword. If there is performance degradation, it is pretended because of the regularity in the connections between variable nodes and check nodes. Another drawback is that they sometimes exist only for selected parameters such as code rate, column weight and girth combination, which may be unfit for some application. The constraints used plays the important role as the 'structure' and structured code performance depends on it. Nevertheless, structured codes offers best performance and hardware employment tradeoffs as compared to the random LDPC codes. The main contribution of our research is based on constructing structured codes over a wide range of code length, rate and girth.

3.6 Constructing QC-LDPC Codes

Although LDPC codes with large block-length usually provide a good performance but at the cost of huge memory requirement and computation complexity of the **H** matrix. To overcome this problem, Quasi-cyclic LDPC (QC-LDPC) codes were proposed by Fossorier [7], which is based on algebraic and geometric theories and combinatorial designs and mostly accepted form of structured LDPC codes. However, the flexibility of code rate and code length is restricted by the matrix construction theories [4, 9, 46, 61]. Nevertheless, good QC-LDPC codes are well suited for certain practical applications such as data storage systems, DVB-T2/S2, IEEE 802.16e, IEEE 802.11n, and 10Gb Ethernet, because they can be easily encoded using shift-registers, thus requiring less memory and less computational complexity [61, 62]. These features motivate us to take an intensive interest in the construction of large block-length QC-LDPC codes with high girth for future applications in data

storage and communication systems. Note that the term "girth" implies the shortest cycle in a Tanner graph or in the **H** matrix.

In addition, remarkable efforts have been carried out to find various QC-LDPC constructions with explicit algebraic and combinatorial designs. For example, Fan [63] introduced an array code with no 4-cycle length that can be viewed as one of the properties of QC-LDPC codes. Another approaches to design large girth structured QC-LDPC codes based on CPM by deleting certain block-rows and block-columns of the **H** matrix were proposed in [58, 62, 64-66]. Recently, QC-LDPC codes up to the girth 8 were proposed by Sudarsan *et al*. [67], which based on complete protograph. Moreover, Eleftheriou et al. [64] presented a modified array code (MAC) by applying a cyclic shift to a Fan's array code so as to reduce the number of 1's in a lower triangular **H** matrix, and its performance is superior to the Fan's array code. Additionally, Shu Lin *et al.* [10] had significant contribution for algebraic QC-LDPC codes, which have shown good performance with low error-floor and reduced-complexity.

3.7 Algebraic constructions of QC-LDPC Codes based on Circulant matrices

3.7.1 Quasi-Cyclic LDPC Codes

The **H** matrix of a (j,k) QC-LDPC code with column weight j and row weight k, is called regular if the **H** matrix has uniform column weight and row weight [7]. It is based on $L \times L$ CPMs, defined as a mother matrix, $\mathbf{M(H)}$, of size $mL \times nL$, which can be uniquely constructed by shifting the order of an identity matrix, **I**, based on its corresponding CPM, as given by

$$\mathbf{M(H)} = \begin{bmatrix} \mathbf{I}_{a_{11}} & \mathbf{I}_{a_{12}} & \cdots & \mathbf{I}_{a_{1n}} \\ \mathbf{I}_{a_{21}} & \mathbf{I}_{a_{22}} & \cdots & \mathbf{I}_{a_{2n}} \\ \vdots & \vdots & \ddots & \\ \mathbf{I}_{a_{m1}} & \mathbf{I}_{a_{m2}} & \cdots & \mathbf{I}_{a_{mn}} \end{bmatrix}, \tag{3.2}$$

where $a_{ij} \in \{0,1\ldots,L-1,\infty\}$ and $\mathbf{I}_{a_{ij}}$ is defined as the **I** matrix of size $L \times L$ for $1 \le j \le m$ and $1 \le j \le n$, which is obtained by cyclically right shifting the rows of the

I matrix by a_{ij} times. The zero matrix of size $L \times L$ is represented when $a_{ij} = \infty$. The **H** matrix consists of m block-rows indexed from 0 to $m-1$, and n block-columns indexed from 0 to $n-1$. It is noted in ([7], Theorem 2.5) that the girth of an ultra-sparse QC-LDPC code, where $j \geq 3$ cannot be greater than 12.

In addition, the matrix $\mathbf{E}(\mathbf{H})$ is called the exponent or shifting matrix and it can be obtained by replacing each element $\mathbf{I}_{a_{ij}}$ in $\mathbf{M}(\mathbf{H})$ by a_{ij} as follows:

$$\mathbf{E(H)} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}. \tag{3.3}$$

By combining the exponent matrix $\mathbf{E}(\mathbf{H})$ and the CPM $\mathbf{I}_{a_{ij}}$, it will give the **H** matrix. For example, the $\mathbf{M}(\mathbf{H})$ matrix in (3.2) can be constructed using an exponent coupling procedure according to

$$\mathbf{M(H)} = \mathbf{E(H)} \circ \mathbf{I}_{a_{ij}}, \tag{3.4}$$

where $\circ$ is a coupling operator.

A cycle of length $2l$ in the Tanner graph of $\mathbf{E}(\mathbf{H})$ is called a $2l$-block cycle, which can be represented by an exponent chain in the $\mathbf{M}(\mathbf{H})$ matrix according to

$$(\mathbf{I}_{a_{i_1 j_1}} \to \mathbf{I}_{a_{i_1 j_2}} \to \mathbf{I}_{a_{i_2 j_2}}, \cdots, \mathbf{I}_{a_{i_l j_l}} \to \mathbf{I}_{a_{i_l j_1}} \to \mathbf{I}_{a_{i_1 j_1}}) \tag{3.5}$$

or in the $\mathbf{E}(\mathbf{H})$ matrix according to

$$(a_{i_1 j_1} \to a_{i_1 j_2} \to a_{i_2 j_2}, ..., \to a_{i_l j_l} \to a_{i_l j_1} \to a_{i_1 j_1}). \tag{3.6}$$

Due to the presence of short length cycle in the **H** matrix, the performance of LDPC codes will degrade. It is very important to understand the structure of the **H** matrix. The theorem mentioned below was first proposed by Fossorier in [7], which stated that in QC-LDPC codes, the necessary and sufficient condition for the existence of length $2l$-block cycle is given by

$$\sum_{k=1}^{2l} (a_{m_k, n_k} - a_{m_{k+1}, n_k}) \equiv 0 \bmod L, \tag{3.7}$$

where $i_k \neq i_{k+1}$, $j_k \neq j_{k+1}$, and $i_{l+1} = i_l$.

Example 3.1: Let *c* be a length 16 codeword described by **I** size of 4×4 and **M(H)** as follows-

$$\mathbf{M(H)} = \begin{bmatrix} \mathbf{I}_{a_{11}} & \mathbf{I}_{a_{12}} & \mathbf{I}_{a_{13}} & \mathbf{I}_{a_{14}} \\ \mathbf{I}_{a_{21}} & \mathbf{I}_{a_{22}} & \mathbf{I}_{a_{23}} & \mathbf{I}_{a_{24}} \end{bmatrix} \tag{3.7}$$

Let $\mathbf{H} = \begin{bmatrix} \mathbf{I}_0 & \mathbf{I}_0 & \mathbf{I}_0 & \mathbf{I}_0 \\ \mathbf{I}_0 & \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_3 \end{bmatrix}$ which can be represented in binary form as

$$\mathbf{H} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1&0&0&0 & 1&0&0&0 & 1&0&0&0 & 1&0&0&0 \\ 0&1&0&0 & 0&1&0&0 & 0&1&0&0 & 0&1&0&0 \\ 0&0&1&0 & 0&0&1&0 & 0&0&1&0 & 0&0&1&0 \\ 0&0&0&1 & 0&0&0&1 & 0&0&0&1 & 0&0&0&1 \\ \hline 1&0&0&0 & 0&1&0&0 & 0&0&1&0 & 0&0&0&1 \\ 0&1&0&0 & 0&0&1&0 & 0&0&0&1 & 1&0&0&0 \\ 0&0&1&0 & 0&0&0&1 & 1&0&0&0 & 0&1&0&0 \\ 0&0&0&1 & 1&0&0&0 & 0&1&0&0 & 0&0&1&0 \end{array}\right] \tag{3.8}$$

For this code $j=2$, $k=4$ and $L=4$.

3.7.2 QC-LDPC Implementation

Hardware implementation of QC-LDPC codes are mostly done to achieve high throughput required by most applications. Encoding complexity is quadratic with respect to the code length. There are numerous method discussed for reducing the encoding complexity by prior processing to the **H** matrix. The complexity of the encoder also depends on the structure of the QC-LDPC codes. As stated earlier, large codes require more hardware in terms of memory size for processing node requirement. One of the proved techniques for reducing the encoding complexity is by using Chinese Remainder Theorem discussed [11]. On the other hand, the requirement of QC-LDPC decoder depends upon the structure of the code and implementation architecture. Semi-parallel decoder architecture based of structure QC-LDPC codes are often implemented. QC-LDPC codes offer a good trade-off between hardware complexity, cost and the throughput. For a particular application, it is desirable to consider several issues such as type of decoding algorithm, encoding

complexity, and decoding delay, numeric precision on the estimated probabilities, power consumption, and programmability delay.

3.7.3 QC-LDPC applications in modern communication systems and recent advancements

QC-LDPC codes are a class of LDPC codes in which there is cyclic connection between rows or columns of a sub matrix. QC-LDPC codes structure depends on the arrangement of sub matrices and the value's by which they are shifted, random shifting of a sub matrices may result in poor performance of the QC-LDPC codes. Practically, good QC-LDPC codes are well suited for certain practical applications, such as data storage systems, DVB-T2/S2, IEEE 802.16e, IEEE 802.11n, etc., because they can easily be encoded using shift-registers, thus requiring less memory and less computational complexity [8].

The most appealing LDPC codes are QC-LDPC for practical systems as the structure of quasi-cyclic matrix allows for linear time encoding using only shift registers and also rendering efficient routing for decoding implementation [62, 66, 67]. Furthermore, it enables the storage of the coding matrix with requirements of few memory units. Active research on QC-LDPC codes has been carried out on the efficient **H** matrix construction with large girth, meaning the length of the shortest cycle in tanner graph representation [55].

3.8 Summary of various existing QC-LDPC encoding schemes

Table 3-3: Various existing encoding schemes

Encoding scheme Classification category	Name of Encoding scheme
Category -1 Approximate Lower Triangulation Schemes	• Richardson Encoding Scheme • Adaptive Message Length Encoding Scheme • Arbitrary Bit-Generation and Correction

	• Encoding with a systematic approximate lower triangular form • Encoding for GLDPC codes • Two stage encoding with Triangular Factorization
Category -2 Families of Algebraic Construction of QC-LDPC codes	• Algebraic Construction of QC-LDPC codes: Bresnan Code • Algebraic Construction of QC-LDPC codes by Dispersion • Algebraic Construction of QC-LDPC codes: Rakibul Code
Category -3 Other Existing Encoding Schemes	• Encoding of QCLDPC Codes Related to Cyclic MDS Codes • Efficient Encoding of IEEE 802.11n LDPC Codes • Encoding of Array LDPC Codes by FAN • Magic square method to construct QC-LDPC codes

Here we summarize the work from different research group around the globe. In the Table 3-3 only the work in recent research has been mentioned and indeed each group has many contributions in the domain of encoding QC-LDPC codes.

Chapter 4: Proposed construction method of QC-LDPC Code

As we discussed in the previous chapter, some structured LDPC codes have limitations in their interconnection patterns, girth and implementation complexity. In this category of codes they vary in their performances and pattern of connections for variable nodes and check nodes. However the connection pattern between rows or columns in these codes which resulted from chosen sub-matrices of searched codes may not be same for all rows or columns. The more the size of sub-matrices, the more will be its storage and handling cost and hence it increases the complexity of the decoder.

QC-LDPC codes are known to be efficiently encoded with shift registers [34, 68] and their decoder design which require simple address generation algorithm and less memory requirement. In QC-LDPC codes, rows and column connections are constructed by shifting identity sub-matrices. Therefore, if we know the location of one row or a column in the decoder, one can find out the location of the remaining rows and columns in the same sub-matrix. It will provide a simple and easy construction of **H** matrix Furthermore, QC-LDPC codes can perform close to the channel limit as shown in [69].

QC-LDPC codes have been constructed by using several methods. The structural properties of the codes majorly depend on the shift values of sub-matrices and their arrangements. If we use random shift values of identity sub-matrices then it will result in reduced girth and so poor performance. Various construction methods as we discussed in chapter three have constraints on no-four cycle. Code constructions using these methods have at least girth of six and wide range of code rate and lengths. All construction methods have certain limitations so developed codes are restricted in all or one of the properties such as code rate, code length and girth. In addition, a recursive approach develop as in [70, 71] would be applied for wide range of girths, code rates and lengths. Using this method, firstly a base QC-LDPC code is constructed from one of already developed methods such as geometric or algebraic

construction techniques. Further, base matrix expanded using replacement of '1' entries with randomly shifted sub-matrix of size $p \times p$ and '0' entries by zero sub-matrix of size $p \times p$ with the constraints of at least same girth and minimum hamming distance as of base matrix.

In this chapter, we develop a new search algorithm which is useful for construction of large block-length LDPC codes. We search for exponent matrices with high girth, keeping in mind to reduce its circular permutation matrix (CPM) size. It will lead to less memory requirement and less implementation cost. For reducing the encoding implementation complexity, we use Chinese Remainder Theorem (CRT) for combining proposed component matrices. Another major advantage of the proposed algorithm is that it can be applied to higher column weight and different sub-matrix arrangements. Our method having reduced encoder complexity and its hardware cost. This algorithm could be used to construct codes optimized for better decoding performance as well. It also offers the flexibility as compared to the random codes. The Technical details associated with this is the development of the new codes with following criteria's and their technical details are given are as follows.

1. Component **H** matrix generation.
2. Finding Lower bound on CPM size for optimized higher girth.
3. Combining component matrices to obtain desired medium to large size block-length **H** matrix using Chinese Remainder Theorem (CRT).

4.1 Chinese Remainder Theorem

Let I be a positive integer, $\{L_1, L_2, \cdots, L_s\}$ be s moduli, and $\{r_1, r_2, \cdots, r_s\}$ be s remainders of I, i.e.,

$$r_b \equiv I \left| L_b \right|, \tag{4.1}$$

where $0 \le r_b \le L_b$ for $1 \le b \le s$. If all the moduli L_b's are co-prime to one another and $0 \le I < \prod_{b=1}^{s} L_b$, then I can be uniquely reconstructed from its s remainders via a simple CRT theorem [72] according to:

$$I = \sum_{b=1}^{s} r_b a_b \overline{L_b} \left| L \right|, \tag{4.2}$$

where $L = \prod_{b=1}^{s} L_b$, $\overline{L_b} = L / L_b$, and $A_b \overline{L_b} \equiv 1 \left| L_b \right|$.

4.2 Generalized Combination of QC-LDPC codes via CRT

Let C_b be a QC-LDPC codeword, where $b = 1, 2, .., s$, whose $\mathbf{H}_b$ is an $m \times n$ array of $L_b \times L_b$ CPMs and/or zero matrices. Let $\mathbf{E(H}_b\mathbf{)} = (a_{ij}^{(b)})$ be the exponent matrix and $L = \prod_{b=1}^{s} L_b$. A QC-LDPC code C with the $mL \times nL$ parity-check matrix, **H,** can be constructed by using the generalized combining method, which obtains the exponent matrix $\mathbf{E(H)} = (a_{ij})$ according to (4.2). In the case, where $a_{ij}^{(b)} \neq \infty$ in **E(H)** for all $b = 1, 2, .., s$, we find a_{ij} according to

$$a_{ij} = \sum_{b=1}^{s} a_{ij}^{b} A_b \overline{L_b} \left| L \right|, \tag{4.3}$$

Proposition 4.1 [72], For $b = 1, 2, .., s$, let g_b denote the girth of C_b and g denote the girth of C, then

$$g \geq \max\{g_1, g_2, ..., g_s\}. \tag{4.4}$$

In the next section, we propose a novel method to construct a large block-length **H** matrix having a high girth with less complex encoding by combining the component QC-LDPC codes and the CRT algorithm.

