SOFT-DECISION DECODING OF MODERATE LENGTH BINARY CYCLIC CODES BASED ON PARITY-CHECK TRANSFORMATION

by

BABALOLA Oluwaseyi Paul

Dissertation presented for the degree of
Doctor of Philosophy in Electronic Engineering in the
Faculty of Engineering at Stellenbosch University

Supervisor: Prof. Daniel Jaco J. Versfeld
Co-supervisor: Dr. Olayinka O. Ogundile

March 2020

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own original work, and that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

BABALOLA P. Oluwaseyi

March 2020

Abstract

Soft-decision decoding of Moderate Length Binary Cyclic Codes based on Parity-check Transformation

O.P. BABALOLA

Department of Electrical and Electronic Engineering, University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Dissertation: PhD (Electronic Engineering)

March 2020

This thesis focuses on obtaining low complexity soft-decision (SD) decoding of binary cyclic codes with coding performance close to the optimal decoding algorithm. The belief propagation (BP) algorithm is commonly used to obtain near-optimal decoding but inappropriate for high-density parity-check (HDPC) codes. Therefore, alternative solutions such as the adaptive belief propagation (ABP) algorithm and the parity-check transformation algorithm (PTA) have been proposed in the literature, based on matrix transformation, to effectively apply the BP decoding for HDPC codes.

The extended parity-check transformation algorithm (EPTA) is introduced in this thesis to obtain a transformed parity-check matrix for the class of binary cyclic (BC) codes. The EPTA reduces the computational complexity of the known adaptive belief propagation (ABP) algorithm. However, it requires more iterative processes to attain comparable results to the ABP. Hence, a generalized parity-check transformation (GPT) algorithm for iterative SD decoding of the class of BC codes is developed. The proposed GPT algorithm is motivated by the EPTA and the belief propagation. The algorithm utilizes a new approach of matrix transformation to overcome the limitation with the BP algorithm for HDPC codes. The transformed matrix is obtained by permuting the columns of the initial parity-check matrix based on the reliability information received from the channel. Results show that the GPT offers a significant performance gain when compared with the hard decision Berlekamp-Massey (B-M) and belief propagation (BP) algorithms. It also produces a reasonable performance gain as compared with other iterative SD decoders. An important feature of the decoder is that it functions within a practical decoding time complexity and can be generally implemented for the class of linear block codes.

Furthermore, a perfect knowledge model is developed to verify the optimality of all BP based algorithms for HDPC codes. The PKM computes a list of candidate matrices based on the prior knowledge of the transmitted codeword and it selects the best parity-check matrix according to a distance metric. The selected matrix is optimal since it minimizes the probability of error over various choices in the list. As a result, we show that for a given channel condition, the conventional transformed matrix, obtained by Gaussian reduction, is sub-optimal and will not necessarily contain unitary weighted columns at corresponding columns of the unreliable bits. Here, there exist specific scenarios where this matrix is not the same as the selected matrix from the PKM, giving room for improvement in the matrices of the BP in general. More so, the model can be used to verify performances of newly developed iterative SD decoders based on parity-check equations.

In conclusion, the discovery of this thesis is important as it proposes a reduced computational time complexity soft-decision decoder for algebraic block codes. In view of some studies where the potentials of these coding techniques have been successfully demonstrated for cellular telephony, remote radio, spread spectrum communications, and satellite transmissions, the generalized parity-check matrix transformation algorithm can be implemented as a real-time decoder in order to reduce the number of transmission errors in digital communications.

UNIVERSITEIT STELLENBOSCH

Opsomming

Sagte-besluit Dekodering van Matige Lengte Binre Sikliese Codes gebaseer op Parity-check-Transformasie

O.P. BABALOLA

Departement Elektriese en Elektroniese Ingenieurswese, Universiteit van Stellenbosch, Privaatsak X1, Matieland 7602, Suid Afrika.

Verhandeling: PhD (Elektroniese Ingenieurswese)

Maart 2020

Hierdie tesis fokus op die verkryging van lae-kompleksiteit sagte-besluit (SD) dekodering van binre sikliese kodes met koderingsprestasie naby die optimale dekodering salgoritme. Die geloof svermeerderingsalgoritme word algemeen gebruik om byna-optimale dekodering te verkry, maar onvanpas vir HDPC-kodes met 'n ho digtheid. Daarom is alternatiewe oplossings soos die ABP-algoritme (adaptive belief propagation) en die pariteitstjek transformering salgoritme (PTA) in die literatuur voorgestel, gebaseer op matriks-transformasie, om die BP-dekodering effektief toe te pas vir HDPC-kodes.

Die uitgebreide pariteitstjek transformering salgoritme (EPTA) word in hierdie tesis bekendgestel om 'n getransformeerde pariteitstjek matriks vir die klas binre sikliese (BC) kodes te verkry. Die EPTA verminder die berekeningskompleksiteit van die bekende algoritme vir adaptive belief propagation (ABP). Dit vereis egter meer iteratiewe prosesse om vergelykbare resultate met die ABP te bereik. Gevolglik word 'n veralgemeende algoritme vir pariteitstjek transformasie (GPT) vir iteratiewe SD-dekodering van die klas BC-kodes ontwikkel. Die voorgestelde GPT-algoritme word gemotiveer deur die EPTA en die uitbreiding van geloof. Die algoritme gebruik 'n nuwe benadering van matriks-transformasie om die beperking met die BP-algoritme vir HDPC-kodes te oorkom. Die getransformeerde matriks word verkry deur die kolomme van die aanvanklike pariteitstjek matriks toe te laat, gebaseer op die betrouba arheidsinligting wat van die kanaal ontvang word. Resultate toon dat die GPT 'n beduidende prestasieverbetering bied in vergelyking met die harde besluit Berlekamp-Massey (B-M) en geloofs propagasie (BP) algoritmes. Dit lewer ook 'n redelike prestasieverbetering in vergelyking met ander iteratiewe SD-dekodeerders. 'N Belangrike kenmerk van die dekodeerder is dat dit binne 'n praktiese dekodering stydkompleksiteit funksioneer en in die algemeen gemplementeer kan word vir die klas linere blokkodes.

Verder word 'n perfekte kennismodel ontwikkel om die optimaliteit van alle BP-gebaseerde algoritmes vir HDPC-kodes te verifieer. Die PKM bereken 'n lys kandidaat matrikse op grond van die voorafkennis van die oordraagbare kodewoord en kies die beste matriks-toetsmatriks volgens 'n afstandmetriek. Die geselekteerde matriks is optimaal, aangesien dit die waarskynlikheid van foute as gevolg van verskillende keuses in die lys verminder. As gevolg hiervan, wys ons dat die konvensionele getransformeerde matriks, verkry deur Gaussiese reduksie, vir 'n gegewe kanaaltoestand sub-optimaal is en nie noodwendig eenheidsgeweegde kolomme by ooreenstemmende kolomme van die onbetroubare stukkies sal bevat nie. Hier bestaan spesifieke scenario's waar hierdie matriks nie dieselfde is as die geselekteerde matriks uit die PKM nie, wat ruimte gee vir verbetering in die matrikse van die BP in die algemeen. Meer nog, die model kan gebruik word om prestasies van nuut ontwikkelde iteratiewe SD-dekodeerders te verifieer gebaseer op pariteitstjek vergelykings.

Ten slotte is die ontdekking van hierdie proefskrif belangrik, aangesien dit 'n verminderde berekeningstyd vir sagte besluitneming vir algebraese blokkodes voorstel. In die lig van enkele studies waar die potensiaal van hierdie koderingstegnieke suksesvol aangetoon is vir sellulre telefonie, afstandradio, verspreide spektrumkommunikasie en satellietuitsendings, kan die veralgemeende algoritme vir pariteitstjekmatriks transformasie as 'n intydse dekodeerder gemplementeer word. om die aantal transmissiefoute in digitale kommunikasie te verminder.

I dedicate this to God the Father, the Son, and the Holy Spirit.

Also, I dedicate this to my Wife (Toluwalope M. Babalola) and my lovely Daughters (Oluwadarasimi Marie and Oluwadamilola Michelle Babalola).

They have always supported and trusted my ability to achieve the best in whatsoever I do.

Acknowledgements

I sincerely appreciate and thank my supervisor, Prof. D.J.J. Versfeld and my co-supervisor, Dr. O.O. Ogundile for their mentorship and support throughout the period of my research. They stood by me when the going was tough and ensured I successfully completed this work. I shall be forever grateful.

Special thanks to my parents (Prof. and Mrs. J.B. Babalola) for the moral, spiritual, and financial supports they gave throughout the period of my studies. They thought me to know that the best investment in life is education. I say thank you.