4.3 Proposed search algorithm for QC-LDPC codes using CRT

In this section, we propose construction methods using which a wide range of girths, code rates and code lengths could be produced with less CPM size. Furthermore this works leads to construction more general form of **H** matrix, which consists of medium to large block-length size. The proposed QC-LDPC codes based structure of **H** matrix can be useful in real application in upcoming and modified version of various standards, using LPDC codes as a FEC channel code.

4.3.1 Flowchart of designing H matrix.

Designing steps involve in construction of desired **H** matrix, can be summarized in a form of flow chart as shown in Figure 4-1:

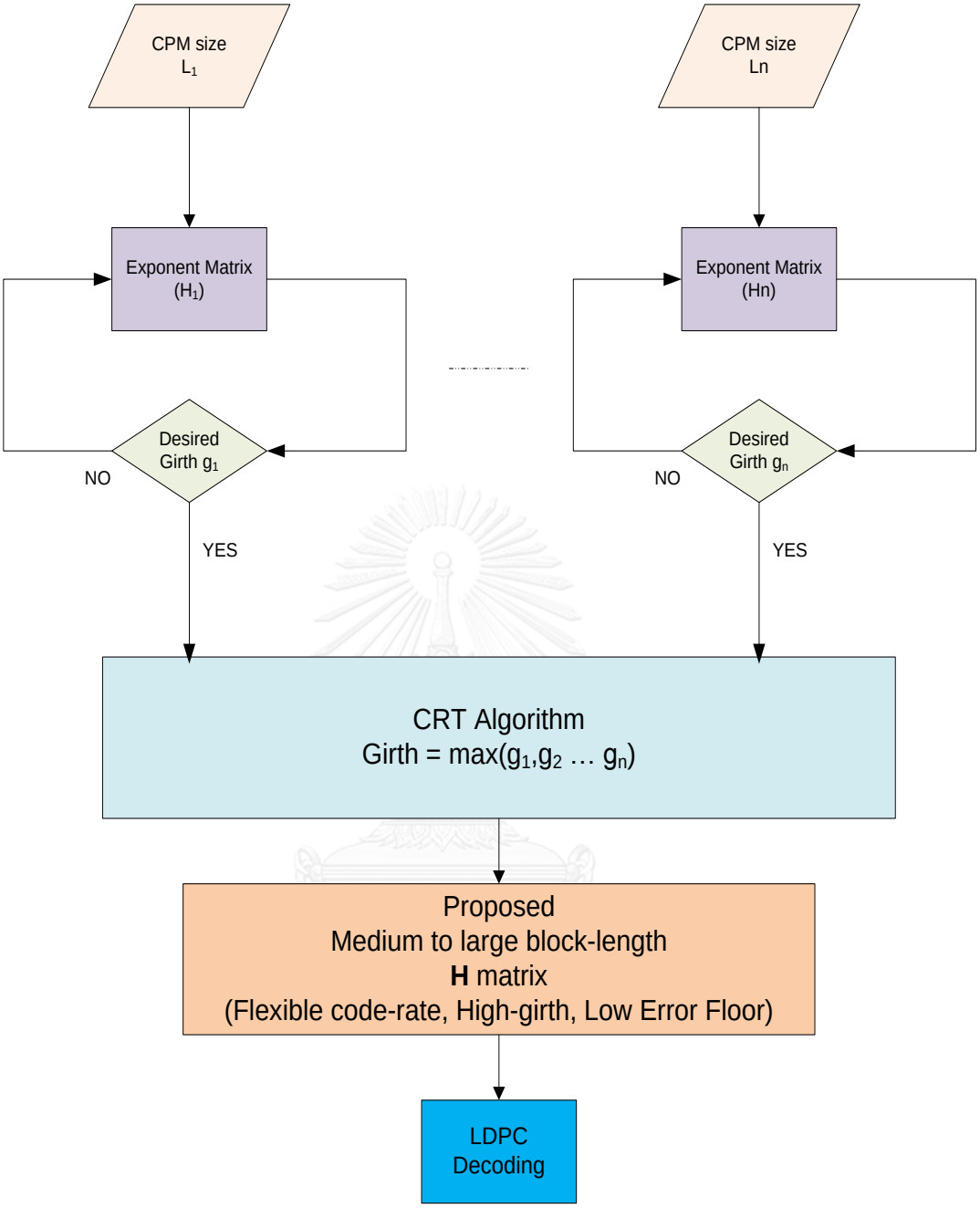


Figure 4-1: Flow Chart for proposed algorithm

4.3.2 Proposed Method

This section introduces a novel method for constructing the **H** matrix for medium to large block-length, which has high girth and less complex in terms of computation.

Assume that L_1 and L_2 are the prime numbers and they indicate the CPM size of two component matrices $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$, which have the girth g_1 and g_2, respectively.

The procedure explained here is for constructing the **H** matrix of size $jL \times kL$ such that $L = L_1 \times L_2$ by using CRT as in (4.2) without losing its local girth. Later in this work, it can be extended to combine $\overline{\mathbf{H_1}}, \overline{\mathbf{H_2}}, ..., \overline{\mathbf{H_s}}$ component matrices having the CPM size of $L_1, L_2, \cdots, L_s$ to obtain the **H** matrix such that $L = L_1 L_2, \cdots, L_s$. Below are the steps of the proposed method.

Step 1 To construct a component $\bar{\mathbf{H}}_1(j,k)$ matrix, where *j* and *k* are the number of block-rows and block-columns, respectively. The starting condition for the $\bar{\mathbf{H}}_1(j,k)$ matrix is given in Table 4-1, where the indexed number 0 is an $L_1 \times L_1$ identity matrix **I**, and 1 represents one right shifted order of **I**, and so on. The indexed number *Z* is an intended shifted order of $L_1 \times L_1$ CPM that we need to find out. It should be noted that the size of $\bar{\mathbf{H}}_1$ matrix is $jL_1 \times kL_1$.

Step 2 For each column-block (starting from the leftmost column to the right), replace each *Z* from the 1st to the *j*th row using a number between 0 to $L_1 - 1$. To do so, we find all possible data patterns of each column-block, where the maximum number of data patterns denoted as P_{fc} is given by

$$P_{fc} = \binom{p}{1}^{j-1}, \tag{4.5}$$

where *p* is the size of the chosen CPM's. For example, if $L_1 = 3$, there will be $P_{fc} = \binom{3}{1}^{3-1} = 3^2 = 9$ different data patterns available for the 1st column. Then, we replace all remaining block-rows with index value *Z* in Table 4-1. To calculate *Z*, we replace each data pattern in the 1st column and compute the local girth assuming that the remaining *Z*'s in the other columns by the $L_1 \times L_1$ zero matrix. Note that if we cannot find the local girth (i.e., no cycle), we will assume that the girth is infinite.

Table 4-1: A proposed generalized component matrix

Block-column index

Block-row index

$\bar{\mathbf{H}}_1$	1	2	…	$k-1$	k
1	0	1	…	$k-2$	$k-1$
2	Z	Z	…	Z	Z
⋮	⋮	⋮	⋱	⋮	⋮
$j-1$	Z	Z	Z	Z	Z
j	Z	Z	Z	Z	Z

Step 3 The data pattern that yields the largest local girth with minimum value indexing in data patterns will be selected for the 1st column. Then, we proceed the same procedure as explained in Step 2 in a column by column manner until all block-columns are filled with the chosen number of data patterns. Table 4-2 shows an example of the component matrix $\bar{\mathbf{H}}_1$ after obtaining all Z's for $L_1 = 29$ and $g_1 = 8$. This process ensures the minimum size of CPM's in QC-LDPC codes, which will be useful for constructing a good **H** matrix with high girth value and low memory requirement for hardware implementation. Table 4-3 illustrates the minimum lower bound of the CPM size for various block lengths of the component matrix based on extensive simulation search for regular $(3,k)$ LDPC codes. The obtained CPM size will be the optimized lower bound for constructing the QC-LDPC parity-check matrix with high girth and variable code rates.

Table 4-2: A designed $\bar{\mathbf{H}}_1$ index matrix

Block-column index

lock-row index

$\bar{\mathbf{H}}_1$	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6
2	0	3	8	0	0	10	24
3	0	0	13	1	8	0	15

Table 4-3: Estimation of minimum CPM size *L* with corresponding girth

k (Block-length)	$g=6$	$g=8$	$g=10$	$g=12$
5	7	17	83	223
7	7	29	239	709
9	11	47	499	1399
11	11	61	743	3271

Step 4 The other component matrix $\bar{\mathbf{H}}_2$ can be obtained by choosing a suitable size of a prime number L_2 based on Table 4-4, such that it maintains the optimum lower bound for the desired girth g_2. For example, we choose a lower bound of the CPM size from Table 4-3, i.e., $L_2 \geq 7$ for $g=6$. The construction procedure for $\bar{\mathbf{H}}_2$ is similar to that for $\bar{\mathbf{H}}_1$. Table 4-4 shows an example of the component matrix $\bar{\mathbf{H}}_2$ after obtaining all *Z*'s for $L_2 \geq 7$ and $g=6$.

Step 5 Finally, we construct the exponent matrix **E(H)** by combining all the component matrices via CRT and replacing each entry a_{ij} of **E(H)** with $\mathbf{I}_{a_{ij}}$ so as to obtain the **H** matrix of size $mL \times nL$ with girth *g*, which still satisfies the condition in (4.4), i.e., $g \geq \max\{g_1, g_2\}$ as shown in Table 4-5.

It should be pointed out that with carefully selecting the CPM size and block length, we can construct any large block-length **H** matrix up to the girth of 12 for QC-LDPC codes.

Table 4-4: A designed $\bar{\mathbf{H}}_2$ index matrix

Block-column index

Block-row index	$\bar{\mathbf{H}}_2$	1	2	3	4	5	6	7
	1	0	1	2	3	4	5	6
	2	0	4	1	1	5	2	1
	3	0	2	4	2	2	1	3

Table 4-5: A combined exponent matrix, **E(H)** via CRT

Block-column index

Block-row index	E(H)	1	2	3	4	5	6	7
	1	0	1	2	3	4	5	6
	2	0	32	8	29	145	184	169
	3	0	58	158	30	37	29	73

4.4 Simulation and Results

The bit-error rate (BER) performance of the proposed method and some well known existing methods compared by considering an $M \times N$ size **H** matrix, where *M* is the number of parity bits, *N* is the length of the codeword with code rate *R* and *R* is equal to $1-M/N$. To evaluate the performance, we simulate the system based on an additive white Gaussian noise (AWGN) channel, where a binary input sequence $a_k \in \{\pm 1\}$ of length $N-M$ bits is encoded by an LDPC encoder and is mapped to an *N*-bit coded sequence $b_k \in \{\pm 1\}$. Therefore, the received sequence is given by $y_k = b_k + n_k$, where n_k is AWGN with zero mean and variance of σ^2. At the receiver end, the received sequence y_k is decoded by LDPC decoder based on a message passing algorithm in log domain with 10 numbers of iterations. The signal-to-noise ratio (SNR) is defined as $\text{SNR} = 10\log_{10}\left(1/2R\sigma^2\right)$ in decibel (dB). Each BER point is computed based on a minimum number of 10000 data packets.

4.4.1 Girth 8 codes

Example-4.1: In this example, we study the proposed method based on the component QC-LDPC codes combined with CRT to construct a large block-length **H** matrix. The attained matrix has a constant degree of 3 for each symbol node. By using our proposed algorithm, we construct a code C_1 for girth $g_1 = 8$ whose exponent matrix $\bar{\mathbf{H}}_1$ is of size 3×7 as shown in Table 4-2. To expand $\bar{\mathbf{H}}_1$, we first select $s = 2$. Then, we carefully select choose $L_1 = 29$ and $L_2 = 7$ in such a way to maintain lower bound on CPM, for $g_1 = 8$, $g_2 = 6$. We found $L_1 = 29$ and $L_2 = 7$ respectively

according to Table 4-3. Hence after combining, CPM size of the **E(H)** matrix will be of size $L = L_1 \times L_2 = 203$. Similarly, we construct the 3×7 exponent matrix $\bar{\mathbf{H}}_2$ using our proposed algorithm for $g_1 = 6$ as shown in Table 4-4. Then, we obtain **E(H)** by combining $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ via CRT as shown in Table 4-5. Finally, we replace each indices a_{ij} of **E(H)** with $\mathbf{I}_{a_{ij}}$. The obtained **H** matrix gives a QC-LDPC code with girth $g = 8$.

Figure 4-2 illustrates the BER simulated performance of the proposed QC-LDPC-CRT (1421,609) code, which is compared with some well-known existing LDPC codes, where FAN array-CRT is the code from shortened array codes based on CRT as in [73], and QC-LDPC-PEG are PEG based QC-LDPC codes as described in [8]. Noticeably, the proposed algorithm performs improved than the other algorithms when the E_b / N_0 ratio is high.

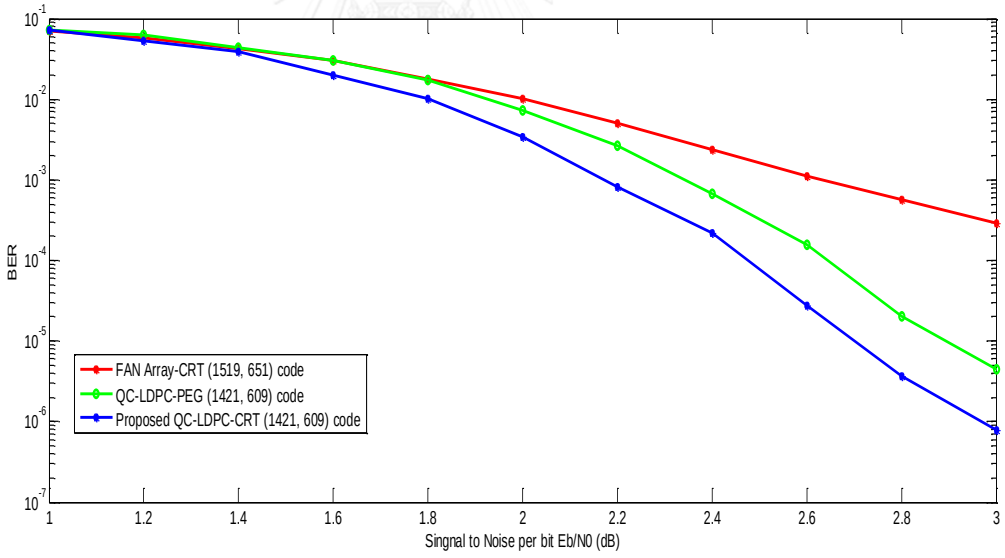


Figure 4-2: BER performance comparison

We also compare the BER performance of different methods as a function of the number of iterations at $E_b / N_0 = 3$ dB as in Figure 4-2. It is apparent that the proposed algorithm converges faster than other algorithms. Furthermore, we also examine the local girth of each algorithm as shown in Figure 4-3. Clearly, the proposed algorithm offers the girth in the range of 8, if compared to other algorithms.

However, the girth is not better than QC-LDPC-PEG codes, as QC-LDPC PEG based codes are most greedy and can achieve girth up to 12. Our proposed CRT based QC-LDPC code can have higher girth by carefully selections of CPMs and block-length size as in Table 4-3.

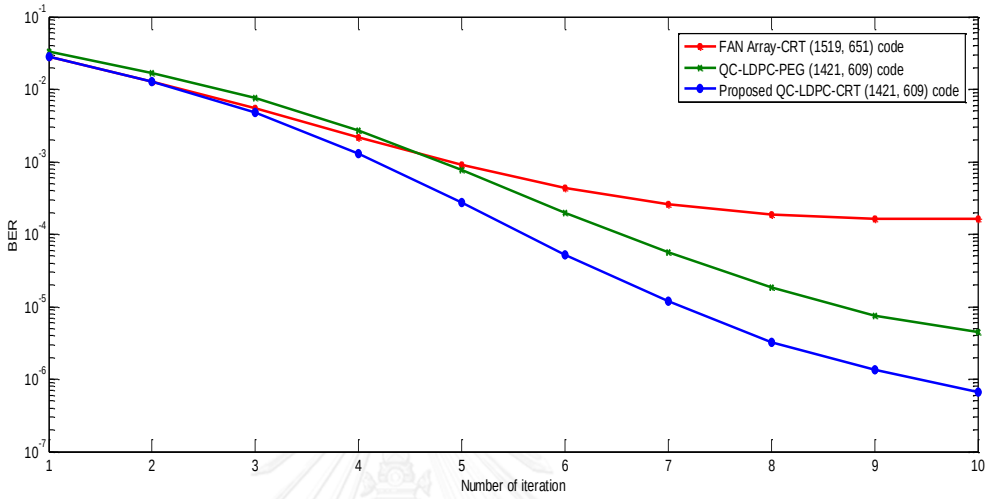


Figure 4-3: BER performance as a function of the number of iterations for different **H** matrices at SNR = 3 dB

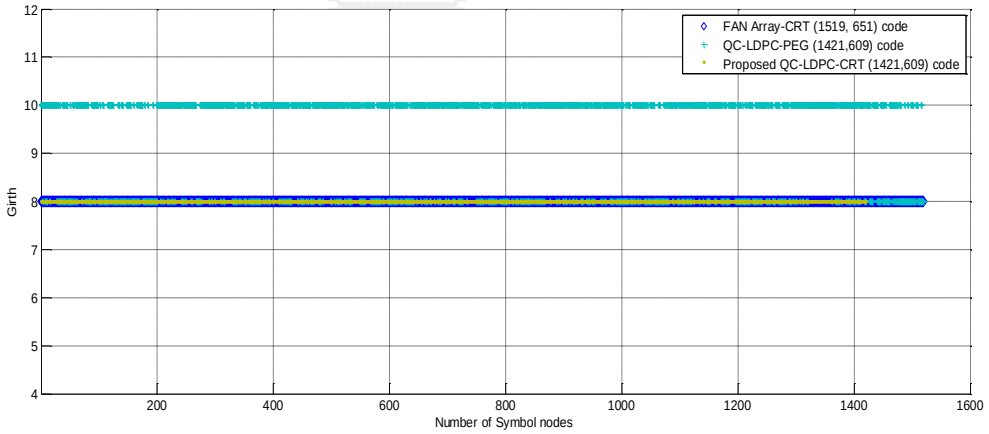


Figure 4-4: Girth comparison of proposed-CRT code

Example-4.2: In this example, we study the proposed method based on the component matrices combined with CRT to construct a large block-length **H** matrix for different code rate. The obtained matrix has a uniform degree of 3 for each symbol node. By using our proposed algorithm, we first construct a code C_1 for girth $g_1 = 8$ whose exponent matrix $\overline{\mathbf{H}_1}$ is of size 3×5 as shown in Table 4-6. To expand $\overline{\mathbf{H}}_1$, we

first select $s = 2$. Then, we carefully choose $L_1 = 17$ and $L_2 = 7$ in such a way to maintain lower bound on CPM, for $g_1 = 8$, $g_1 = 6$ we found $L_1 = 17$ and $L_2 = 7$ respectively according to Table 4-3. Hence after combining, CPM size of **E(H)** matrix will be $L = L_1 \times L_2 = 119$. Similarly, we construct the 3×5 exponent matrix $\bar{\mathbf{H}}_2$ using our proposed algorithm for $g_1 = 6$ as shown in Table 4-7. Then, we obtain **E(H)** by combining $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ via CRT as shown in Table 4-8. Finally, we replace each entities a_{ij} of **E(H)** with $\mathbf{I}_{a_{ij}}$. The obtained **H** matrix gives a QC-LDPC code with girth $g = 8$.

Table 4-6: A designed $\bar{\mathbf{H}}_1$ index matrix

Block-column index

Block-row index	$\bar{\mathbf{H}}_1$	1	2	3	4	5
	1	0	1	2	3	4
	2	4	6	10	10	0
	3	0	3	0	9	0

Table 4-7: A designed $\bar{\mathbf{H}}_2$ index matrix

Block-column index

Block-row index	$\bar{\mathbf{H}}_2$	1	2	3	4	5
	1	0	1	2	3	4
	2	1	0	0	3	0
	3	0	2	1	0	0

Table 4-8: A combined exponent matrix **E(H)** via CRT

Block-row index \ Block-column index	1	2	3	4	5
E(H)					
1	0	1	2	3	4
2	106	91	112	10	0
3	0	37	85	77	0

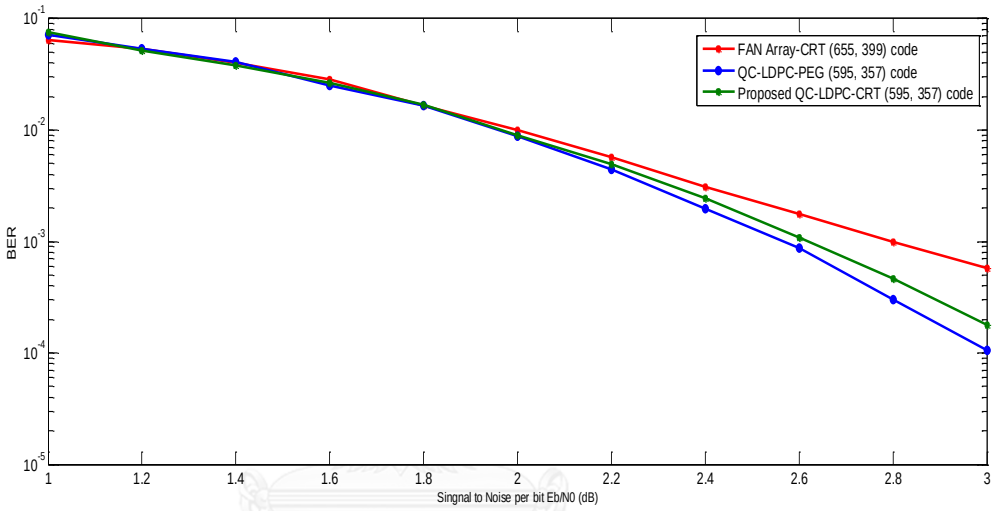


Figure 4-5: BER performance comparison

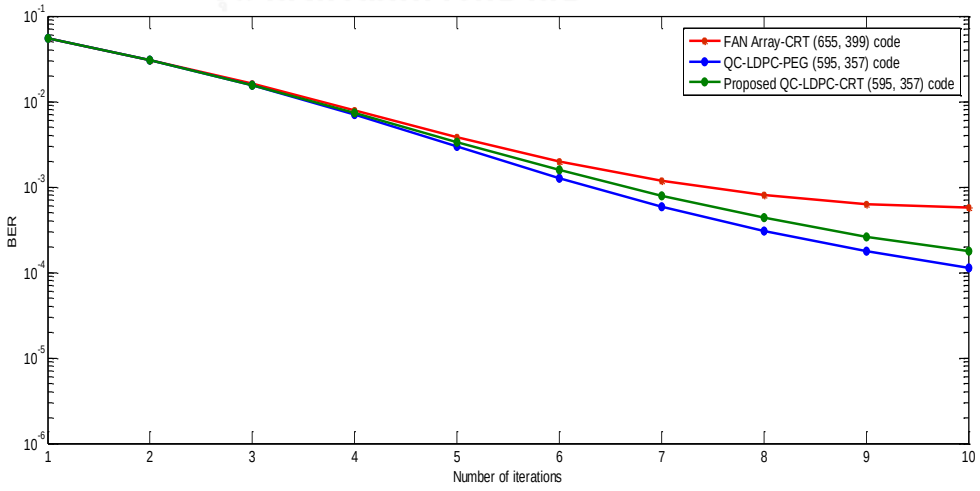


Figure 4-6: BER performance as a function of the number of iterations for different **H** matrices at SNR = 3 dB.

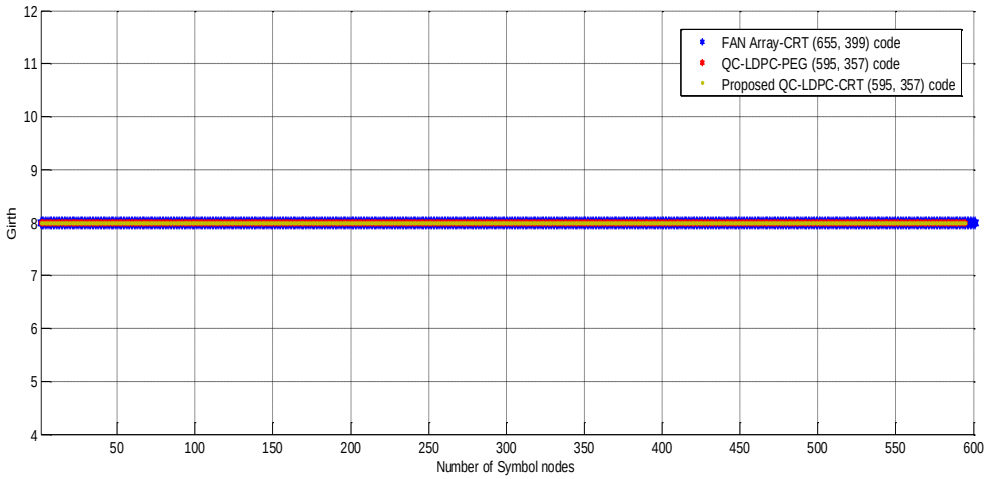


Figure 4-7: Girth comparison of proposed-CRT code

Example-4.3: In this example, we study the proposed method based on the component matrices combined with CRT to construct a large block-length **H** matrix. The obtained matrix has a uniform degree of 3 for each symbol node. By using our proposed algorithm, we construct a code C_1 for girth $g_1 = 8$ whose exponent matrix $\overline{\mathbf{H}_1}$ is of size 3×9 as shown in Table 4-9. To expand $\bar{\mathbf{H}}_1$, first select $s = 2$. Then, we carefully choose $L_1 = 47$ and $L_2 = 7$ in such a way to maintain lower bound on CPM, for $g_1 = 8$, $g_1 = 6$ we found $L_1 = 47$ and $L_2 = 7$ respectively according to Table 4-3. Hence after combining, CPM size of **E(H)** matrix will be $L = L_1 \times L_2 = 517$. Similarly, we construct the 3×9 exponent matrix $\bar{\mathbf{H}}_2$ using our proposed algorithm for $g_1 = 6$ as shown in Table 4-10. Then, we obtain **E(H)** by combining $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ via CRT as shown in Table 4-11. Finally, we replace each entities a_{ij} of **E(H)** with $\mathbf{I}_{a_{ij}}$. The obtained **H** matrix gives a QC-LDPC code with girth $g = 8$.

Table 4-9: A designed $\bar{\mathbf{H}}_1$ index matrix

Block-column index

Block-row index	$\bar{\mathbf{H}}_1$	1	2	3	4	5	6	7	8	9
	1	0	1	2	3	4	5	6	7	8
	2	8	10	0	16	1	19	0	23	0
	3	0	3	12	6	15	1	21	0	0

Table 4-10: A designed $\bar{\mathbf{H}}_2$ index matrix

Block-column index

Block-row index	$\bar{\mathbf{H}}_2$	1	2	3	4	5	6	7	8	9
	1	0	1	2	3	4	5	6	7	8
	2	1	0	0	3	6	2	0	2	0
	3	0	2	1	0	0	0	4	9	0

Table 4-11: A combined exponent matrix, **E(H)** via CRT

Block-column index

Block-row index	**E(H)**	1	2	3	4	5	6	7	8	9
	1	0	1	2	3	4	5	6	7	8
	2	243	198	0	157	424	442	0	211	0
	3	0	332	12	429	297	330	440	141	0

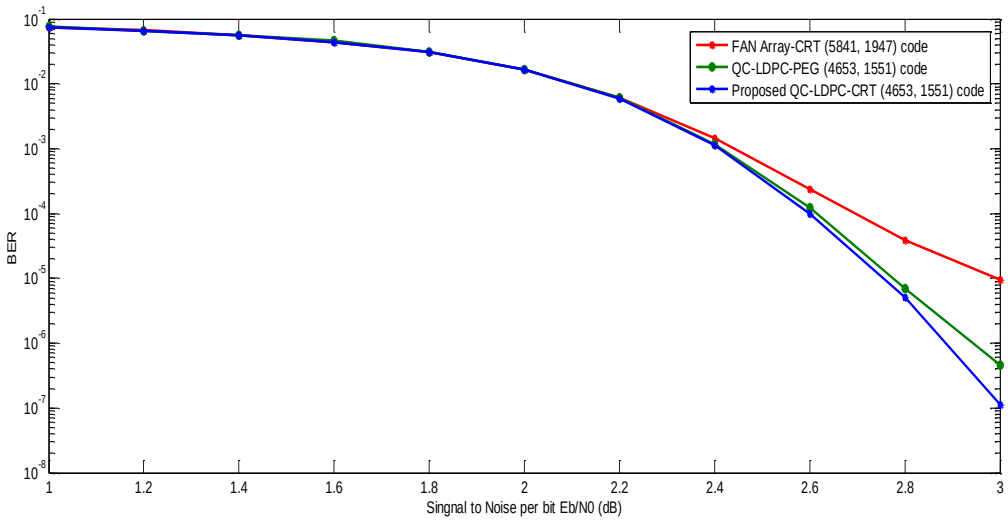


Figure 4-8: BER performance comparison

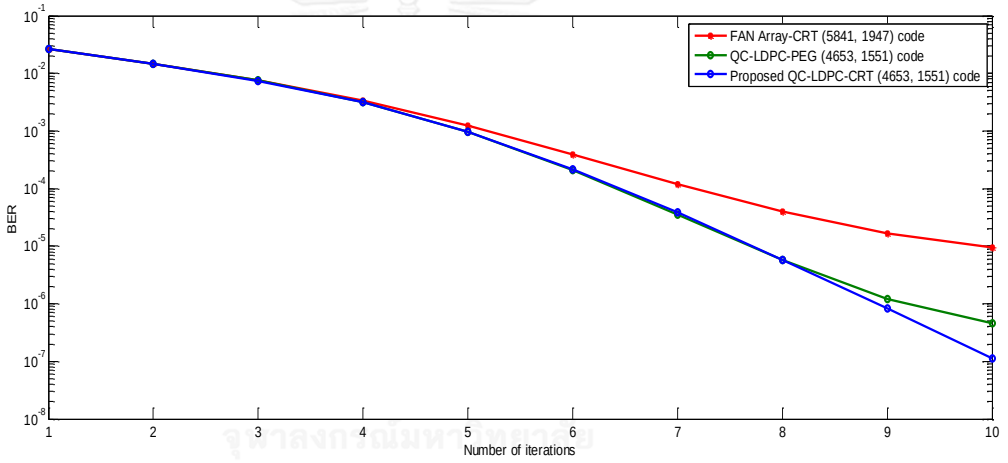


Figure 4-9: BER performance as a function of the number of iterations for different **H** matrices at SNR = 3 dB