My profound gratitude goes to Prof. H.A. Engelbrecht and the MIH Media Laboratory for the financial support to the completion of this research.

I am extremely grateful to my friends, colleagues, members of the Association of Nigeria Students in Stellenbosch University and the Deeper Life Campus Fellowship, Stellenbosch University that have extended all kind of help and prayers for accomplishing this work.

Finally, I appreciate the moral support received from my siblings, you guys are the best.

Contents

List of Figures

List of Tables

List of Abbreviations

AWGN Additive White Gaussian Noise

HD hard decision

SD soft decision

ML maximum likelihood

FEC forward error correction

LDPC low density parity-check

RS Reed-Solomon

BCH Bose-Chaudhuri-Hocquenghem

BP Belief propagation

MAP maximum a posteriori

HDPC high density parity-check

GPCM generalized parity-check matrix

ABP adaptive belief propagation

PTA parity-check transformation algorithm

MDS maximum distance separable

KV Koetter and Vardy

GPT generalized parity-check transformation

MATLAB Matrix Laboratory

LRPs least reliable positions

MRPs Most reliable positions

DVB-S2 digital video broadcasting

EPTA extended parity-check matrix transformation algorithm

PKM perfect knowledge model

PCEs parity-check equations

BPSK binary phase shift keying

iid independent and identically distributed

pdf probability density function

LHS left hand side

LLR log likelihood ratio

FHT Fast Hadamard Transform

FFT Fast Fourier Transform

eHDD enhanced hard-decision decoding

GMD Generalized Minimum Distance

CGA Chase and GMD algorithms

SNR signal-to-noise ratio

OSD ordered statistics decoding

GS Guruswani and Sudan

APP a posterior probability

SPA sum-product algorithm

BMA box and match algorithm

RRD random redundant soft-decision decoding

MBBP multiple-bases belief-propagation

PBP permuted belief propagation

BM Berlekamp-Massey

BER bit error rate

Thesis Output

The results and analysis from this thesis have been submitted and published in the following peer-reviewed conference and journals:

[1] O. Babalola and J. Versfeld, "Iterative Soft-Decision Decoding of Binary Cyclic Codes Based on Extended Parity-Check Transformation Algorithm," in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, 2018. DOI: 10.1109/ccece.2018.8447536

[2] O.P. Babalola, O.O. Ogundile and D.J. Versfeld, "A Generalized Parity-check Transformation for Iterative Soft-Decision Decoding of Binary Cyclic Codes," in *IEEE Communications Letters*, 2019. DOI: 10.1109/LCOMM.2019.2956935

[3] O.P. Babalola, O.O. Ogundile and D.J. Versfeld, "A baseline parity-check matrix for iterative soft-decision decoding of binary cyclic codes," *Submitted for publication in IET Communications*, 2019.

Chapter 1

Introduction

Coding theory is the study of techniques to efficiently and accurately transfer information from one end to another. It has different applications such as reducing the power that is needed to reliably transmit a bit of information in a digital communication system, optimize the number of bit per square-inch in a storage device, cryptography, mathematical games, among others. The field of error coding dates back to the mid-twentieth century where completely combinatoric, and discrete approaches were used to detect and correct errors. However, in recent years, there is an improvement in this field by the construction of codes that can attain the Shannon channel capacity [2].

The digital communication system contains functionality that performs different actions on a message as indicated in Figure 1.1. On the transmitter side, the message sequence from the source form a discrete encoded sequence known as the codeword by structurally adding redundant information to the message bits. The encoder functions by taking a block of k message input bits/symbols to produce a block of $n > k$ bits/symbols as its output, which allows the channel errors to be corrected. Discrete bits/symbols are inappropriate for physical channel transmission. Therefore, the modulator converts each output bit/symbol of the encoder to signals that are appropriate for channel transmission. The modulated signal is corrupted due to the condition of the channel. For instance, noise may be added to the signal, or

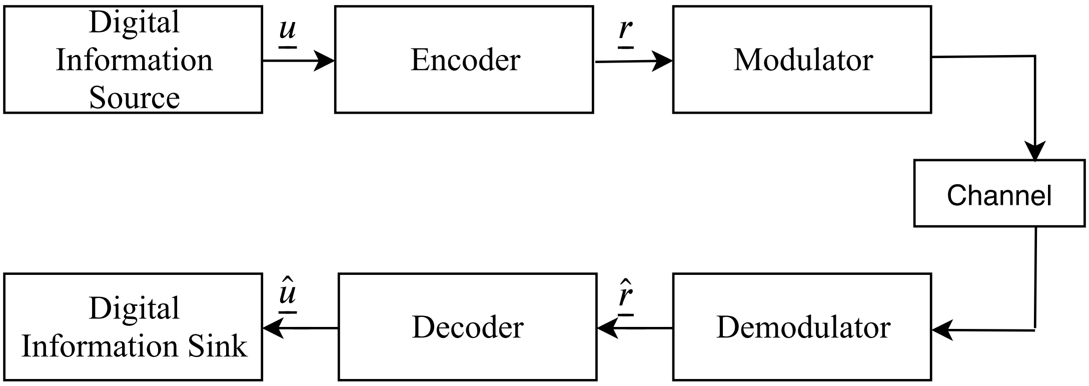

FIGURE 1.1: Simple block diagram of a digital communication system [1]

attenuation due to the transmission range or carrier offset. It may even experience interference from other channels and lots more. Therefore, channels are represented by mathematical models to enable proper analysis. The Additive White Gaussian Noise (AWGN) channel is assumed throughout this thesis since they represent a basis for the majority of the complex channel models. The demodulator operates on each received signal to generate either a discrete (quantized) or a continuous (unquantized) received sequence, which are sent to the decoder. Thus, the decoder utilizes the redundancy provided by the encoder and the noise characteristic of the channel to correct the errors as much as possible. Note that the decoder can combine the demodulation, equalization and decoding processes, as demonstrated in [3, 4].

There are different decoders for the class of linear codes obtained from the literature. The decoders are categorized into hard decision (HD) and soft decision (SD). Hard decision decoding samples, quantize and converts the received signal from the channel to a vector with similar bits/symbols as the channel input. Most HD decoding algorithms estimate the transmitted signal based on the Hamming distance of the code. However, the soft-decision decoder directly utilizes the unquantized received signal from the channel output and estimates the transmitted bit/symbol vector based on the Euclidean distance of the code. Although the soft-decision decoding accepts continuous-valued inputs that make the decoder difficult to implement, the SD algorithm provides a significant improvement in performance compared to the HD decoding algorithms [1].

Both the HD and SD decoders generate an approximation $\hat{\underline{u}}$ of the sequence of information $\underline{u}$ at the output based on the received sequence $\underline{r}$. Similarly, an approximation $\hat{\underline{c}}$ of the codeword $\underline{c}$ can be generated using $\underline{r}$, since there is a one-to-one relationship between the information sequence and the codeword. Therefore, the estimated codeword $\hat{\underline{c}}$ for each possible received vector $\underline{r}$ is chosen using a decision rule. Suppose the codeword $\underline{c}$ was transmitted, the decoder produces a decoding error whenever $\hat{\underline{c}} \neq \underline{c}$. Given that $\underline{r}$ is received as the decoder's input, the decoder has a conditional probability of error $P(E|\underline{r}) \triangleq P(\hat{\underline{c}} \neq \underline{c}|\underline{r})$ [1]. Thus, the decoder's probability of error is given as $P(E) = \sum_{\underline{r}} P(E|\underline{r})P(\underline{r})$, where $P(\underline{r})$ signifies the probability of $\underline{r}$. The $P(\underline{r})$ does not dependent on the decoding rule used. The maximum likelihood (ML) decoding has been proved to be the optimal decision rule in [5, Theorem 1.1 on p. 64]. The ML decision rule minimizes the probability of error by selecting $\hat{\underline{c}}$ as that value of $\underline{c}$ which maximizes $P(\underline{c}|\underline{r})$, where the possible values of $\underline{c}$ are those in the signal constellation $\mathcal{S}$. That is, $\underline{c}$ is selected as the most likely codeword given that $\underline{r}$ is received.

Furthermore, Shannon [2] determined the capability of a noisy channel to reliably transmit information. The Shannon's theorem (noisy channel coding theorem) establishes that every channel has a capacity C, and that for any rate $R < C$, there exist codes with rates that enable the decoding probability of error $P(E)$ to become arbitrarily small using ML decoding. For block codes of fixed-rate $R < C$, long block lengths n are needed to achieve very low error probabilities, such that the bound $P(E) \leq 2^{-nE_b(R)}$ holds, where E_b is the energy-per-information bit [1]. This requires that there must be a very large number of codewords 2^k. Hence, the number of computations required by the ML decoder becomes very high as n increases, since ML decoding has to calculate $\log P(\underline{r}|\underline{c})$ for each codeword before choosing the codeword that produces the maximum. Consequently, achieving very low error probabilities become impracticable using ML decoding. Thus, the problem of communicating over noisy channels at rates close to the Shannon limit can be approached from the perspective of constructing good long codes with performances satisfying the ML decoder, and obtaining techniques for generating estimates of the transmitted codeword from the received sequence of the noisy channel with reduced decoding complexity.