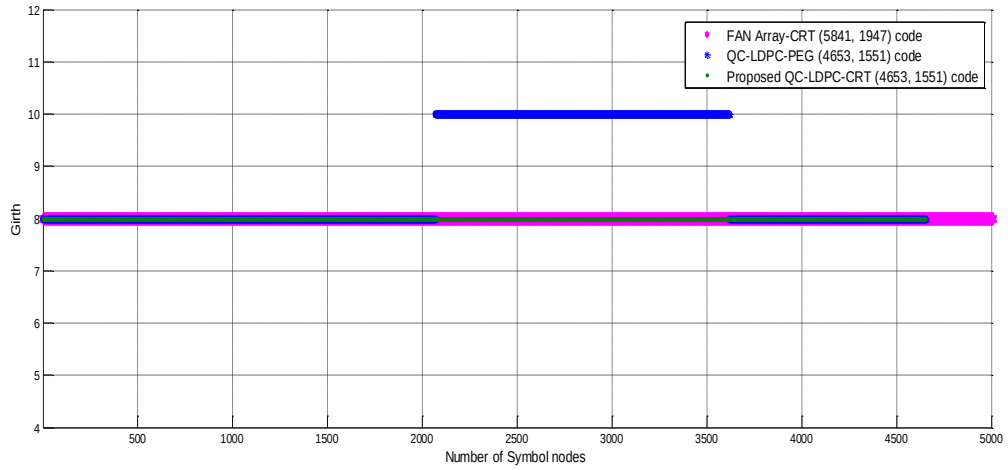


Figure 4-10: Girth comparison of proposed-CRT code

4.4.2 Girth 10 codes

Example-4.4: In this example, we study the proposed method based on the component matrices combined with CRT to construct a large block-length **H** matrix. The obtained matrix has a uniform degree of 3 for each symbol node. By using our proposed algorithm, we construct a code C_1 for girth $g_1 = 10$ whose exponent matrix $\bar{\mathbf{H}}_1$ is of size 3×5 as shown in Table 4-12. To expand $\bar{\mathbf{H}}_1$, we first select $s = 2$. Then, we carefully choose $L_1 = 83$ and $L_2 = 7$ in such a way to maintain lower bound on CPM, for $g_1 = 10$, $g_1 = 6$ we found $L_1 = 83$ and $L_2 = 7$ respectively according to Table 4-3. Hence after combining, CPM size of **E(H)** matrix will be $L = L_1 \times L_2 = 581$. Similarly, we construct the 3×5 exponent matrix $\bar{\mathbf{H}}_2$ using our proposed algorithm for $g_1 = 6$ as shown in Table 4-13. Then, we obtain **E(H)** by combining $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ via CRT as shown in Table 4-14. Finally, we replace each entities a_{ij} of **E(H)** with $\mathbf{I}_{a_{ij}}$. The obtained **H** matrix gives a QC-LDPC code with girth $g = 10$.

Table 4-12: A designed $\bar{\mathbf{H}}_1$ index matrix

	Block-column index				
Block-row index	$\bar{\mathbf{H}}_1$	1	2	3	4	5
	1	0	1	2	3	4
	2	14	11	39	66	0
	3	3	19	8	34	0

Table 4-13: A designed $\bar{\mathbf{H}}_2$ index matrix

Block-row index	Block-column index					
$\bar{\mathbf{H}}_2$	1	2	3	4	5
1	0	1	2	3	4
2	1	0	0	3	0
3	0	2	1	0	0

Table 4-14: A combined exponent matrix, **E(H)** via CRT

Block-row index	Block-column index				
E(H)	1	2	3	4	5
1	0	1	2	3	4
2	512	343	371	66	0
3	252	268	8	532	0

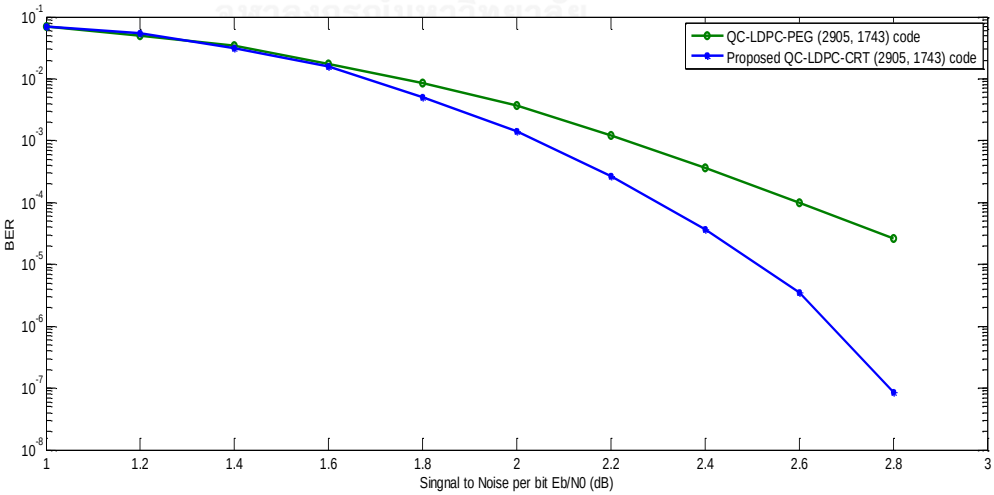


Figure 4-11: BER performance comparison

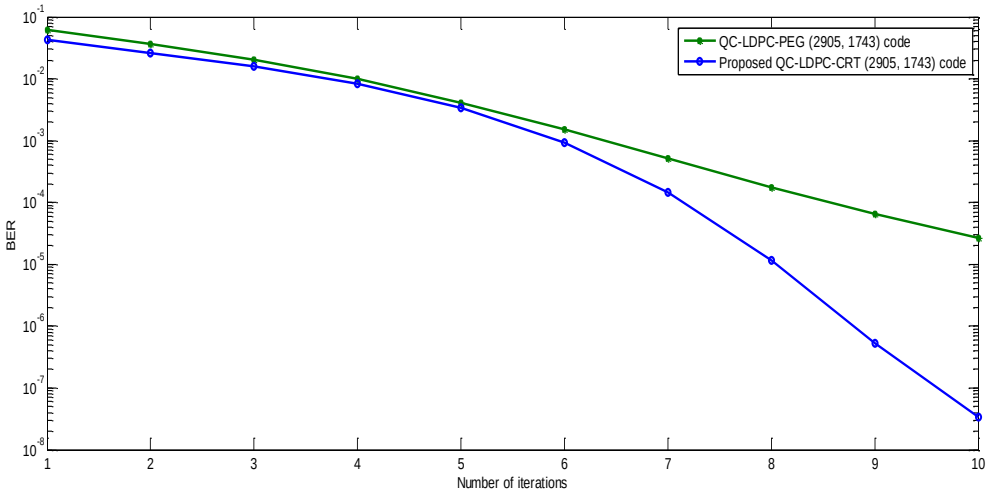


Figure 4-12: BER performance as a function of the number of iterations for different **H** matrices at SNR = 2.8 dB

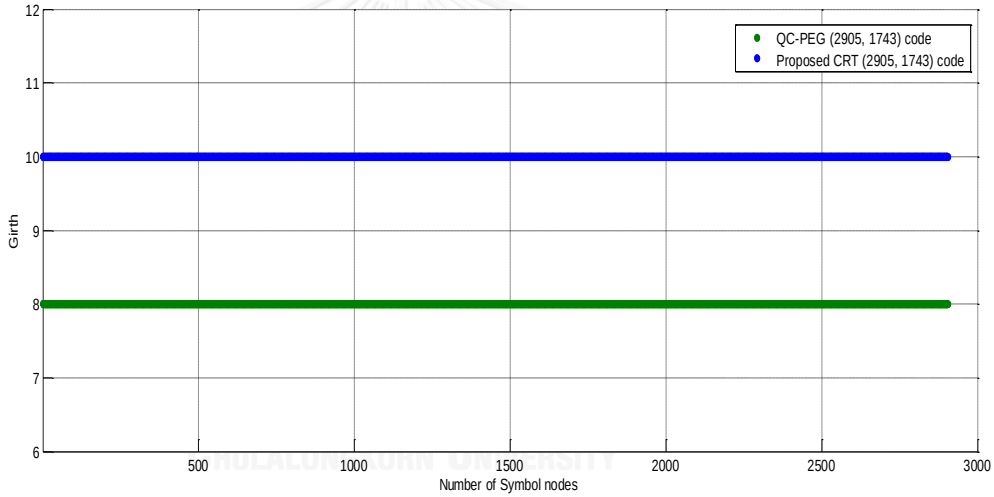


Figure 4-13: Girth comparison of proposed-CRT codes

Example-4.5: In this example, we study the proposed method based on the component matrices combined with CRT to construct a large block-length **H** matrix for girth 10. The obtained matrix has a uniform degree of 3 for each symbol node. By using our proposed algorithm, we construct a code C_1 for girth $g_1 = 10$ whose exponent matrix $\bar{\mathbf{H}}_1$ is of size 3×7 as shown in Table 4-15. To expand $\bar{\mathbf{H}}_1$, we first select $s = 2$. Then, we carefully choose $L_1 = 239$ and $L_2 = 7$ in such a way to maintain lower bound on CPM, for $g_1 = 10$, $g_1 = 6$ we found $L_1 = 239$ and $L_2 = 7$ respectively according to Table 4-3. Hence after combining, CPM size of **E(H)**

matrix will be $L = L_1 \times L_2 = 1673$. Similarly, we construct the 3×7 exponent matrix $\bar{\mathbf{H}}_2$ using our proposed algorithm for $g_1 = 6$ as shown in Table 4-16. Then, we obtain **E(H)** by combining $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ via CRT as shown in Table 4-17. Finally, we replace each entities a_{ij} of **E(H)** with $\mathbf{I}_{a_{ij}}$. The obtained **H** matrix gives a QC-LDPC code with girth $g = 10$.

Table 4-15: A designed $\bar{\mathbf{H}}_1$ index matrix

	Block-column index						
Block-row index	$\bar{\mathbf{H}}_1$	1	2	3	4	5	6	7
	1	0	1	2	3	4	5	6
	2	0	3	7	55	78	74	181
	3	0	0	10	34	133	47	97

Table 4-16: A designed $\bar{\mathbf{H}}_2$ index matrix

	Block-column index						
Block-row index	$\bar{\mathbf{H}}_2$	1	2	3	4	5	6	7
	1	0	1	2	3	4	5	6
	2	0	4	1	1	5	2	1
	3	0	2	4	2	2	1	3

Table 4-17: A combined exponent matrix, **E(H)** via CRT

	Block-column index						
Block-row index	**E(H)**	1	2	3	4	5	6	7
	1	0	1	2	3	4	5	6
	2	0	242	1212	533	1034	1269	659
	3	0	478	249	751	611	764	1053

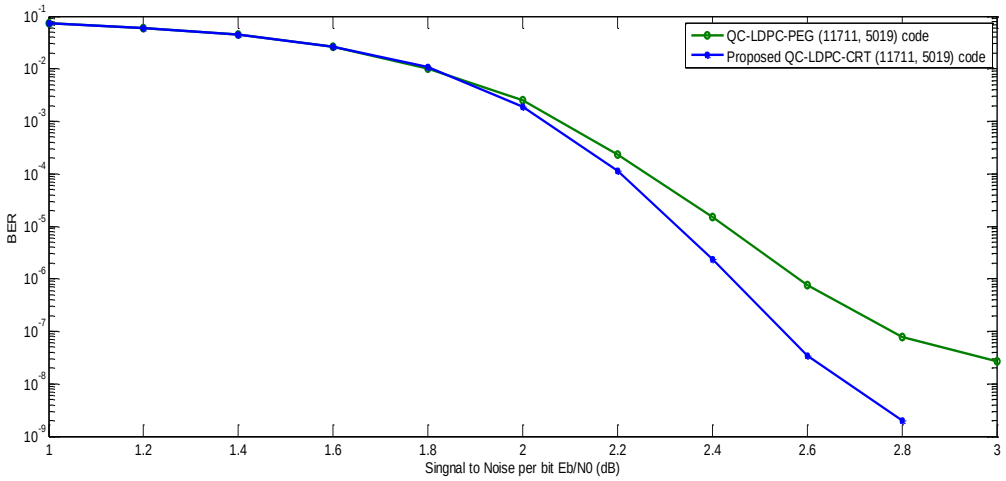


Figure 4-14: BER performance comparison

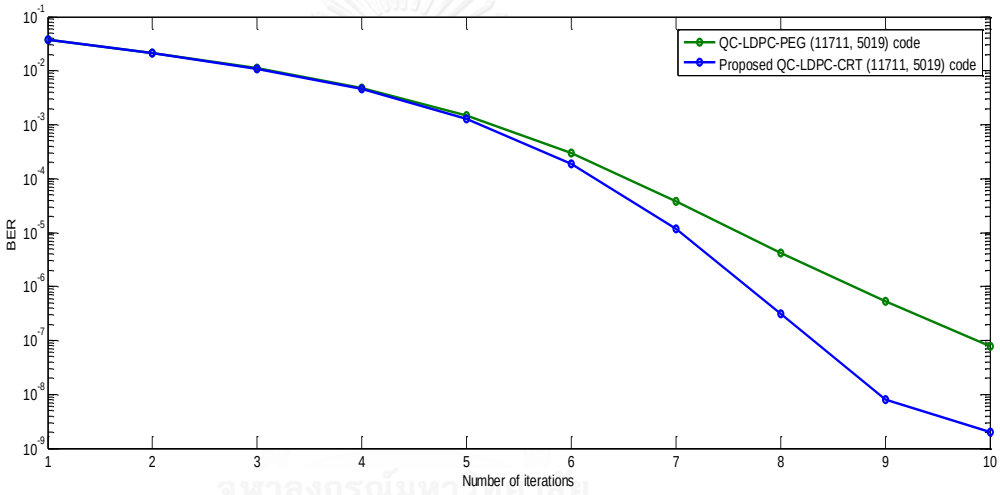


Figure 4-15: BER performance as a function of the number of iterations for different **H** matrices at SNR=2.8 dB

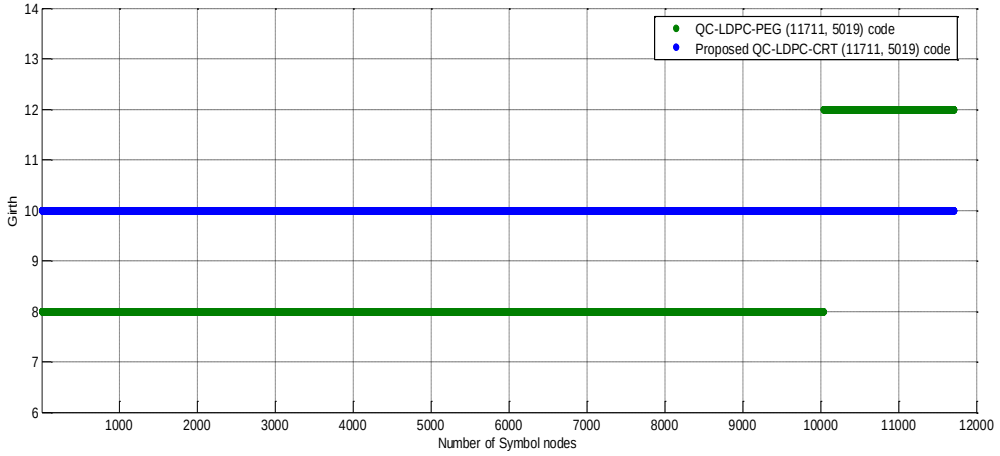


Figure 4-16: Girth comparison of proposed-CRT code

4.4.3 Girth 12 codes

Example-4.6: In this example, we study the proposed method based on the component matrices combined with CRT to construct a large block-length **H** matrix. The obtained matrix has a uniform degree of 3 for each symbol node. By using our proposed algorithm, we construct a code C_1 for girth $g_1 = 12$ whose exponent matrix $\bar{\mathbf{H}}_1$ is of size 3×5 as shown in Table 4-18. To expand $\bar{\mathbf{H}}_1$, we first select $s = 2$. Then, we carefully choose $L_1 = 223$ and $L_2 = 7$ in such a way to maintain lower bound on CPM, for $g_1 = 12$, $g_1 = 6$ we found $L_1 = 223$ and $L_2 = 7$ respectively according to Table 4-3. Hence after combining, CPM size of **E(H)** matrix will be $L = L_1 \times L_2 = 1561$. Similarly, we construct the 3×5 exponent matrix $\bar{\mathbf{H}}_2$ using our proposed algorithm for $g_1 = 6$ as shown in Table 4-19. Then, we obtain **E(H)** by combining $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ via CRT as shown in Table 4-20. Finally, we replace each entities a_{ij} of **E(H)** with $\mathbf{I}_{a_{ij}}$. The obtained **H** matrix gives a QC-LDPC code with girth $g = 12$.