With regards to code design and decoding, forward error correction (FEC) schemes such as low density parity-check (LDPC) block codes, Turbo codes, *Reed-Solomon (RS)* codes, BCH codes have been used to improve the reliability of information sent over the channel. FEC utilizes error-correcting codes that automatically correct detected errors at the receiver. Cyclic codes, such as the BCH and RS codes, are an important subclass of linear block codes that are commonly used in digital communications for reliable data transmissions. The iterative belief propagation principle has been proposed as one of the several methods for SD decoding of cyclic codes.

When the BP algorithm is applied to codes characterized by sparse parity-check matrix such as the LDPC codes, Gallager [6] showed these codes are capable of approaching the Shannon limit. In this thesis, attention is focused on using cyclic codes to improve the reliability of the information in a communication system. The cyclic codes are specifically efficient for error detection [1] due to the considerable inherent algebraic structure of the code. Additionally, the study centers around binary cyclic codes but can be extended to the class of general linear block codes.

1.1 Motivation and Research Question

Low complexity decoding of linear block codes with coding performance close to optimal ML decoding [5] is an open problem in error correction. In practice, most coding and decoding methods proposed up to date are constrained by the decoder's complexity. Hence, lots of research is continuously conducted to design efficient decoding techniques, in terms of error rate performance and computational complexity.

Belief propagation (BP) is a soft-decision decoding algorithm, which is generally used to obtain a near-optimal decoding performance for linear block codes defined over the sparse parity-check matrix. The algorithm operates on Tanner graphs, such that the variable and check nodes correspond to the code bits and parity equations respectively. More so, it was shown in [7] that iterative maximum a posteriori (MAP) algorithm used for decoding Turbo codes is a specific instance of the BP algorithm, which yields a near ML decoding performance. Nevertheless, a well-known problem with the BP algorithm is that it exhibits poor performance when applied to codes with high density parity-check (HDPC) matrix. The HDPC codes, such as the BCH and RS codes, contain short cycles in the associated Tanner graph and irregular row and column weight distributions in the graph [8]. Consequently, reduces the efficiency of the BP algorithm.

However, several techniques that make the BP algorithm applicable to the parity-check matrix of the algebraic codes have been proposed in the literature. For instance, the addition of rows to the parity-check matrix of cyclic codes to obtain a regular Tanner graph of the code was discussed in [9]. Also, a generalized parity-check matrix (GPCM) was proposed to minimize the number of 4-cycles for short length and low rate codes in [10]. The authors in [11] further investigated the GPCM and presented an algorithm that removes 4-cycles in the Tanner graphs after a finite number of iterations. These techniques involve the addition of supplementary bits that do not coincide with the transmitted bits. Consequently, it reduces the decoding performance of the algorithms.

In the quest to obtain a suitable matrix for the BP decoding, the ABP algorithm was introduced in [12–14], which transforms the parity-check matrix of a code based on Gaussian elimination, according to the bit reliabilities at each iteration. For the class of a binary cyclic code $\mathcal{C}(n, k)$ of length n and dimension k, the Gaussian elimination approach reduces $(n-k)$ columns of the parity-check matrix to single weight columns, which correspond to the $(n-k)$ unreliable bits, while the remaining densely weighted k columns coincide with the reliable bits. This transformation approach improves the convergence rate of the iterative decoder with reasonable complexity. The foremost problems with the ABP are the facts that the Gaussian elimination process may not be completely successful for transforming the matrix of non-MDS codes, and the transformation technique may be inappropriate for real-time implementation.

More recently, a symbol level parity-check transformation algorithm (PTA) [15] is proposed for the class of maximum distance separable (MDS) codes with reduced decoding complexity. The PTA is similar to the ABP algorithm as it transforms the parity-check matrix based on the bit reliabilities at each decoding iterations. Unlike the ABP, the PTA requires matrix inversion to obtain the transformed matrices for MDS codes. The algorithm exhibits improved decoding performances compared to the Koetter and Vardy (KV) algorithm [16] and the traditional algebraic HD decoding algorithm [17] for RS codes, since matrix inversion is always guaranteed for this class of MDS codes. Unfortunately, for non-MDS codes such as the BCH codes, the transformation fails whenever a linearly dependent column of the matrix is not reducible to an identity form. As such, the performance of the PTA reduces for non-MDS codes due to the use of partially transformed parity-check matrix at each iteration.

The matrix inversion of the PTA has not been applied to the class of non-MDS codes, such as the BCH code, since matrix inversion is only guaranteed for MDS codes. Therefore, it is foreseen that extensive research on transforming the parity-check matrix at each decoding iteration to obtain the best matrix will improve the performance of the SD decoding algorithm for cyclic codes.

Accordingly, in this thesis, the broad research question is stated as:

How can the error performance of iterative soft-decision decoders be enhanced based on the parity-check matrices of cyclic codes?

1.2 Research Hypotheses

The following five key hypotheses are made in order to start investigations and to answer this research question.

Hypothesis 1 (H1). Implementing the PTA will yield poor performance in comparison to the algebraic HD decoding for non-MDS codes such as the binary cyclic codes.

Hypothesis 2 (H2). Relaxing the PTA's matrix transformation condition will enhance the decoding performance of the PTA for non-MDS codes.

Hypothesis 3 (H3). Permuting the parity-check matrix of cyclic codes based on the reliability information will produce a better performance compared to the other iterative SD decoding algorithms.

Hypothesis 4 (H4). The transformed parity-check matrix for BP decoding of cyclic codes is suboptimal, and will not usually contain unitary weighted columns at corresponding columns of unreliable bits.

Hypothesis 5 (H5). Obtaining an optimal parity-check matrix with respect to a given distance metric will enhance performance and reduce the maximum number of iterations needed for iterative decoding of cyclic codes.

1.3 Research Aim and Objectives

The design of a reduced computational complexity decoding algorithm to achieve a very low probability of error is of significant importance in real-time applications. Therefore, this study aims to enhance the efficiency of the iterative soft-decision decoding algorithm, in particular the PTA, for the class of binary cyclic codes with coding performance close to the ML decoding. To achieve the aim of this thesis, the following objectives are highlighted.

1. *To analyze and show that the PTA fails for non-MDS codes.*
2. *To develop an enhanced parity-check matrix transformation technique based on the existing symbol level PTA.*
3. *To develop a GPT algorithm for iterative SD decoding of cyclic codes.*
4. *To optimize the decoding complexity of the GPT for cyclic codes.*
5. *To model and simulate the proposed algorithms and analyze their performances using Matrix Laboratory (MATLAB) software.*
6. *To perform a comparison study of the results in step 5 with existing methods in the literature.*

1.4 Research Argument

The BP algorithm ensures that LDPC codes approach maximum likelihood performance while maintaining low complexity compared to ML decoding [18]. BP decoding operates on Tanner graphs, which are bipartite graphs with variable nodes and check nodes corresponding to code bits and parity-check equations respectively. The variable node v_i and the check node z_j will only have connected edge whenever the associated parity-check matrix to the Tanner graph has a participating bit 1 at position (j, i). Consider applying the BP algorithm to the HDPC matrices of binary cyclic codes that consist of short cycles in the related Tanner graph. The BP algorithm exhibits poor performance on the HDPC codes at each iteration. Thus, the matrix must be adapted to produce a graph representation for the code, which is suitable for BP decoding. The popular ABP and PTA transform the parity-check matrix to effectively apply BP decoders to binary cyclic codes based on bit reliability, according to Gaussian elimination and matrix inversion respectively. Therefore, an efficient way of deriving the reliability information is of great importance to enhance the performance of the iterative SD decoder. This thesis utilizes the PTA methods of obtaining the reliability information presented in [15].

Additionally, results of the parity-check equations are used to refine the initial reliability matrix. Thus, carefully transforming the parity-check matrix will further reduce the error probability of the estimated codeword as it is derived from hard decision detection on the reliability matrix. Generally, the transformed parity-check matrix is formed in such a way that the unreliable bits correspond to a sparse submatrix. This thesis proves that the PTA transformation method fails for binary cyclic codes whenever a linearly dependent column of the matrix cannot be reduced to an identity form. That is, the submatrix of the parity-check matrix will not be invertible. Chapters 3 and 4 show results of implementing the PTA for non-MDS codes, subsequently validating Hypothesis 1. In this regards, an extended parity-check matrix transformation technique is introduced in Chapter 3, where a relaxed condition is introduced on the transformed parity-check matrix of the code to prevent the PTA from failing at the matrix inversion step. The finding of this chapter verifies Hypothesis 2.