Table 4-18 A designed $\bar{\mathbf{H}}_1$ index matrix

		Block-column index				
Block-row index	$\bar{\mathbf{H}}_1$	1	2	3	4	5
	1	0	1	2	3	4
	2	0	3	17	55	78
	3	0	0	10	34	133

Table 4-19: A designed $\bar{\mathbf{H}}_2$ index matrix

Block-column index

Block-row index	$\bar{\mathbf{H}}_2$	1	2	3	4	5
	1	0	1	2	3	4
	2	1	0	0	3	0
	3	0	2	1	0	0

Table 4-20: A combined exponent matrix, **E(H)** via CRT

Block-column index

Block-row index	**E(H)**	1	2	3	4	5
	1	0	1	2	3	4
	2	1338	672	112	724	301
	3	0	1115	85	1372	133

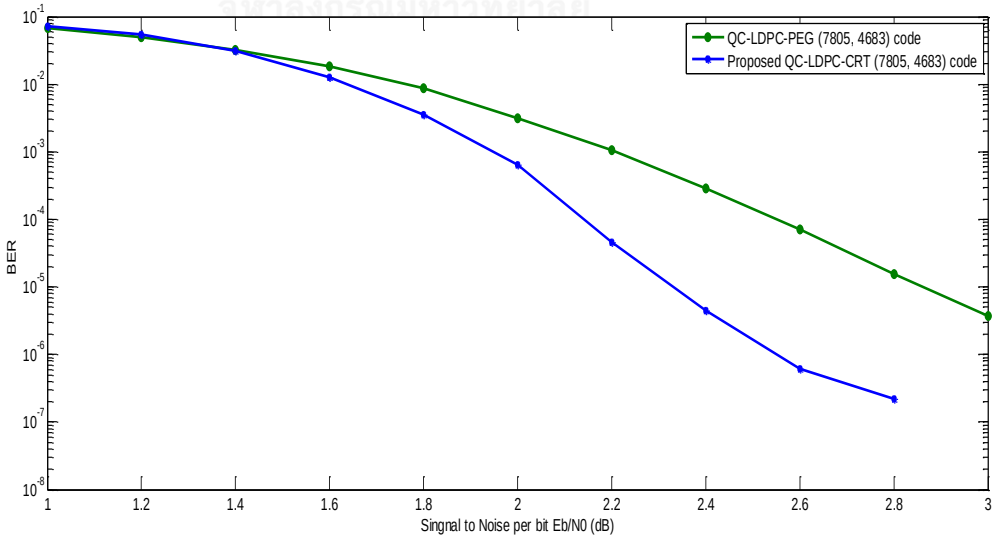


Figure 4-17: BER performance comparison

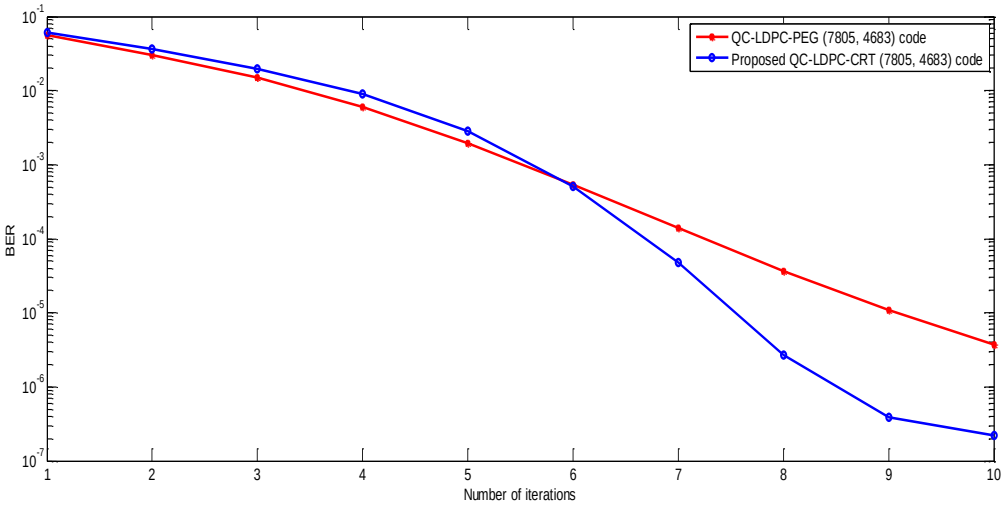


Figure 4-18: BER performance as a function of the number of iterations for different **H** matrices at SNR = 2.8 dB

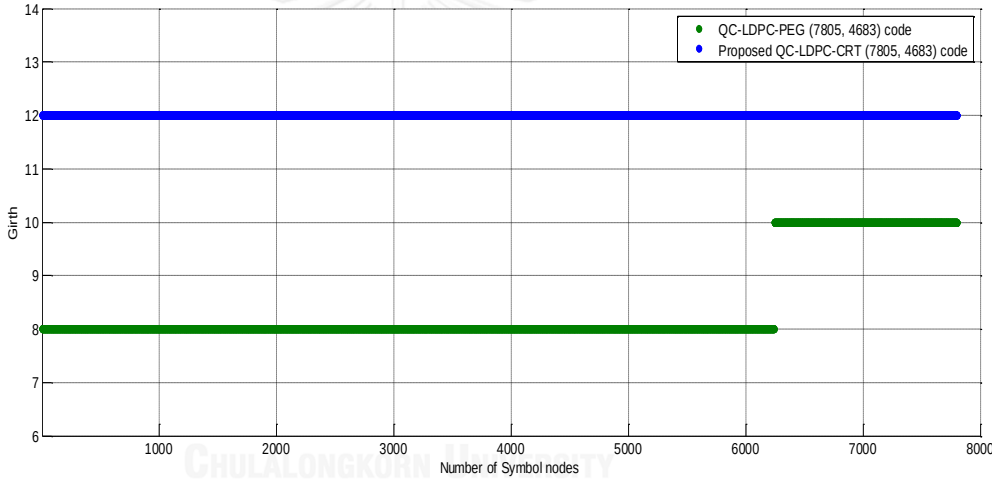


Figure 4-19: Girth comparison of proposed-CRT code

4.5 Properties of the proposed codes

The component QC-LDPC codes which are constructed by using the proposed method when combined with CRT to construct large block-length **H** matrix have good attributes such as large girth, less complexity, good storage, flexible code rates and flexible code lengths. Details of some properties mentioned are discussed in the following section. Furthermore the proposed-CRT LDPC codes have lower computational complexity and are much more practical as compared to those of PEG algorithm based LDPC codes.

4.5.1 Girth

It is one of the well-known parameters determining the performance of decoding. In iterative belief propagation decoding, the algorithm converges to the most optimal solution, if the **H** matrix is free of lower length cycles. Cycles of 4 and 6 leads to undesirable decoded data. When short cycles exist in **H** matrix, the algorithm breaks down very soon. Therefore, the **H** matrix with large girth is particularly taken in to interest. Our algorithm validates as in (4.4), the constructed **H** matrix having girth $g \geq \max\{g_1, g_2\}$ as shown in Figure 4-4.

4.5.2 Complexity

Let us analyze the computational complexity of proposed-CRT algorithms and the storage uses of the proposed-CRT method.

4.5.2.1 Computational Complexity

Computational complexity of the proposed-CRT algorithm principally depends on algorithms exploration time to obtain exponent matrix indices. Exploration time depends on row weight and column weight of desired exponent matrix. In the **H** matrix, the row and column weights are small numbers irrespective of code length. Basically, we can consider the computational complexity of our proposed codes into two categories, namely the first one for obtaining the exponent matrix and the other for combining exponent matrices by using CRT algorithm to obtain the **H** matrix for large block-length QC-LDPC codes. However, both categories depend on the length of the codeword, but the complexity of the combining procedure does not grow much with the size of **H** matrix. For the first category, the complexity of acquiring the exponent matrix relies on the value of CPM size. Specifically, the complexity of the search algorithm increases as the value of the CPM size increases [53, 61, 64, 67]. Then, we compare the complexity of different methods based on the CPM size as shown in Table 4-21. It is apparent that the proposed method requires a smaller CPM size (with a better BER performance) than the other methods as shown in Section 4.4. In addition, once we obtained the exponent codes, the importance of the second category comes into a play. For the second category,

from the CRT formulas as described in Section 4.2 and 4.3, we can see that each CRT computation needs only $(s-1)$ additions, $2(s-1)$ multiplications and 1 modulo operation. Some of the values like $L, \overline{L_b}$ and A_b can be computed prior to the processing algorithm and $L_1, L_2, \cdots, L_s$ should be selected optimally as per Table 4-3. Hence, the complexity of each CRT computation can be negligible if compared to the complexity of designing the parity-check component codes.

Table 4-21: CPM size comparison

Block-column size (k)	Proposed Method		Array codes		QC-PEG codes	
	$g_1 = 6$	$g_1 = 8$	$g_1 = 6$	$g_1 = 8$	$g_1 = 6$	$g_1 = 8$
5	7	17	7	19	7	17
7	7	29	7	31	7	29
9	11	47	11	59	11	47
11	11	61	11	83	11	61

On the other hand, the computational complexity of QC-PEG based codes mainly depends upon the computation load of finding a tree and the elements in sets of $N_{s_j}^l$ or $\bar{N}_{s_j}^l$, which also depend on the column weight and row weight as well as on depth *l*. Usually, a practical regular **H** matrix has a small column weight and a small row weight, irrespective of the code length, and thus the complexity lies on depth *l*, which grows logarithmically [9, 49]. Furthermore, the computation complexity on the worst case of the PEG based QC-LDPC codes are scaled as $O(nm)$. For a fair comparison, we choose the same CPM size as used in our proposed code. It should be noted that as in Section 4.4, for the same CPM size, our proposed code has a better BER performance than the QC-PEG LPDC code as in Section 4.4.

4.5.3 Storage Usage

In a general class of an **H** matrix, row and column indices of '1' entries can be pre-defined and stored in the shift registers for practical applications. Therefore, our proposed-CRT codes method have the significant advantage of storing a smaller

values matrix of less CPM size, as shown $\bar{\mathbf{H}}_1$ and $\bar{\mathbf{H}}_2$ in our examples, discussed in Section 4.4, which has a minimum number of CPM size with ensuring large girth. This may reduce the storage space of the decoder for the proposed-CRT LDPC codes. Moreover, the scope of this method can be expanded in hardware implementation as well [74].

Chapter 5: Proposed Explicit QC-LDPC codes

In this chapter, we introduces two explicit methods for the construction of QC-LDPC codes. The first proposed novel method presents a simple, less computational complexity method for constructing exponent matrix $(3, K)$ of girth 8, 10 and 12 of QC-LDPC based on generation of base matrix. The simulations are shown in comparison with some existing appreciable work.

The second proposed method by the autor deals with a simple, less computational complexity method for constructing exponent matrix $(3, K)$ having girth at least 8 of QC-LDPC codes based on subtraction method. The construction of code deals with the generation of exponent matrix by three formulas. This method is flexible for any block-column length *K*. The codes with girth 8 are constructed with circulant permutation matrix (CPM) size $P \geq max\{a_{2,r},\ a_{2,r},\ldots.a_{2,k}\} + 1$. The details regarding each newly proposed methods is described in the subsequent sections and for illustration purpose each method is also explained with the help of examples.

5.1 A base matrix method to construct column weight 3 Quasi-Cyclic LDPC codes with high girth

This section deals with the construction of exponent or shifting matrix of QC-LDPC codes by base matrix method. Through this method we are able to reduce time complexity for generating **H** matrix by a good amount. The construction of code deals with generation of base matrix by a simple algorithm and element wise element method for girth 8, while only element wise element method for girth 10 and 12. These methods are flexible for block-column length *K*.

5.1.1 Necessary conditions:

There are three easy rules for the generation of base matrix as follow:

1. The first row and the first column of an exponent matrix both are fixed to be a zero vector.
2. It is mandatory that the 2nd row will always be in the ascending order.
3. Repetitions of indices are not allowed, i.e. at different indices we will have different values.

5.1.2 Base matrix generation for girth 8

Let **B** be a base matrix of girth 8 then its CPM size P will be $P \geq \max\{a_{21},\ a_{22,},...,a_{2k}\} + 2$ for any $k\ (0 \leq k \leq K-1)$. For the generation of $(3,K)$ matrix of girth 8, where K is the block-column length. We will follow a simple algorithm as follows:

The 2nd row of the matrix is constructed according to this algorithm:

Step 1: For $(3,K)$ exponent matrix, s will lie between $1 \leq s \leq 3$.

Step 2: We assume that t will be the set of odd number's and for the construction in our example as in section 5.1.5 for constructing $\mathbf{E}(\mathbf{H})$ having size $(3,9)$, the required limit of *t* is up to 9 but for the further increase of block-columns of $\mathbf{E}(\mathbf{H})$, we can move to any size of *t* i.e. $t \in \{1,3,5,7,...,x\}$, where *x* can be any number $x \geq 9$.

Therefore t lies between $1 \leq t \leq x$ and the algorithm goes as follows, keep noted whenever there is a repetition of index due to any value of *s* we will increase *s*, keeping the value of *t* as it is.

Let ***a*** be an empty vector, which will store the indices of the 2nd row, after the working of algorithm i.e. indices after 0 value.

5.1.3 Algorithm for generating $(3,K)$ exponent matrix of girth 8

a=[ ],

for $t = 1:2:x$ do

for $s = 1:3$ do

$\mathbf{E(H)}_{2ndrow} = ((2^{(s)}) \times t) - 1$

$a = [a\ \mathbf{E(H)}_{2ndrow}]$

// To append all elements in "*a*" and to have 2nd row of base matrix as "*a*" //

If $(\mathbf{E(H)}_{2ndrow} \sim = a)$

break;

end

end

end

By using our algorithm, we will first construct $(3,5)$ base matrix and it will be used as a base matrix for $(3,6)$ exponent matrix and similarly for $(3,7)$ exponent matrix $(3,6)$ will act as a base matrix and so on. By our algorithm we have constructed 2nd row of $(3,5)$ base matrix, after that we will find our 3rd row by element wise element method as explained. Now the thing is from which index we have to start, the answer is very simple for $(3,K)$ exponent matrix, we will start substituting number from 1 to CPM size. By our algorithm the indices in 2nd row for $(3,K)$, it's {maximum value (2nd row) +2} will be the size of CPM for girth 8 as explained in necessary condition.

By element wise element method, we mean that we will first put an index at 2nd position of 3rd row leading to obtain a sub matrix of 3×2 and will check it's girth to be 8 and then we will move on to 3rd index of 3rd row for 3×3 sub matrix, by same process we will check its girth and then follow same procedure till 3×5 base matrix.

After generating 3×5 base matrix, our further work will be very time convenient and less complex. By time efficient we really mean our method to work in seconds because the last index of second row for generating consecutive matrix can be obtained by our algorithm. Further work is to just find the last element of 3rd row by element wise element method. Hence it is a very time efficient algorithm, as it can be seen in Section 5.1.5 in *example 5.1*.

5.1.4 Base matrix generation for girth 10 and 12

For the generation of base matrix of girth 10 and 12 we will follow element wise element method with same rules, as in necessary conditions 5.1.1. Let us assume that CPM size is to be $P \leq 30 \times K$. As the first row and the first column is fixed to be a zero vector so we will first generate the 2^{nd} row by element wise element method i.e. 2^{nd} element of 2^{nd} row which leads to a 2×2 sub matrix. After that we will check its girth and will then move to 3^{rd} element i.e. to 2×3 sub matrix so on up to 2×5 sub matrix, by doing this we will find the 2^{nd} row for the base matrix of size $(3,5)$ exponent matrix.

After finding 2^{nd} row we will apply the above mentioned procedure for the generation of 3^{rd} row i.e. firstly construct 3×2 sub matrix and then will check its girth, after that 3×3 sub matrix till 3×5 base matrix. By going through this method we will fix our second row and make all the changes in the third row so as to get the base matrix.

Suppose at any point (after getting 2^{nd} or 3^{rd} index of 3^{rd} row) we are not getting any index in 3^{rd} row by such procedure for desired girth then we will delete that pair of preceding indices and will start the same method from beginning of 3^{rd} row from the original set of indices.

Since in the beginning we have assumed our P size to be $P \leq 30 \times K$, so to get optimized CPM size, we will now decrease its value until we get the desire girth of base matrix. It can be seen in example 2 of section 5.1.5.

Based on our algorithm, all we have to work is on generation of base matrix for girth 8, 10 and 12. For constructing base matrix we have to follow the necessary conditions, algorithm and element wise element method for girth 8, while for girth 10 and 12 we just have to follow element wise element method with the necessary conditions. Thus we will reduce the computational complexity of the program.

5.1.5 Simulation and Results

This section deals with the examples of the above mentioned procedure and also deals with the bit error rate (BER) performance of our algorithm with some well-known existing methods. For computing the BER performance we have considered a

$m \times n$ size **H** matrix, where *n* is the length of a codeword, and *m* is the number of parity bits. The code rate *R* will be $(1-m/n)$. The BER plot based on AWGN channel model, in which a binary input sequence $a_k \in \{0,1\}$ of length $n-m$ bits is encoded and is mapped to *n* bit coded sequence $b_k \in \{\pm 1\}$. After mapping, the received sequence is y_k which is given by $y_k = b_k + n_k$, where n_k stands for AWGN with variance σ^2 and zero mean. A LDPC decoder is used at the receiver end to decode received sequence y_k with 50 iterations by using message passing algorithm.

A minimum of 10000 data packets are used to compute each BER point. Signal to noise ratio (SNR) is defined in decibel as dB. The Mathematical formula for SNR is defined as-

$$\text{SNR} = 10\log_{10}\left(\frac{1}{2R\sigma^2}\right) \tag{5.1}$$

Example 5.1: By using our algorithm, the exponent matrix **E(H)** for the case of $K = 9$ having girth 8 is defined by

$$\mathbf{E(H)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 7 & 11 & 23 & 39 & 55 & 71 \\ 0 & 2 & 6 & 5 & 9 & 13 & 15 & 17 & 19 \end{bmatrix}$$

Table 5-1: Base matrix of size $(3,5)$ for girth 8

Block-row index	**Block-column index**					
	B	1	2	3	4	5
	1	0	0	0	0	0
	2	0	1	3	7	11
	3	0	2	6	5	9

The base matrix is generated by finding the numbers of 2nd row by our algorithm as in Table 5-1. Since the maximum index of 2nd row is 71, so the size of CPM $P = 71 + 2 = 73$ for girth 8. The $(3,9)$ sub matrix will act as a base matrix for $(3,10)$ matrix, we just need to calculate 10th index of 2nd row by our algorithm and regarding the 10th index of 3rd row, and it can be achieved by substituting

numbers from 1 to CPM size *P*. By taking a loop from 1 to size *P* and then substituting it we can easily get the desired index which is 87 (i.e. last element of 2nd row) and 20 (i.e. last element of 3rd row). Thus after getting our base matrix, we just have to find one element that too the last element (i.e. K^{th} index of 3rd row) of $(3, K)$ matrix. Thus all preceding matrix act as a base matrix for the consecutive exponent matrix. Hence by doing this, we are reducing computational complexity of constructing **H** matrix. The comparison of CPM's size of our method with Zhang [75] as shown in Table 5-2, where construction I refer to method II of Zhang [75] and construction II refers to our proposed method.

Table 5-2: CPM's Size compare for girth 8

K	5	6	7	8	9	10	11	12
I	19	27	37	48	61	75	91	108
II	13	25	41	57	73	89	105	121

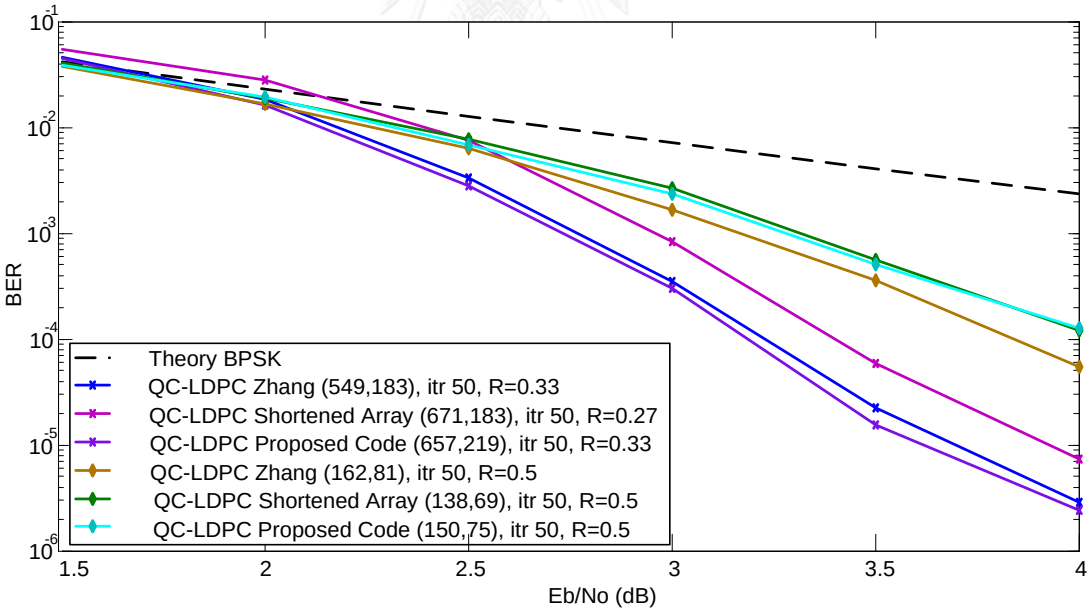


Figure 5-1: BER performance comparison

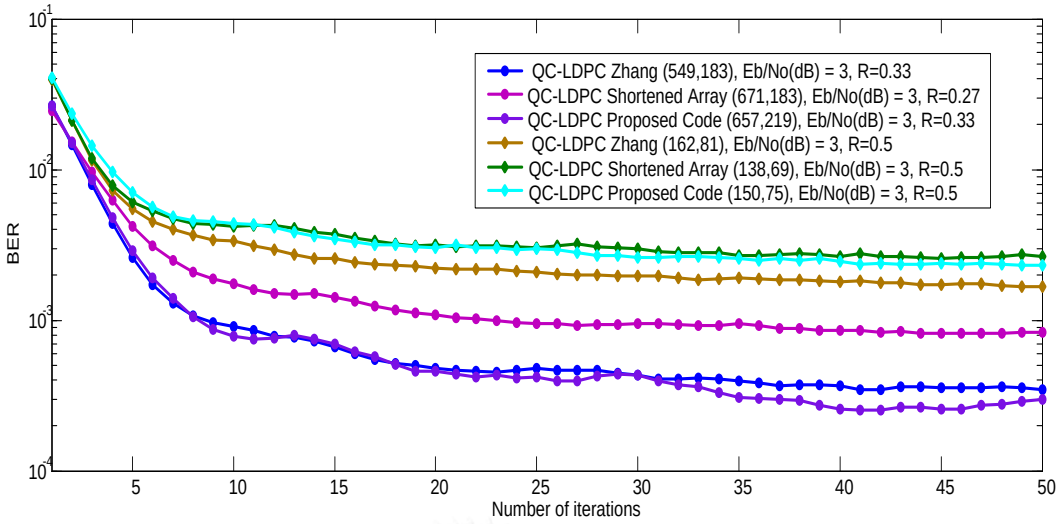


Figure 5-2: BER performance as a function of the number of iterations for different **H** matrices at SNR = 4 dB

Example 5.2: By using the element wise element method the **E(H)** for the case of $K = 7$ having girth 10 is obtained as

$$\mathbf{E(H)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 7 & 12 & 34 & 73 \\ 0 & 18 & 20 & 24 & 65 & 55 & 103 \end{bmatrix}$$

Table 5-3: **E(H)** for $(3,6)$ of girth 10

Block-column index

Block-row index **B**	**1**	**2**	**3**	**4**	**5**	**6**
1	0	0	0	0	0	0
2	0	1	3	7	12	34
3	0	18	20	24	65	51

Hence, it is clearly seen that all the entries of $(3,7)$ **E(H)** are almost same as for its base matrix $(3,6)$ as in Table 5-3, we just need to find the 7^{th} index of 2^{nd} and 3^{rd} row by our element wise element method. The original base matrix of $(3,5)$ for girth 10 was achieved at CPM size $P = 75$ where *P*, is as follow in Table 5-4.

Table 5-4: Base matrix $(3,5)$ of girth 10

Block-row index	Block-column index					
	B	**1**	**2**	**3**	**4**	**5**
	1	0	0	0	0	0
	2	0	1	3	7	12
	3	0	18	20	24	65

Let us assume a case in the 3rd row, like we are at the 4th column index, we have found previous indices but we are not able to find any value at 4th index, so in that case we will delete all the indices before 4th index i.e. from 1 to *P* and will start the process from beginning for the 3rd row. The 3×6 base matrix for the above example, which was achieved at CPM size $P = 91$.

We also compares BER performance as illustrates in Figure 5-1 of the proposed code for different code rates of one third and half code rate respectively, which is compared with some well-known existing QC-LDPC codes such as shortened array codes as in [58] and QC-LDPC codes as described in [75]. Clearly, the proposed algorithm performs better than other algorithms when the SNR is high with reduced construction complexity.

Furthermore, we also compare the BER performance of different schemes as a function of the number of iterations at SNR=4 dB as shown in Figure 5-2. It is apparent that the proposed algorithm converges faster than other compared algorithms at around 50 iterations.

5.2 Subtraction method for girth 8 QC-LDPC codes

This section deals with the construction of exponent or shifting matrix of QC-LDPC codes by subtraction based method. By using this method we are able to reduce time complexity for generating **H** matrix by a good amount.

5.2.1 Essential conditions:

There are three easy rules for the generation of proposed base matrix as follow:

1. The first row and the first column of an exponent matrix both are fixed to be a zero vector.
2. It is mandatory that the 2nd row will always be in the ascending order.
3. Repetitions of indices are not allowed, i.e. at different indices we will have different values.

   For simplicity, we demonstrate $3\times K$ exponent matrix of non-negative integers is expressed as

$$\mathbf{E(H)} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & a_{1,1} & \cdots & a_{1,K-1} \\ 0 & a_{2,1} & \cdots & a_{2.K-1} \end{bmatrix} \tag{5.2}$$

To obtain high girth, we should take care of indices in (2) in order to avoid presence of small cycle. Before exploring more towards proposed construction, we start with following lemma

Lemma 5.1*:* For any $l\left(1\le l\le K-1\right)$ and $k\left(0\le k\le l-1\right)$

$a_{2,l}-a_{2,\,k}\ge a_{1,l}-a_{1,k}$.

Proof: Since-

$$a_{2,l}-a_{2,l-1}>a_{1,K-1}-a_{1,l-1} \tag{5.3}$$

and also $$a_{1,K-1}-a_{1,l-1}\ge a_{1,l}-a_{1,l-1} \tag{5.4}$$

Therefore from (5.3) and (5.4) we have $a_{2,l}-a_{2,k}\ge a_{1,l}-a_{1,k}$.

5.2.2 Formula for constructing matrix of girth 8

Since we have fixed our first row and first column to be a zero vector as in (2), so we have to work basically for only the 2nd row and 3rd row indices. To obtain the 2nd row of our exponent matrix, we replace $a_{1,\,l}=l$, which means $a_{1,\,1}=1$, $a_{1,\,2}=1$ and so on. For attaining the 3rd row, we have to apply the below three formulas so as to get the desired row

$$\begin{aligned} a_{2,1} &= a_{2,0}+a_{1,1}+\left(max\left\{a_{1,1},\,a_{1,2},...,a_{1,K-1}\right\}-a_{1,0}\right) \\ a_{2,K-1} &= a_{2,K-2}+a_{1,K-1}+\left(max\left\{a_{1,1},\,a_{1,2},...,a_{1,K-1}\right\}-a_{1,1}\right) \end{aligned} \tag{5.5}$$

The above two formulas will generate the first non-zero element and the last non-zero element of the 3rd row. In between, the indices can be calculated by the formula as follows for *t* where $(2 \le t \le K-2)$ -

$$a_{2,t} = a_{2,t-1} + a_{1,t} + \left(max\left\{ a_{1,1}, a_{1,2}.....a_{1,K-1} \right\} - a_{1,t} \right) \qquad (5.6)$$

By using the subtraction method we are able to reduce the computational complexity by a very good amount, since we have already fixed our 1st row and 1st column, so the other entries in 2nd row are sequence wise indices from 1 onwards. In the 3rd row the elements can be generated by simple mathematical formulas as in (5.5) and (5.6), which takes less than a second to execute, hence our computational complexity is reduced by a very good amount.

Theorem 5.1: Let $\mathbf{E(H)}$ be a base matrix. For $\mathbf{E(H)}$ to be of girth 8 it's CPM size $P \ge max\left\{a_{2,r}, a_{2,r},\ldots,a_{2,k}\right\} + 1$ for any $k\left(0 \le k \le l-1\right)$.