Furthermore, a generalized parity-check matrix transformation algorithm is developed in Chapter 4 to overcome the PTA limitation and to provide an efficient decoding algorithm for cyclic codes. The proposed algorithm permutes columns of the initial parity-check matrix using the reliability information from the channel. Performance evaluations indicate that the newly developed algorithms produce better performance compared to the other known decoding algorithms for binary cyclic codes. Results in Chapter 4 verify the performance analysis and confirm Hypotheses 1 and 3 of this thesis.

Moreover, the existing transformation methods of ABP and PTA reduces all the $(n-k)$ columns of the parity-check matrix at the corresponding least reliable positions (LRPs). This is always the case for MDS codes. But for binary cyclic codes, it is not certain that the LRPs will correspond to the independent columns of the matrix. Thus, the process attempts to reduce the next $n-k+1$ reliable position to weight one column and continues till all the $n-k$ independent columns are reduced. Consequently, the results of ABP, PTA, and poposed GPT, which permutes the columns of the matrix based on bit reliability do not achieve the maximum likelihood decoding performance. In this regards, it is important to analyze the performance of the transformed matrix for BP decoding of binary cyclic codes. Chapter 5 analyses the performances of the transformed matrices using a perfect knowledge model. The model is introduced to generate a baseline parity-check matrix, which is optimal (best) since it minimizes the probability of error over various choices of matrices. The results in Chapter 5 assert Hypotheses 4 and 5 of this thesis.

In summary, this study explored these five hypotheses to develop a performance efficient and low complexity decoder based on parity-check matrix transformation for cyclic codes. The outcome of this study as presented in Figure 1.2 shows that combining these hypotheses answer the thesis research question. This implies that transforming the parity-check matrix based on a refined reliability information improves the performance of the BP decoding algorithm for cyclic codes. The three dotted

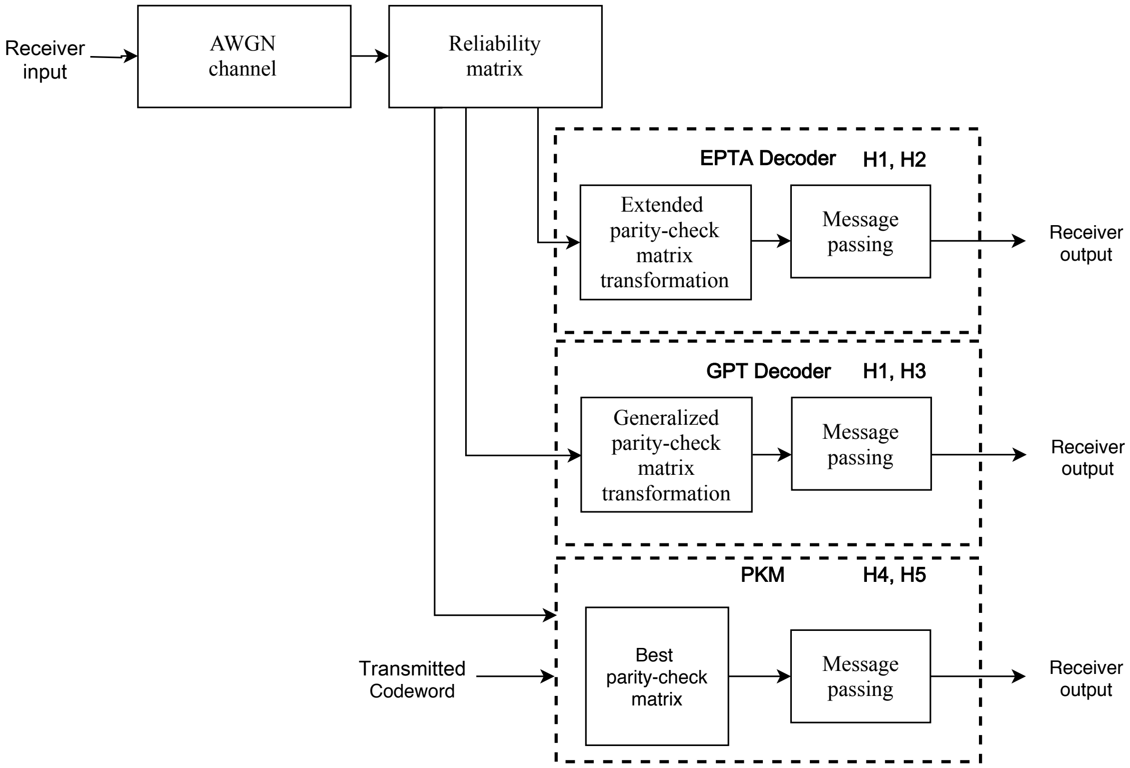

FIGURE 1.2: Thesis Framework.

blocks in Figure 1.2 shows the main contributions of the thesis. The algorithm in Chapter 3 introduced a relaxed condition for matrix transformation. In Chapter 4, the columns of the parity-check matrix are permuted based on the refined reliability information at each iterative step. Thus, a generalized parity-check transformation algorithm is presented in this chapter. Nevertheless, the existing transformation methods produce suboptimal parity-check matrices (Chapter 5). However, the possibility of improving the matrices of the BP algorithm still holds in general as presented in Chapter 5.

1.5 Research Relevance and Applications

This study shows the importance of parity-check matrix transformation for the iterative belief propagation decoding of cyclic codes. The generally used adaptive belief propagation algorithm in the literature transforms the matrix at each iteration based on the reliability matrix, using the Gaussian elimination approach. The matrix transformation enhances the performance of the BP algorithm compared to algebraic hard decision decoding and standard BP decoding. In reality, the complexity is high due to the number of operations required for Gaussian elimination, and usually not suitable for execution in real-time applications, as required in multimedia transmissions. On the other hand, considering the simplicity and performance efficiency of the PTA for symbol level RS codes, this thesis extends the Reed-Solomon PTA to the class of binary cyclic codes. As such, the performances of the proposed algorithms in this study overcome the PTA limitation and reduce the complexity of the ABP at the message passing stage.

Besides, performances of the transformed parity-check matrices used by the BP algorithm for cyclic codes are verified in this study based on the perfect knowledge model. The model selects the best parity-check matrix at each iteration, which offers room for improvement in the matrices of the BP algorithm in general. Also, the perfect knowledge model can be used in place of the ML decoding to verify the performances of newly developed iterative SD decoders for binary cyclic codes based on parity-check equations.

Despite the long existence of cyclic codes such as the BCH and RS codes, they are still adopted in most telecommunication standards. For instance, the European standard for satellite digital video broadcasting (DVB-S2) includes an error correction scheme attributed to the concatenation of an outer BCH code with inner LDPC code [19]. These traditional codes are also adopted for deploying broadcast services over various networks, such as packet-switched mobile networks, where the American CDMA2000 standard included RS codes in the implementation of high-rate broadcast data services [20]. Therefore, the work in this thesis is significant to enhance these wireless applications.

1.6 Thesis Organization

This thesis is structured as follows. Chapter 1 provides the general introduction, research motivation and research question, hypotheses, research aim and objectives. Some other details in Chapter 1 include the research argument, and the research relevance and applications.

Chapter 2 presents a detailed background of basic concepts and relevant literature. Basic definitions and descriptions of linear block codes are presented in terms of the generator and parity-check matrices. In Section 2.2.3, the relationship between the minimum distance of code and the parity-check matrix is discussed. Also, an important family of the linear codes is included in Section 2.3. Furthermore, detailed literature on coded modulation and soft-decision decoding is presented in this chapter. Several soft-decision decoders, such as soft-decision optimum decoders, dual implementation of the bitwise-MAP decoder, and enhanced hard decision decoders, algebraic list decoders, belief propagation decoder are reviewed. More so, soft-decision decoding algorithms in the literature that modifies the parity-check matrices of cyclic codes for BP decoding are examined in Chapter 2.

In Chapter 3, an extended parity-check matrix transformation algorithm (EPTA) that avoids the matrix inversion of the PTA is developed for iterative soft-decision decoding. The time complexity analysis of the algorithm is presented using big-$\mathcal{O}$ notation to allow a fair comparison of the message passing stage between the algorithm and the ABP algorithm. The simulation result presents performance analysis for different updating factors. Also, other result presents a performance comparison analysis of the proposed EPTA with other iterative soft-decision decoding algorithms.

Chapter 4 develops a novel parity-check matrix transformation algorithm for iterative soft-decision decoding of binary cyclic codes. The generalized parity-check transformation algorithm permutes the columns of the parity-check matrix based on the reliability information from the channel's output. The matrix transformation and message passing stages of the algorithm are discussed in Sections 4.3.1 and 4.3.2.

Performance analysis shows that the developed GPT exhibits better performance compared to the algebraic hard decision decoding algorithm, the conventional BP algorithm, and other soft-decision decoders. The worst-case time complexity analysis of the GPT, PTA, and ABP is performed using the big-$\mathcal{O}$ notation, showing that the ABP has the highest complexity. This implies that the GPT is an efficient SD decoder for the class of cyclic codes, and it can be implemented in real-time applications.

In Chapter 5, a baseline parity-check matrix for iterative soft-decision decoding of binary cyclic codes is developed based on the PKM. The model computes all the possible parity-check matrices according to the channel condition and selects the best (baseline) matrix based on minimum distance criteria as detailed in Section 5.2.1. Also, a numerical example is given in Section 5.2.2 showing that the matrices obtained from the proposed PKM are optimal. Results show that selecting an optimal parity-check matrix enhances decoding performance of the ABP and proposed GPT algorithms compared to the maximum likelihood decoder.

Finally, Chapter 6 discusses the general contribution of this thesis. First, a description of the study's aim and achievement is summarized. Subsequently, the results of each chapter are summarized to give an overview of the precise contribution of each chapter. Also, the chapter provides recommendations and ideas for future research possibilities.

Chapter 2

Background and Literature

2.1 Introduction

This chapter introduces the fundamental concepts of linear block codes and soft-decision decoding. The codes are defined and described in terms of the generator and parity-check matrices. Also, the relationship that exists between minimum distance and parity-check matrix of codes is shown in this chapter. Moreover, cyclic codes are discussed in Section 2.3 as an important subclass of linear block codes. Furthermore, we review the soft-decision decoding algorithms which utilize information from channel measurement to improve on the performances of known hard-decision decoders for cyclic codes in Section 2.4.2. The essence of this chapter is to provide a background for generating new results and methods in this thesis.

2.2 Linear Block Codes

A linear (n, k) block code $\mathcal{C}$ over the field $\mathbb{F}$ is a k-dimensional vector subspace of the vector space of n-tuples over $\mathbb{F}$. Suppose that $\mathbb{F}$ is a finite field $GF(q)$ of q symbols. A k-dimensional subspace of $\mathbb{F}_q^n$ contains q^k vectors of length n, where n represents the block length of the code. Hence, the code rate is given as $R = k/n$.

2.2.1 Basic Definitions

Definition 2.1. The **Hamming weight**, $w_H(\underline{c})$ of a codeword $\underline{c} \in \mathcal{C}$ is the number of places where the codeword is nonzero.

Definition 2.2. The **Hamming distance**, $d_H(\underline{c}, \underline{s})$ between two codewords $\underline{c}$ and $\underline{s}$ of the same length n is the number of positions where both codewords differ. A very useful relationship for the Hamming distance of linear codes is given by the additive operation between the two codewords as:

$$d_H(\underline{u}, \underline{v}) = w_H(\underline{u} + \underline{v}) \tag{2.1}$$

Proof. See proof in [1, 5, 21]. □

Definition 2.3. The **minimum distance** d_{min} of an (n, k) code $\mathcal{C}$ is the smallest Hamming distance between any two different codewords $\underline{u}$ and $\underline{v}$ in the code:

$$d_{min} = \min_{\underline{u},\underline{v} \in \mathcal{C}, \underline{u} \neq \underline{v}} d_H(\underline{u}, \underline{v}). \tag{2.2}$$

By [1, Theorem 3.1 on p. 88], the minimum distance of a linear code $\mathcal{C}$ is equal to the minimum Hamming weight of its nonzero codewords.

Definition 2.4. A **linear encoder** receives the message $\underline{m}$ of k symbols and outputs it as a codeword $\underline{c}$ of n symbols. The linear code $\mathcal{C}$ is a vector space with k-dimensional, thus there are k vectors $\underline{g}_i \in \mathbb{F}_q^n, i = 1, 2, \ldots, k$, which are linearly independent. Therefore, the linear encoder represents every codeword in the code as a linear combination of $m_i \in \mathbb{F}_q$ and $\underline{g}_i$ as:

$$\underline{c} = \left[m_1\underline{g}_1 + m_2\underline{g}_2 + \cdots + m_k\underline{g}_k \right]. \tag{2.3}$$

Also, a scalar multiplication of m_i and $\underline{g}_i$ in the vector space of n-tuples over $\mathbb{F}$ gives:

$$m_i\underline{g}_i = \begin{bmatrix} m_i g_{i_1} & m_i g_{i_2} & \cdots & m_i g_{i_k} \end{bmatrix}. \tag{2.4}$$

Expressing vectors $\underline{g}_i$ in the form of row vectors yield a $(k \times n)$-dimensional matrix $\mathbf{G}$, so that Equation (2.4) can represented as:

$$\underline{c} = \underbrace{\begin{bmatrix} m_1 & m_2 & \cdots & m_k \end{bmatrix}}_{\underline{m}} \underbrace{\begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \\ \vdots \\ \underline{g}_k \end{bmatrix}}_{\mathbf{G}} . \tag{2.5}$$

The rows of matrix $\mathbf{G}$ generate the linear code $\mathcal{C}(n, k)$, thus $\mathbf{G}$ is referred to as the generator matrix for $\mathcal{C}$. The matrix generates q^k distinct codewords $\underline{c}$ whenever all the q^k possible symbol vectors of the linear codes are linearly independent [5, 21]. This result depends on the rank of $\mathbf{G}$, obtained by reducing the matrix to a row reduced echelon form *rref*, that is, performing elementary row operations on $\mathbf{G}$. From [21], the rank of $\mathbf{G}$ is at most k. Hence, if $rank(\mathbf{G}) = k$, there are q^k distinct codewords. Otherwise, if $rank(\mathbf{G}) < k$ there are $q^{rank(\mathbf{G})}$ distinct codewords.

Definition 2.5. Systematic encoder: An encoder is said to be systematic if k-symbols message can be obtained directly and unaltered in the codeword. A systematic generator of the encoding operator in Theorem 2.4 is said to be systematic if the matrix $\mathbf{G}$ can be partitioned into $k \times k$ identity matrix $\mathbf{I}_k$ and $k \times (n - k)$ parity matrix $\mathbf{P}$,

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_k & \mathbf{P} \end{bmatrix} . \tag{2.6}$$

It can be inferred from the systematic $\mathbf{G}$ in Equation (2.6) that $rank(\mathbf{G}) = k$, since it is in the row reduced echelon form. Thus the encoded operation becomes:

$$\underline{c} = \underline{m}\mathbf{G} = \begin{bmatrix} \underline{m} \; \underline{m}\mathbf{P} \end{bmatrix} . \tag{2.7}$$

Note that elementary row operations on rows of $\mathbf{G}$ produce the same code. However, interchanging any two columns of the generator matrix changes the related locations of the code, while the structure of the distance is maintained [5].

2.2.2 Dual codes and Parity-check matrix

A linear code $\mathcal{C}$ is a k-dimensional vector subspace of $\mathbb{F}_n^q$, which has an $n-k$ dimensional dual space to $\mathcal{C}$ [5, Theorem 2.8 on p. 127]. Therefore, by [5, Definition 3.6 on p. 134] the dual space to an (n,k) code of dimension k is the $(n, n-k)$ dual of $\mathcal{C}$, represented by $\mathcal{C}^{\perp}$. The dual code has basis $\underline{h}_i, i = 1, 2, \ldots, n-k$ that forms the rows of a matrix $\mathbf{H}$,

$$\mathbf{H} = \begin{bmatrix} \underline{h}_1 \\ \underline{h}_2 \\ \vdots \\ \underline{h}_{n-k} \end{bmatrix}. \tag{2.8}$$

$\mathbf{H}$ is known as the **parity check matrix**, which provides information about the minimum distance of the code $\mathcal{C}$. The generator matrix and parity-check matrix of a code satisfies

$$\mathbf{G} \cdot \mathbf{H}^T = 0. \tag{2.9}$$

Thus, given a vector $\underline{v} \in \mathbb{F}_q^n$, [5, Theorem 3.2 on p. 137] showed that $\underline{v}$ is a codeword if and only if

$$\underline{v} \cdot \mathbf{H}^T = 0. \tag{2.10}$$

Equation (2.10) sets linear constraints between the bits/symbols of the codewords defined as the parity-check equations (PCEs).