Proof: For simplicity we will write $\mathbf{E(H)}$ to be $\mathbf{B}$. We will use induction method to justify our proof and the absence of 4 and 6 cycles. To prove it, suppose $P = +\infty$ then 4 cycles cannot exist according to definition of $\mathbf{E(H)}$ (by using theorem 2.1 [7] mod equation will become a normal equation). Now to prove that cycle 6 is also absent we will assume that $\mathbf{B}\left(l-1\right)$ be the current setting of exponent matrix having 0th and 1st row and the first $\left(l-1\right)$ elements of the 2nd row. The new setting is assumed to be $\mathbf{B}\left(l\right)$ which is obtained by adding a new entry $(a_{2,k})$ of 2nd row to be $\mathbf{B}\left(l-1\right)$. We will prove that no 6 cycle exist in $\mathbf{B}\left(l-1\right)$. The proof is by induction method so we will assume that there exists a 6 cycle in $\mathbf{B}\left(l\right)$, so if this exist then there are only two patterns of cycle 6 as in [80]. Let us denotes *u, v* and *w* be the three columns $\left(0 \le u \le P-1, 0 \le v \le u-1, u \ne v, v \ne w\right)$ respectively which form a 6-cycle, as per in *Theorem 1* [58] it is impossible to have 6-cycle if $P \ge max\left\{a_{2,r}, a_{2,r},\ldots,a_{2,k}\right\} + 1$.

5.2.3 Simulation and results

We choose same constraints as we described in Section 5.1.5

Example 5.3: By using the subtraction method proposed in Section 5.2.2, the exponent matrix **B** for block-column length of $K = 9$ having girth 8 is expressed as-

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 9 & 17 & 25 & 34 & 42 & 50 & 58 & 73 \end{bmatrix}$$

The 1st row and 1st column of **B** matrix filled as per our defined necessary conditions in Section 5.2.1. To obtain remaining indices of 2nd row we follow Section III-B. After obtaining indices of 2nd row we move for the indices of the 3rd row, in order to get first non-zero element of the 3rd row which is $9\left(0+1+\left(8-0\right)\right)$ and the last non-zero element of 3rd row is $73\left(58+8+\left(8-1\right)\right)$ according to (5.5), in between indices can be obtained by third formula as in (5.6), for example $a_{2,3} = 25$ is basically $\left(17+3+\left(8-3\right)\right) = 25$ according to **B** matrix and so on. Therefore we can obtain the rest of the indices of the 3rd row. Since the maximum index of the 3rd row is 73 so according to the *Theorem 5.1*, the size of CPM should be $P = 73 + 1 = 74$ for girth 8. In this way, we get our desired exponent matrix, hence **H** matrix, which is having reduced computational time complexity by a good amount.

The Table 3 compares the CPM size of Construction **I** (refer to the method I of Zhang [75]) and Construction **II** based on our proposed algorithm. We can obtain better BER performance while losing lower bound as compared to construction **I**; still, there exists a trade-off between performance and complexity.

Table 5-5: CPM size comparison of the proposed algorithm

K	5	6	7	8	9	10	11	12
I	19	27	37	48	61	75	91	108
II	21	31	43	57	74	91	111	133

We also compares BER performance as illustrates in Figure 5-3 of the proposed code for different code rates of one third and half code rate respectively, which is compared with some well-known existing QC-LDPC codes such as shortened array codes as in [58] and QC-LDPC codes as described in [75].

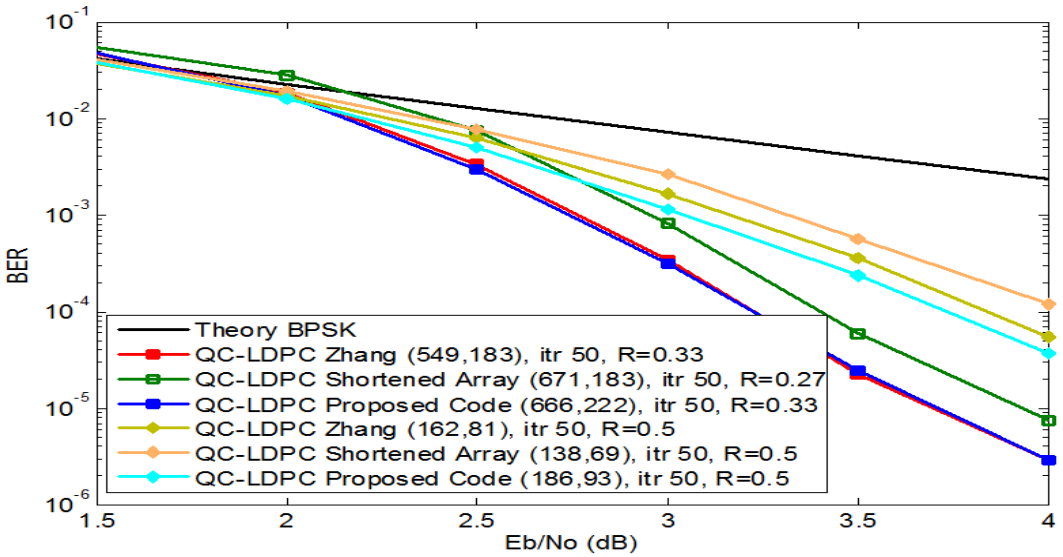


Figure 5-3: BER performance comparisons

Clearly, the proposed algorithm performs better than other algorithms when the SNR is high with reduced construction complexity.

Furthermore, we also compare the BER performance of different schemes as a function of the number of iterations at SNR=3 dB as shown in Figure 5-4. It is apparent that the proposed algorithm converges faster than other compared algorithms at around 50 iterations. Our simulation results can be useful to construct good QC-LDPC codes in less computation time with comparable performance to other applicable existing work in the domain of QC-LDPC codes.

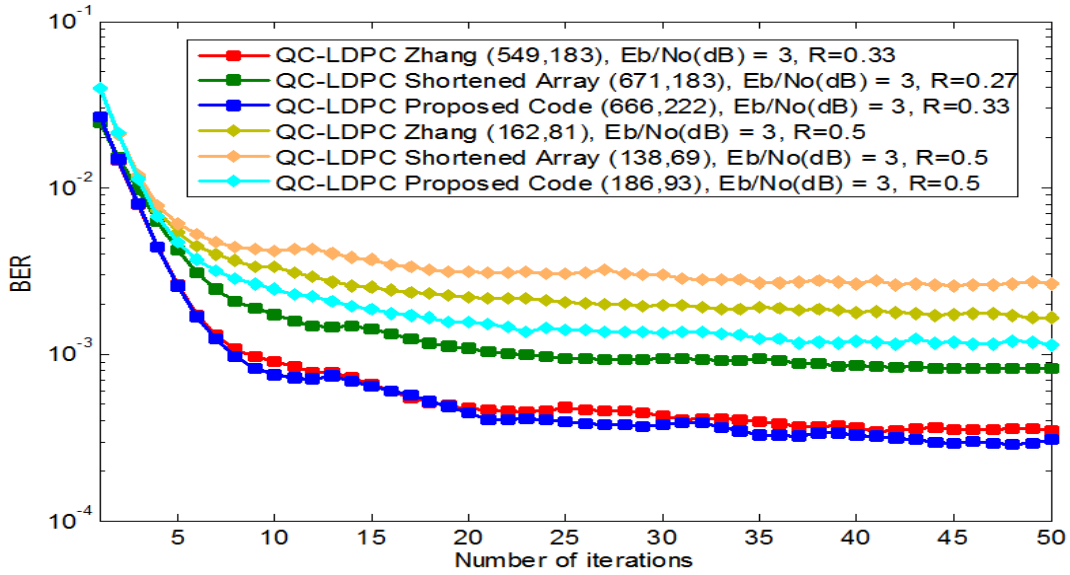


Figure 5-4: BER performance as a function of the number of iterations for different **H** matrices

Chapter 6: Conclusion and Future work

6.1 Conclusion

The main subject of this thesis is to construct a novel QC-LDPC code. We investigated the problems associated with designing algebraic QC-LDPC codes. The proposed codes have definite benefits over traditional randomly constructed codes such as QC-PEG based LDPC codes, particularly when considering medium to large block-length codes. In addition, proposed codes design have guaranteed girth properties, design parameters flexibilities of design parameters, ease of implementation and reduced memory storage requirements as CPM size is less. In Chapter 2 and 3, the LDPC codes and various QC-LDPC codes with their design parameters properties were studied respectively. One of the most vital parameters that affect the performance of finite length codes is the girth, which is emphasized in the methods presented. One of the proposed method as in Chapter 4 applies concept of CRT for constructing medium to large QC-LDPC codes, which ensures less computational complexity. This design also offers guaranteed girth of at least 8, in all the cases.

Three main QC-LDPC code construction methods were proposed based on search criteria with CRT, base matrix generation method and subtraction based method. In the first method, we constructed the **H** matrix of QC-LDPC codes that aims for selecting the indices of the exponent matrix with a maximized local girth for column weight 3, by sequentially assigning proper sub-matrix for each column of **E(H)** matrix. A class of structured regular QC-LDPC codes has been constructed by using a CRT algorithm. This method can also be generalized to any number of column weights. Simulation results show that the proposed code outperforms the well-known algorithms as in [8, 73] in certain cases. Any general case of large block-length LDPC codes with good performance can be constructed using our proposed method. It fulfills almost all the parameters required for good LDPC codes and

suitable for practical applications in terms of cost efficiency. Nevertheless, the author found that the proposed algorithm might require higher computational search time than some existing algorithms. Consequently, one should trade-off between performance and complexity when designing the QC-LDPC codes.

In the second method, we presented a simple, less time consuming construction method for **H** matrix, having girth 8, 10 and 12. The proposed method is based on first obtaining the base matrix, which leads to generate consecutive block-column sub matrices for desired **E(H)** size. The author obtained a class of QC-LPDC codes as explained in Chapter 5. The performance of proposed QC-LDPC codes is simulated in terms of BER and number of iterations, which is comparable to the recent work as in [75]. We have also compared our obtained CPM size for block-column length up to 12 with well-known existing work. The results are helpful in construction of binary regular QC-LDPC codes.

In the third method, we presented a simple, time efficient construction method for **H** matrix, the construction of QC-LDPC codes having girth 8. The choice of block-column length *K* is kept flexible and the method was able to reduce the computational complexity of the **H** matrix by a decent amount. The CPM size *P* can be obtained by adding one to maximum indices of the 2^{nd} row of **E(H)** matrix. We obtained a class of CPM-QC-LPDC codes having girth 8 as explained in our example in Chapter 5. The results are helpful in the construction of binary QC-LDPC codes.

In conclusion, we suggest that the first proposed method can be useful for applications where high girth and large block-length codewords are required; it also suits for the hardware friendly applications as it leads to less CPM size requirement. However, the second proposed method can be useful for medium block-length codewords with girth up to 12, and lastly, the subtraction based method is good for applications that need medium to large block-length QC-LDPC codes with girth 8.

Structured QC-LDPC codes have known properties and performance as compared to the randomly constructed QC based algorithm for an example QC-PEG LDPC code. It represents a large family free from 4-cycle and regular or irregular LDPC codes with simple linear-time encoding. The investigation suggested that considering the coefficient of the exponent matrices by maintaining the QC property

is the best solution to construct **H** matrix. The proposed codes have column weight 3, which is widely adopted in current standards such as IEEE 802.11n and DVB S2/T2. This thesis has addressed extension of search based codes, which can be constructed algebraically and shows these codes can perform superior to already proven structured QC-LDPC codes along with certain random based codes such as QC-PEG LDPC codes. Although random constructions do have the advantage of more flexibilities and comparably good BER performance despite of deterministic structure.

6.2 Future directions (Non-Binary LDPC Codes)

Interestingly, there are many more features of the LDPC codes over GF(q) of continuous research. There are still various questions that need to be answered in terms of encoder complexity issues, decoder design and performance analysis point of view and problems are still left unsolved. In so far work, randomly chosen entities of **H** matrix are considered but it would be possible to further optimize the coefficient for better results in higher order of q. Furthermore, deployment of the encoder and decoder algorithms for LDPC codes over GF(q) are interesting domain that has only started to be explored in the recent years.

For small to moderate block length and higher code rate, a non-binary LDPC code normally provides better performance than a binary LDPC and may also be better suited to some communications channels [76, 77] However, recent interest in non-binary designs such as non-binary protograph-based LDPC codes [78] may provide useful structures for non-binary LDPC codes. In addition, non-binary LDPC codes can be combined with higher order of modulation with ease. Specifically, it can avoid bit to symbol conversion (and vice versa) leading to a potential candidate for various wireless communication with higher order modulation and multi-input and multi-output (MIMO) technique.

In a broader prospective, the field of LDPC codes is huge and well-studied but several areas especially non-binary LDPC codes still offer challenging problems. There would be some features that will be of great interest in specific when applied to our class of QC-LDPC codes. This class of codes will benefit from enhancements of both QC-LDPC codes and GF(q) codes. Their implementation on the optimization of the values of the entries of the exponent matrix will advance with its

performances. Furthermore, it will take gain of improvements on the domain of QC-LDPC codes. Moreover this research targeted to future implement proposed codes for non-binary QC-LDPC codes.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *Information Theory, IRE Transactions on,* vol. 8, pp. 21-28, 1962.

[2] D. J. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics letters,* vol. 32, pp. 1645-1646, 1996.

[3] C. E. Shannon, "A note on the concept of entropy," *Bell System Tech. J,* vol. 27, pp. 379-423, 1948.

[4] D. J. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics letters,* vol. 33, pp. 457-458, 1997.

[5] S.-Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *Communications Letters, IEEE,* vol. 5, pp. 58-60, 2001.

[6] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 5, pp. 3-55, 2001.

[7] M. P. Fossorier, "Quasicyclic low-density parity-check codes from circulant permutation matrices," *Information Theory, IEEE Transactions on,* vol. 50, pp. 1788-1793, 2004.

[8] Z. Li and B. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, 2004, pp. 1990-1994.

[9] S. Lin and D. J. Costello, *Error control coding*: Pearson Education India, 2004.

[10] B. Vasic, K. Pedagani, and M. Ivkovic, "High-rate girth-eight low-density parity-check codes on rectangular integer lattices," *Communications, IEEE Transactions on,* vol. 52, pp. 1248-1252, 2004.

[11] S. Myung and K. Yang, "A combining method of quasi-cyclic LDPC codes by the Chinese remainder theorem," *Communications Letters, IEEE,* vol. 9, pp. 823-825, 2005.

[12] C. Berrou and A. Glavieux, "Turbo codes," *Encyclopedia of Telecommunications*, 2003.

[13] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, pp. 147-160, 1950.

[14] M. J. Golay, "Notes on digital coding," vol. 37, ed, 1949, pp. 657-657.

[15] D. E. Muller, "Application of Boolean algebra to switching circuit design and to error detection," *Electronic Computers, Transactions of the IRE Professional Group on*, pp. 6-12, 1954.

[16] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Information Theory, Transactions of the IRE Professional Group on*, vol. 4, pp. 38-49, 1954.

[17] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, pp. 300-304, 1960.

[18] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and control*, vol. 3, pp. 68-79, 1960.

[19] V. D. Goppa, "Codes on algebraic curves," in *Soviet Math. Dokl.*, 1981, pp. 170-172.

[20] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, 1993, pp. 1064-1070.

[21] G. Forney and G. Forney, "Concatenated codes, vol. 11," ed: Cambridge: MIT press, 1966.

[22] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, pp. 260-269, 1967.

[23] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *IEEE Transactions on Information Theory*, vol. 20, pp. 284-287, 1974.

[24] P. Elias, "Error-free Coding," *Information Theory, Transactions of the IRE Professional Group on*, vol. 4, pp. 29-37, 1954.