2.2.3 Relationship between Minimum distance and Parity-check matrix

Given an (n,k,d) code $\mathcal{C}$ with parity-check matrix $\mathbf{H}$, the following properties must be satisfied to design codes with a certain guaranteed minimum distance [5, Theorem 3.3 on p. 136].

1. Sum of any $d-1$ or less columns is non-zero. This gives a bound on the minimum distance d_{min}, such that $d_{min} \geq d$.
2. There exist some d columns which add to zero.

Moreover, some special cases exists as a relationship between the parity-check matrix and minimum distance $\mathcal{C}$ such as [21]:

- If the column of $\mathbf{H}$ are all zeros, then $d = 1$.
- If there exist two identical columns in $\mathbf{H}$, then $d \leq 2$.
- If all the columns of $\mathbf{H}$ are distinct and non-zero, then $d \geq 3$.

These properties result in some basic bounds, such as the Singleton and Hamming bounds on block codes. The Singleton bound in [5, Theorem 3.4 on p. 136] ensures that the minimum distance of an (n, k) linear code is bounded by

$$d_{min} \leq n - k + 1. \tag{2.11}$$

Any linear code that completely satisfies the Singleton bound in Equation (2.11) is called a **maximum distance separable code**. Such codes have the greatest error-correcting capability as this ability depends on the minimum distance of the code.

Useful characterization of MDS codes, based on parity-check matrices are presented and proved in [5]. A code $\mathcal{C}$ has been shown in [5, Lemma 6.6 on p. 293] to be MDS if and only if every set of $n-k$ columns of its parity-check matrix is linearly independent. Furthermore, [5, Lemma 6.7 on p. 293] showed that the $(n, n-k)$ dual code $\mathcal{C}^{\perp}$ is MDS. In [5, Lemma 6.8 on p. 293], every k columns of a generator matrix for $\mathcal{C}$ are said to be linearly independent. This implies that every square submatrix of the parity matrix in a systematic $\mathbf{G}$ is nonsingular. Finally, it is shown in [5, Lemma 6.9 on p. 294] that the number of codewords in a q-ary (n, k) MDS code of weight $d_{min} = n - k + 1$ is equal to $(q-1)\binom{n}{n-k+1}$. These MDS properties will be referred to throughout this thesis.

2.3 Cyclic Codes

Cyclic codes constitute a subclass of linear block codes that are built on polynomial operations with algebraic structure. This code's error correction capabilities of are enhanced in terms of burst error corrections and detection due to the algebraic structures. Also, the codes are hardware compatible since encoding and decoding can be easily implemented through shift registers with linear sequential circuits.

By definition [1], it is said that an (n, k) linear code $\mathcal{C}$ is cyclic if each cyclic shift of a codeword in $\mathcal{C}$ gives a codeword in the code. This codeword can be expressed uniquely using a polynomial, where the components of the codeword $\underline{c} = (c_0, c_1, \cdots, c_{n-1})$ are regarded as the polynomial coefficients, given as:

$$\underline{c}(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}. \tag{2.12}$$

Algebraically, cyclic codes of length n in a vector space $\mathbb{F}_q^n$, conform to the analysis of ideals in the residue class ring $\mathcal{R}_n = \mathbb{F}_q[x]/(x^n - 1)$. Moreover, studying the ideals in $\mathcal{R}_n$ focuses on obtaining irreducible factors of $x^n - 1$ over $\mathbb{F}_q$. Given an (n, k) cyclic code $\mathcal{C} \in \mathcal{R}_n$, [21, Theorem 1 on p. 200] proves that a unique monic polynomial $g(x)$ of the smallest degree exists in $\mathcal{C}$. Also, the polynomial $g(x)$ generates the ideal, which is known as the generator polynomial of $\mathcal{C}$. Suppose $g(x)$ has degree $n - k$, that is,

$$g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k}$$

Therefore, multiplying $g(x)$ with a message polynomial $m(x)$ of degree less than k, say $k-1$, yields a resultant polynomial $c(x)$ of degree $n-1$. Moreover, [21, Theorem 1 on p. 200] shows the generator polynomial $g(x)$ to be a factor of $x^n - 1$ in $\mathbb{F}_q[x]$.

The code polynomial associated to $m(x) = m_0 + \cdots + m_{k-1}x^{k-1}$ is derived by multiplying $m(x)$ by $g(x)$ as:

$$\begin{aligned} c(x) &= m(x)g(x) \\ &= m_0 g(x) + m_1 x g(x) + \cdots + m_{k-1}x^{k-1}g(x). \end{aligned} \tag{2.13}$$

The operation in Equation (2.13) can be represented as

$$c(x) = \underbrace{\begin{bmatrix} m_0 & m_1 & \ldots & m_{k-1} \end{bmatrix}}_{\underline{m}} \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}, \tag{2.14}$$

where $\underline{m}$ is the corresponding message vector to $m(x)$. More so, the generator polynomial $g(x)$ of the cyclic codes $\mathcal{C}$ can be represented in terms of a $(k \times n)$ generator matrix $\mathbf{G}$, since the codes constitute a subset of the linear codes. Thus, Equation (2.14) is denoted as [5]:

$$\underline{c}_m = \begin{bmatrix} m_0, m_1, \ldots, m_{k-1} \end{bmatrix} \begin{bmatrix} g_0 & g_1 & \cdots & g_r & 0 & 0 & 0 & 0 & 0 \\ 0 & g_0 & g_1 & \cdots & g_r & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & \cdots & g_r & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & \cdots & g_r & 0 \\ 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & \cdots & g_r \end{bmatrix}. \tag{2.15}$$

Furthermore, an (n, k) cyclic code with $g(x)$ has a corresponding **parity-check polynomial** polynomial $h(x)$ with degree k that satisfies $h(x)g(x) = x^n - 1$ [5]. Also, since $\mathcal{C}$ contain all the codewords that are multiples of $g(x)$, thus for each codeword,

$$\begin{aligned} c(x)h(x) &= m(x)g(x)h(x) \\ &= m(x)(x^n - 1) = 0. \end{aligned} \tag{2.16}$$

Therefore, a given polynomial $r(x)$ is a valid codeword once it satisfies the condition that $r(x)h(x)(\mod x^n - 1) = 0$ [5]. Suppose a valid codeword $\underset{\sim}{c}$ is sent through an error-prone channel as described in Figure 1.1, and a received word $r(x)$ is obtained at the receiver. A corresponding **syndrome polynomial** to $r(x)$ is described as [5]:

$$s(x) = r(x)h(x)(\mod x^n - 1), \tag{2.17}$$

such that $s(x)$ is yields zero whenever $r(x)$ is obtained as a valid codeword. Similar to constructing $\mathbf{G}$ for the cyclic codes, a parity-check matrix $\mathbf{H}$ corresponding to $h(x)$ has been computed in [5, 21].

2.4 Decoding of Cyclic Codes

The cyclic code's algebraic properties are useful for code constructions. Various classes of cyclic codes have been developed over time, including the BCH codes, *RS* codes, projective geometry residue codes, and Fire codes. The BCH codes are a subset of cyclic codes. The codes are famous for their ability to correct multiple errors, as well as the convenience to encode and decode.

2.4.1 Algebraic Hard-Decision Decoding

A well known decoder for the cyclic code is the Meggitt decoder [22], which employs linear feedback shift registers to form parity-check digits and to correct errors. Besides, Peterson in [23] showed that a binary BCH code has cyclic structures and developed a decoding algorithm for the codes. Peterson's decoder has been generalized and improved in [17, 24–28]. But, the Berlekamp and Chien's search algorithms are the most efficient of these algebraic HD algorithms. Also, Sugiyama, Kashara, Hirasawa, and Namekawa [29] showed the possibility of implementing the Euclid's algorithm to efficiently decode both BCH and *RS* codes. These developed decoding algorithms assumed that the output of the channel is also binary. However, this assumption do not necessarily hold for many communication applications. More so, there is a significant performance degradation when a binary channel output is assumed [5].

2.4.2 Soft-Decision Decoding

With regards to enhancing the performance of traditional binary decoders, researchers in the late 1970s started discussing the concept of coded modulations and soft-decision information decoding, alongside the algebraic properties of the code. In this study, we refer to modulation as the process by which bits are converted to signals suitable for transmission. Assume that a binary phase shift keying (BPSK) modulator is used to map an n bits codeword $\underline{c} \in \mathcal{C}$, $c_i \in [0,1]$ to signal constellation points. The bits are randomly generated with probabilities $\Pr_1 = \Pr(c_i = 1)$ and $\Pr_0 = \Pr(c_i = 0)$. Here, it is presumed that 0 and 1 are equally likely, and that $\Pr_1 \neq \Pr_0$. Thus, 0 is mapped to $+1$, while 1 is mapped to -1. Let $\tilde{c}_i$ represent the ± 1-valued bit associated with the $[0,1]$-valued bit c_i. The mappings

$$\tilde{c}_i = -(2c_i - 1) \text{ or } \tilde{c}_i = (2c_i - 1)$$

can be either be used during practical implementation [5]. Subsequently, the modulated vector is transmitted through the AWGN channel, to obtain a vector

$$\underline{r} = \underline{\tilde{c}} + \underline{z}, \tag{2.18}$$

where $\underline{z} \sim \mathcal{N}(0, \sigma^2)$ is an independent and identically distributed (iid) variable. Thus, a soft-decision decoder directly operates on $\underline{r}$ to yield an estimated codeword $\underline{\hat{c}}$.

2.4.2.1 Soft-decision Optimum Decoders

A well known SD optimum decoder is the maximum likelihood decoder [5], which chooses $\underline{c} \in \mathcal{C}$ with the least Euclidean distance to the received vector $\underline{r}$. Given

$\underline{c} = \underline{u} \in \mathcal{C}$, the ML decoder operates by maximizing the probability of $\underline{r}$ over all $\underline{u}$ as

$$\hat{\underline{c}} = \max_{\underline{u}\in\mathcal{C}} \ \Pr(\underline{r}|\underline{c} = \underline{u}) = \arg\max_{\underline{u}\in\mathcal{C}} f(\underline{r}|\underline{c} = \underline{u}), \tag{2.19}$$

where f represents the probability density function (pdf) of $\underline{r}$. Given a symbol vector $\underline{s}$, the pdf in Equation (2.19) can be simplified as

$$\begin{aligned} f(\underline{r}|\underline{c} = \underline{u}) &= f(\underline{r}|\underline{s} = 1 - 2\underline{u}) \\ &= f([r_1 \ r_2 \ \cdots \ r_n]|[s_1 \ s_2 \ \cdots \ s_n]). \end{aligned}$$

Since $s_i, i = 1, 2, \cdots, n$ is given, the received values are conditionally independent. Thus,

$$\begin{aligned} f(\underline{r}|\underline{c} = \underline{u}) &= \prod_{i=1}^{n} f(r_i|s_i) \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r_i - s_i)^2}{2\sigma^2}} \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(r_i - s_i)^2}, \end{aligned}$$

where $\sum_{i=1}^{n}(r_i - s_i)^2 = \|\underline{r} - \underline{s}\|^2$ is the Euclidean distance between $\underline{r}$ and $\underline{s}$. To minimize the exponential function, it suffices to maximize the left hand side (LHS) of Equation (2.20) as

$$\max_{\underline{u}\in\mathcal{C}} f(\underline{r}|\underline{c} = \underline{u}) \Leftrightarrow \min_{\substack{\underline{u}\in\mathcal{C}, \\ \underline{s}=1-2\underline{u}}} \ \|\underline{r} - \underline{s}\|^2, \ \underline{r}, \underline{s} \in \mathbb{R}^n. \tag{2.20}$$

Assume all the codewords are equally likely a priori, Equation (2.19) becomes optimal by minimizing the probability that $\hat{\underline{c}}$ is not equal to $\underline{c}$ in terms of the block or frame error rate. Generally, there are q^k codewords and correlations to be computed, which makes the ML decoder computational inefficient.

The bitwise optimal soft decoders [5] are used as an approximation to ML decoding for practical implementation purposes. There is a possibility of iterating on this decoder, which makes it incredibly useful in practice. Thus, a stopping condition can be imposed during the iterations. The decoder aims at minimizing the probability of bit errors in a codeword rather than the probability of block errors. that is, for every $i = 1, 2, \cdots, n$, it minimizes $\Pr(c_i \neq b_i)$, where $\underline{b} = [b_1\ b_2\ \cdots\ b_n]$ is a received vector from the decoder. Note that there is no constraint that $\underline{b}$ must be a codeword. Here, the decision rule in Equation (2.19) simplifies to the MAP maximum *a posteriori*. The bitwise MAP condition is given as a test between $b_i = 0$ and $b_i = 1$, so that

$$b_i = \begin{cases} 0, & \text{if } \Pr(c_i = 0|\underline{r}) > \Pr(c_i = 1|\underline{r}). \\ 1, & \text{otherwise.} \end{cases} \tag{2.21}$$

Since

$$b_i = 0 : \ \Pr(c_i = 0|\underline{r}) > 1 - \ \Pr(c_i = 0|\underline{r}), \tag{2.22}$$

dividing both sides by the RHS, the test in Equation (2.21) can be represented as a ratio,

$$b_i = 0 : \frac{\Pr(c_i = 0|\underline{r})}{\Pr(c_i = 1|\underline{r})} > 1. \tag{2.23}$$

Using Bayes rule, the LHS of Equation (2.23) becomes

$$\frac{f(\underline{r}|c_i = 0)\Pr(c_i = 0)}{f(\underline{r}|c_i = 1)\Pr(c_i = 1)}, \tag{2.24}$$

where $f(\underline{r}|c_i = 0 \text{ or } 1)$ are corresponding likelihood functions as in Equation (2.20), and $\Pr(c_i = 0 \text{ or } 1)$ are *a priori* probabilities. Assume the message is equally likely (equal prior), it turns out for binary linear codes that $\Pr(c_i = 0)$ and $\Pr(c_i = 1)$ are equal. Hence, considering the entire $\underline{r}$, the *vector likelihood ratio* for the i-th bit is derived as:

$$b_i = 0 : \frac{f(\underline{r}|c_i = 0)}{f(\underline{r}|c_i = 1)} > 1. \tag{2.25}$$

For most channel conditions, it is suitable to express this ratio as the vector log likelihood ratio (LLR)

$$L_i = \log \frac{f(\underline{r}|c_i = 0)}{f(\underline{r}|c_i = 1)}. \tag{2.26}$$

Example 2.1. *Consider a* $(6, 3)$ *code containing all possible* 2^k *codewords*

$$\mathcal{C} = \begin{Bmatrix} 000000 \\ 110100 \\ 101010 \\ 011001 \\ 011110 \\ 110011 \\ 101101 \\ 000111 \end{Bmatrix}$$

Let the obtained vector from the channel be $\underline{r} = [r_1, r_2, r_3, r_4, r_5, r_6]$. *Suppose the last three bits of* $[\underline{c}_1\ \underline{c}_2\ \cdots\ \underline{c}_{2^k}] \in \mathcal{C}$ *are systematic messages. The vector LLR of the i-th bit, say* $i = 4$ *is:*

$$L_4 = \log \frac{f(\underline{r}|000000) + f(\underline{r}|101010) + f(\underline{r}|011001) + f(\underline{r}|110011)}{f(\underline{r}|110100) + f(\underline{r}|011110) + f(\underline{r}|101101) + f(\underline{r}|000111)}. \tag{2.27}$$

From Example 2.1, the *scalar likelihood ratios* of $r_i \in \underline{r}$ given $c_i \in \underline{c}_\mu$, $\mu = 1, 2, \cdots, 2^k$ can be obtained by dividing all the conditional pdfs in Equation (2.27) by an expression $f(\underline{r}|111111)$. For instance, say the conditional pdf $f(\underline{r}|\underline{c} = 101010)$ is selected, the scalar likelihood ratio of r_i given c_i is obtained as:

$$\frac{f(\underline{r}|101010)}{f(\underline{r}|111111)} = \frac{\prod\limits_{i:c_i=0} f(r_i|0) \prod\limits_{i:c_i=1} f(r_i|1)}{\prod\limits_{i:c_i=0} f(r_i|1) \prod\limits_{i:c_i=1} f(r_i|1)} = \frac{\prod\limits_{i:c_i=0} f(r_i|c_i = 0)}{\prod\limits_{i:c_i=0} f(r_i|c_i = 1)}.$$

Therefore, the scalar LLR is given as:

$$l_i = \log \frac{f(r_i|c_i = 0)}{f(r_i|c_i = 1)}. \tag{2.28}$$

For BPSK over AWGN, Equation (2.28) becomes

$$l_i = \log \frac{e^{-(r_i-1)^2}}{e^{-(r_i+1)^2}} = \log e^{\frac{2r_i}{\sigma^2}} = \frac{2r_i}{\sigma^2}. \tag{2.29}$$

Furthermore, since the bitwise MAP utilizes the vector LLR to directly make decisions on the i-th bit of a received vector. The vector LLR in Equation (2.27) can be expressed in terms of the scalar LLR as

$$L_4 = l_4 + \log \frac{e^{l_1+l_2+l_3+l_5+l_6} + e^{l_2+l_6} + e^{l_1+l_5} + e^{l_3}}{e^{l_3+l+5+l_6} + e^{l_1+l_6} + e^{l_1+l_2+l_3}}. \tag{2.30}$$

Hence, a general expression of Equation (2.30) for the code $\mathcal{C}$ is given as follows:

1. Define

$$\mathcal{C}|_{c_i=0} = \{\underline{c} \in \mathcal{C} : c_i = 0\}, \tag{2.31}$$

where $\mathcal{C}$ conditioned on $c_i = 0$ is a subcode (subspace) and can only have two dimensions, k and $k-1$. Assume that the code is such that no one coordinate remains zero all the time, then the dimension of the subcode will be $k-1$. This property is useful in the bitwise MAP decoder expression because the numerator is controlled by codewords of the form Equation (2.31) for L.