[25] Q. Huang, Q. Diao, S. Lin, and K. A. Abdel-Ghaffar, "Cyclic and quasi-cyclic LDPC codes: New developments," in *ITA*, 2011, pp. 186-195.

[26] R. M. Tanner, "A recursive approach to low complexity codes," *Information Theory, IEEE Transactions on,* vol. 27, pp. 533-547, 1981.

[27] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *Information Theory, IEEE Transactions on,* vol. 47, pp. 638-656, 2001.

[28] D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity," *Proceedings of the IEEE,* vol. 95, pp. 1150-1177, 2007.

[29] J. Hagenauer, "The EXIT chart-introduction to extrinsic information transfer in iterative processing," in *Proc. 12th European Signal Processing Conference (EUSIPCO)*, 2004, pp. 1541-1548.

[30] A. Kavčić, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *Information Theory, IEEE Transactions on,* vol. 49, pp. 1636-1652, 2003.

[31] N. Wiberg, *Codes and decoding on general graphs*: Citeseer, 1996.

[32] D. J. MacKay and M. C. Davey, "Evaluation of Gallager codes for short block length and high rate applications," in *Codes, Systems, and Graphical Models*, ed: Springer, 2001, pp. 113-130.

[33] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *Information Theory, IEEE Transactions on,* vol. 47, pp. 585-598, 2001.

[34] Y.-C. He, S.-H. Sun, and X.-M. Wang, "Fast decoding of LDPC codes using quantisation," *Electronics Letters,* vol. 38, p. 1, 2002.

[35] E. Eleftheriou, T. Mittelholzer, and A. Dholakia, "Reduced-complexity decoding algorithm for low-density parity-check codes," *Electronics Letters,* vol. 37, pp. 102-104, 2001.

[36] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Transactions on Information Theory,* vol. 43, pp. 1757-1766, 1997.

[37] D. Gruenewald, "Explicit algorithms for Humbert surfaces," University of Sydney, 2008.

[38] M. E. O'Sullivan, "Algebraic construction of sparse matrices with large girth," *Information Theory, IEEE Transactions on,* vol. 52, pp. 718-727, 2006.

[39] Y. Mao and A. H. Banihashemi, "A heuristic search for good low-density parity-check codes at short block lengths," in *Communications, 2001. ICC 2001. IEEE International Conference on,* 2001, pp. 41-44.

[40] G. Richter, "An improvement of the PEG algorithm for LDPC codes in the waterfall region," in *Computer as a Tool, 2005. EUROCON 2005. The International Conference on,* 2005, pp. 1044-1047.

[41] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *Information Theory, IEEE Transactions on,* vol. 47, pp. 599-618, 2001.

[42] K. Lally and P. Fitzpatrick, "Algebraic structure of quasicyclic codes," *Discrete Applied Mathematics,* vol. 111, pp. 157-175, 2001.

[43] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic LDPC codes for fast encoding," *Information Theory, IEEE Transactions on,* vol. 51, pp. 2894-2901, 2005.

[44] D. J. MacKay, "Good error-correcting codes based on very sparse matrices," *Information Theory, IEEE Transactions on,* vol. 45, pp. 399-431, 1999.

[45] S. J. Johnson and S. R. Weller, "Regular low-density parity-check codes from combinatorial designs," in *Information Theory Workshop, 2001. Proceedings. 2001 IEEE,* 2001, pp. 90-92.

[46] B. Vasic and O. Milenkovic, "Combinatorial constructions of low-density parity-check codes for iterative decoding," *Information Theory, IEEE Transactions on,* vol. 50, pp. 1156-1176, 2004.

[47] S. Ländner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on,* 2005, pp. 630-635.

[48] H. Tang, J. Xu, S. Lin, and K. A. Abdel-Ghaffar, "Codes on finite geometries," *Information Theory, IEEE Transactions on,* vol. 51, pp. 572-596, 2005.

[49] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *Information Theory, IEEE Transactions on,* vol. 51, pp. 386-398, 2005.

[50] D. Mackay. Encyclopedia of Sparse Graph Codes [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html

[51] J. Li and E. Kurtas, "A class of high-rate, low-complexity, well-structured LDPC codes from combinatorial designs and their applications on ISI channels," *Analysis,* vol. 1, p. 1, 2002.

[52] Y. Kou, S. Lin, and M. P. Fossorier, "Low density parity check codes: Construction based on finite geometries," in *Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE,* 2000, pp. 825-829.

[53] Y. Kou, S. Lin, and M. P. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *Information Theory, IEEE Transactions on,* vol. 47, pp. 2711-2736, 2001.

[54] Y. Y. Tai, L. Lan, L. Zeng, S. Lin, and K. A. Abdel-Ghaffar, "Algebraic construction of quasi-cyclic LDPC codes for the AWGN and erasure channels," *IEEE transactions on communications,* vol. 54, pp. 1765-1774, 2006.

[55] R. M. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," *Proc. of ICSTA,* 2001.

[56] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN progress report,* vol. 42, pp. 42-154, 2003.

[57] O. Milenkovic, I. B. Djordjevic, and B. Vasic, "Block-circulant low-density parity-check codes for optical communication systems," *Selected Topics in Quantum Electronics, IEEE Journal of,* vol. 10, pp. 294-299, 2004.

[58] O. Milenkovic, N. Kashyap, and D. Leyba, "Shortened array codes of large girth," *Information Theory, IEEE Transactions on,* vol. 52, pp. 3707-3722, 2006.

[59] H. Song, J. Liu, and B. Kumar, "Low complexity LDPC codes for partial response channels," in *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE,* 2002, pp. 1294-1299.

[60] M. Karkooti and J. R. Cavallaro, "Semi-parallel reconfigurable architectures for real-time LDPC decoding," in *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, 2004, pp. 579-585.

[61] J. Zhang and G. Zhang, "Deterministic girth-eight QC-LDPC codes with large column weight," *Communications Letters, IEEE,* vol. 18, pp. 656-659, 2014.

[62] M. Saadi, A. Bajpai, Y. Zhao, P. Sangwongngam, and L. Wuttisittikulkij, "Design and Implementation of Secure and Reliable Communication Using Optical Wireless Communication," *Frequenz,* vol. 68, pp. 501-509, 2014.

[63] J. L. Fan, "Array codes as LDPC codes," in *Constrained Coding and Soft Iterative Decoding*, ed: Springer, 2001, pp. 195-203.

[64] E. Eleftheriou and S. Ölçer, "Low-density parity-check codes for digital subscriber lines," in *Communications, 2002. ICC 2002. IEEE International Conference on*, 2002, pp. 1752-1757.

[65] W. Singhaudom, S. Noppankeepong, and P. Suphithi, "Design of hign-rate modified array codes for magnetic recording system," in *ECTI International Conference*, 2007.

[66] Y. Zhang and X. Da, "Construction of girth-eight QC-LDPC codes from arithmetic progression sequence with large column weight," *Electronics Letters*, vol. 51, pp. 1257-1259, 2015.

[67] S. V. Ranganathan, D. Divsalar, and R. D. Wesel, "On the girth of (3, L) quasi-cyclic LDPC codes based on complete protographs," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, 2015, pp. 431-435.

[68] H. Fujita and K. Sakaniwa, "Some classes of quasi-cyclic LDPC codes: Properties and efficient encoding method," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 88, pp. 3627-3635, 2005.

[69] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near-Shannon-limit quasi-cyclic low-density parity-check codes," *Communications, IEEE Transactions on*, vol. 52, pp. 1038-1042, 2004.

[70] N. Miladinovic and M. Fossorier, "Systematic recursive construction of LDPC codes," *Communications Letters, IEEE,* vol. 8, pp. 302-304, 2004.

[71] J. Xu, L. Chen, L. Zeng, L. Lan, and S. Lin, "Construction of low-density parity-check codes by superposition," *IEEE transactions on communications,* vol. 53, pp. 243-251, 2005.

[72] Y. Liu, X. Wang, R. Chen, and Y. He, "Generalized combining method for design of quasi-cyclic LDPC codes," *Communications Letters, IEEE,* vol. 12, pp. 392-394, 2008.

[73] X. Jiang and M. H. Lee, "Large girth quasi-cyclic LDPC codes based on the chinese remainder theorem," *Communications Letters, IEEE,* vol. 13, pp. 342-344, 2009.

[74] D. Oh and K. K. Parhi, "Low-complexity switch network for reconfigurable LDPC decoders," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on,* vol. 18, pp. 85-94, 2010.

[75] G. Zhang, R. Sun, and X. Wang, "Several explicit constructions for (3, L) QC-LDPC codes with girth at least eight," *IEEE communications letters,* vol. 17, pp. 1822-1825, 2013.

[76] R.-H. Peng and R.-R. Chen, "Design of nonbinary LDPC codes over GF (q) for multiple-antenna transmission," in *Military Communications Conference, 2006. MILCOM 2006. IEEE,* 2006, pp. 1-7.

[77] S. Song, L. Zeng, S. Lin, and K. Abdel-Ghaffar, "Algebraic constructions of nonbinary quasi-cyclic LDPC codes," in *2006 IEEE International Symposium on Information Theory*, 2006.

[78] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri, "Non-binary protograph-based LDPC codes: Enumerators, analysis, and designs," *Information Theory, IEEE Transactions on,* vol. 60, pp. 3913-3941, 2014.

APPENDIX

Summary of contributions

On the way to realize the designing of an **H** matrix, various contributions have been made in the form of journals and conferences. Here is the list of contribution in the domain of LDPC codes.

- **International Journals**

1. **Ambar Bajpai**, Abhishek Kalsi, Piya Kovintavewat, Lunchakorn Wuttisittikulkij, "**A Base Matrix based method for high Girth Quasi-Cyclic LDPC Codes**" *(Paper to be submitted in IEIE journal).*
2. **Ambar Bajpai**, Gan Srirutchataboon, Piya Kovintavewat, and Lunchakorn Wuttisittikijkulkij, "**A New Construction Method for Large Girth Quasi-Cyclic LDPC Codes with Optimized Lower Bound**" *in Wireless Pers Commun (2016): DOI 10.1007/s11277-016-3465-8 (ISI IF =0.701).*

1. Muhammad Saadi, **Ambar Bajpai**, Yan Zhao, Paramin Sangwongngam, Lunchakorn Wuttisittikulkij, "**Design and Implementation of Secure and Reliable Communication using Optical Wireless Communication**" *in Frequenz 68. 11-12 (2014): 501-509. (ISI IF 0.39).*

- **International Conferences**

1. A. Bajpai, A. Kalsi, L. Wuttisittikulkij, and P. Kovintavewat, "**A greedy search based method with optimized lower bound for high girth QC-LDPC codes**" paper to be submitted in 13th Annual IEEE India Conference (INDICON 2016), Bangalore, 16-18 December 2016.

2. **A. Bajpai**, A. Kalsi, L. Wuttisittikulkij and P. Kovintavewat, "**A subtraction based method to construct column weight 3 Quasi-Cyclic LDPC Codes**", In 12th IEEE

International Siberian Conference on Control and Communications (SIBCON-2016), at Moscow, Russia, 12 May -14 May. 2016.

3. A. Kalsi, **A. Bajpai**, L. Wuttisittikulkij and P. Kovintavewat, "**A Base Matrix method to construct column weight 3 Quasi-Cyclic LDPC Codes with high girth**", In 15th IEEE International Conference on Electronics, Information and Communications (ICEIC 2016), at Danang, Vietnam, 27 Jan. -30 Jan. 2016.

4. **A. Bajpai**, P. Kovintavewat and L. Wuttisittikulkij, "**On the Lifting of Girth for Quasi-Cyclic LDPC Codes using Chinese Remainder Theorem**", In 30th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 15), at Seoul, South Korea, 29 June -2 July 2015.

5. P. Vanichchanunt, **A. Bajpai** and L. Wuttisittikulkij, "**Performance Improvement for IEEE 802.11 LDPC Decoding with Transmit Diversity**", In 30th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 15), at Seoul, South Korea, 29 June -2 July 2015.

6. N. Tuntibut, **A. Bajpai**, P. Kovintavewat and L. Wuttisittikulkij, "**Extended Decoding Performance of LDPC and NB-LDPC Codes with IEEE 802.11n Standard**", In 30th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 15), at Seoul, South Korea, 29 June -2 July 2015.

7. **A. Bajpai**, M. Saadi, T. Phromsa-ard, P. Kovintavewat and L. Wuttisittikulkij. "**Performance and Analysis of Non-Binary LDPC Code Construction Using the Existing PEG Algorithm**" In 29th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 14), at Phuket, Thailand, 1-4 July 2014.

8. G. Srirutchataboon, T. Rattanakritanon, **A. Bajpai**, P. Kovintavewat and L. Wuttisittikulkij. "**A Design of Parity-Check Matrix with Maximized Local Girth for Quasi-Cyclic LDPC codes**" In 29th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 14), at Phuket, Thailand, 1-4 July 2014.

9. M. Saadi, **A. Bajpai**, P. Vonlopvisut, L. Wuttisittikulkij and Y. Zhao. "**A Novel Two Dimensional Visible Light Positioning System Based on Received Signal Strength and Bi-literation**" In 29th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 14), at Phuket, Thailand, 1-4 July 2014.

10. G. Srirutchataboon, **A. Bajpai**, L.Wuttisittikulkij and P. Kovintavewat. "**PEG like algorithm for LDPC codes**" In IEEE conference Proc. iEECON 2014, Thailand, 19-21 Mar 2014.

11. **A. Bajpai**, G. Srirutchataboon, T. Phromsa-ard, Suvit Nakpeerayuth, P. Kovintavewat, and L.Wuttisittikulkij, "**A Study of Non-Binary Low-Density Parity-Check Codes and Its Applications**", In EECON-35, Thailand, 12-14 Dec. 2012.

VITA

Mr. Ambar Bajpai an Indian national and was born on 31th August, 1982 in the city of Banda (Uttar Pradesh), India. He has done his Bachelor of Technology degree from BSA College of Engineering and technology affiliated from UP Technical University Lucknow, India in 2005 in the field of Electronics and Communication Engineering awarded bronze medalist in final year. Afterwards, he completed his Master of Engineering degree from prestigious Birla Institute of Technology and Science (BITS) Pilani, India in 2007 in the field of Communication Systems. He worked as project trainee in ST Microelectronics pvt. Ltd. Bangalore, India with hands on experience of BlueZ 2.0 and Video and image processing. He joined ITM University, Gurgaon India in May 2008, later promoted to Assistant Professor in June 2009 have various responsibility of designing new curriculum, author various lab manuals and research activity. He left ITM University (now North Cap University, Gurugram) in December 2011 for starting new journey of PhD.

Mr. Bajpai started his PhD in semester II of year 2011 from Department of Electrical Engineering, Chulalongkorn University, Thailand. During his PhD, he got various opportunities to work for teaching assistantship in various subjects related to communication systems in International School of Engineering (ISE) and Graduate school for web administrative work. Furthermore, he also volunteered AOTULE meet in Chulalongkorn University.

Mr. Bajpai has successfully published journal and conference papers. He also had an opportunity to serve as a reviewer of ITC-CSCC 2014. His research interest includes Channel Coding, Bluetooth and Visible Light Communication. He is also the recipient of the prestigious 90 years Chulalongkorn Scholarship, overseas presentation scholarship from graduate school to present his work in SibCON-2016 in Moscow, Russia. He also received ITC-CSCC student travel grant 2014.

The career objective of the author is to become a research leader in the area of wireless communication technologies and transfer my knowledge and skills to the future generation of engineering/IT professionals through innovative, research-based teaching. Mr Bajpai is keen of watching Bollywood movies and playing Cricket, badminton, practice yoga and meditation.