2. From the assumption in Item 1, the denominator is obtained as follows.

$$\mathcal{C}|_{c_i=1} = \{\underline{c} \in \mathcal{C} : c_i = 1\}, \tag{2.32}$$

where $\mathcal{C}$ conditioned on $c_i = 1$ has dimension 2^{k-1} and is not a subcode (subspace) since the all zero vector is not contained in $\mathcal{C}$.

3. From Items 1 and 2, Equation (2.30) is generalized as:

$$L_i = \log \frac{\sum\limits_{\underline{u}\in\mathcal{C},u_i=0} \prod\limits_{j:u_j=0} e^{l_j}}{\sum\limits_{\underline{u}\in\mathcal{C},u_i=1} \prod\limits_{j:u_j=1} e^{l_j}}, \tag{2.33}$$

where $l_j = \frac{2r_j}{\sigma^2}$. Equation (2.33) is the same as

$$L_i = \underbrace{l_i}_{intrinsic\ LLR} + \underbrace{\log \frac{\sum\limits_{\underline{u}\in\mathcal{C},u_i=0} \prod\limits_{j\neq i:u_j=0} e^{l_j}}{\sum\limits_{\underline{u}\in\mathcal{C},u_i=1} \prod\limits_{j:u_j=0} e^{l_j}}}_{extrinsic\ LLR}, \tag{2.34}$$

where the intrinsic LLR is the channel information, which is the belief for the LLR that the channel is providing about the i-th bit. On the other hand, the extrinsic LLR is based on the code information, which is extrinsic to the i-th received value. The final decision on the i-th bit is given as

$$b_i = \begin{cases} 0, & \text{if } L_i > 0 \\ 1, & \text{if } L_i \leq 0. \end{cases} \tag{2.35}$$

The bitwise-MAP is 2^k complex, where k represents the code's dimension. This implies that as k increases it become impracticable to completely implement the optimal decoder. However, Item 3 suggests that it might be possible to iteratively or approximately evaluate the extrinsic term, which brings about the idea of sub-optimal soft-decision decoders. In all approximate SD decoders, the main focus is to approximate the extrinsic LLR so that the errors in the extrinsic LLR become bounded.

2.4.2.2 Dual implementation of the bitwise-MAP decoder

The idea of implementing the dual codes for the bitwise-MAP decoder originates from Hartmann and Rudolph [30]. This idea is important in the simplification of the suboptimal decoder. A dual code has fewer codewords compared to the original code at high rates. Therefore, the bitwise-MAP decoder that is described in terms of the codeword of the dual has a computational advantage. For equally probable codewords, a decoding rule to minimize the probability of symbol error over a time-discrete memoryless channel was presented for linear codes in [30]. This rule is also exhaustive since every word in the dual code is used in the decoding process. Thus, the algorithm is efficient only for codes that have a limited number of codewords in its dual code, that is, high or medium rate codes with short lengths.

However, simplified methods to enumerate all codewords of dual codes were proposed in [31, 32]. A simple MAP decoding technique was considered in [31] for the Reed-Muller and Hamming codes of first-order. This technique utilizes the dual code structure to reduce the listing of all codewords. For Hamming codes, the computation of codewords was improved based on the Fast Hadamard Transform (FHT), since listing the codewords of the dual of this code consists of a Hadamard structure. As a result, the MAP decoding was efficiently implemented. Similarly, an efficient MAP decoding implementation for high rate binary cyclic codes was provided in [32]. This approach was based on the dual codeword listing. The enumeration of codewords has a circulant structure for cyclic codes. Therefore, the study in [32] aimed at improving the implementation of multiplication through the use of a block circulant matrix, based on the Fast Fourier Transform (FFT). However, the MAP decoder still remains practically infeasible for low rate long length codes due to the level of computational complexity involved.

2.4.2.3 Enhanced Hard-decision decoding

Additional studies to minimize the complexity of ML decoding have been implemented by the enhanced hard-decision decoding (eHDD) algorithms, like the Generalized Minimum Distance (GMD) decoding [33], Chase decoding [34], and the combination of Chase and GMD algorithms (CGA) [35]. Such algorithms are approximations of ML decoding with low complexity by which reliable information is used to enhance hard decision decoding.

The GMD algorithm uses reliability measures to produce a number of codewords that are compared to the received vector. The algorithm performs a test for each candidate codeword according to a reasonable condition for optimality. Thus, the most probable candidate codeword is selected as the codeword that has been decoded. Similarly, Chase decoding systematically searches through a determined amount of error patterns which are associated to specific unreliable positions. The set of tested positions determines the algorithm's maximum number of codewords considered and decoding efficiency. Consequently, a modification to Chase's algorithm was presented in [36]. This algorithm is a simplified Chase decoding that allows searching only for the positions that correspond to reliabilities, which are lower than a set limit. Suppose that a set of positions is given, the decoding performance is determined by the decided threshold, while the maximum number of calculations relies on both the value of optimum threshold and the signal-to-noise ratio (SNR). Nevertheless, both the Chase and simplified adaptive Chase algorithms display low decoding performances for low SNR or large code lengths.

Moreover, an efficient ML decoding algorithm was suggested in [37]. The decoder produces a larger set of likely codewords when a noisy vector is received. But a smaller set of probable codewords is generated when the vector being received is not in error. That is, the algorithm's decoding complexity is defined by the received vector. Although this method enhances the computational complexities of [34] and [36] for short length codes due to the introduced stopping conditions, the algorithm's complexity exponentially increases with the code length.

A different approach was given in [38] for binary linear codes to preserve the optimization of decoding while avoiding excessive searches. The proposed algorithm relies on ordered statistics, allowing symbols of the received vector to be reordered based on their measure of reliability. The essence of the ordered statistics decoding (OSD) approach is to gradually attain, in a number of stages, the desired error efficiency. A tight bound was established for the error performance, and the decoding terminates either at near-optimum error output or when the required error performance level is reached. This provides flexibility between the complexity of decoding and error performance.

The OSD consists of hard decision decoding and reprocessing steps. For an (n, k) binary linear code $\mathcal{C}$. Unlike the GMD and Chase algorithms that utilize the least reliable symbols, OSD determines the HDD from the obtained symbols' ordered reliability values, resulting in a codeword with little or no errors in the first k most reliable independent symbol positions. Moreover, the reprocessing step is intended to iteratively improve decoded codewords of the HDD till the stopping criterion is reached. Since the order reprocessing determines the decoding complexity, a cost function and resource test are considered to reduce the number of computations at every stage of the reprocessing process. For short length codes, the OSD exhibits a low worst-case computational cost as compared to other optimum or suboptimal decoding algorithms, such as [36], [37]. However, high-order reprocessing is needed to obtain best practical error performance for long length codes.

2.4.2.4 Algebraic Soft-decision decoding

Algebraic soft-decision decoding is another class of SD decoding algorithms for cyclic codes. In 1997, Sudan [39] created a list decoding algorithm to improve the algebraic decoding performance for *RS* codes. The Sudan algorithm is based on producing a list of all possible codewords in the Hamming distance of any received vector. The decoding radius is also expanded to a certain bound as a manner of decoding beyond the code's error-correcting capacity. This algorithm considers decoding low rate *RS* codes as a bivariate polynomial interpolation and factorization problem, which can be resolved within the polynomial-time but with high quadratic computational complexity. Hence, Guruswani and Sudan (GS) [40] have implemented an enhanced list decoding algorithm for decoding *RS* codes. This algorithm reduces the list decoding problem to a curve-fitting problem over a field. The GS algorithm improves over the original Sudan algorithm for all rates, and has been extended to weighted curve fitting, motivated by the soft-decision decoding problem.

Wu [41] has further developed an alternative list decoding technique for the *RS* codes and BCH codes based on a rational curve-fitting algorithm. The polynomial algorithm used in Wu's work is based on rational interpolation and factorization, which has the same list error correction capability as the GS algorithm but reduced computational complexity owing to low multiplicity. A further strategy to correcting errors beyond half the minimum distance is one that depends on calculating an extended syndrome from the word received and deriving a polynomial of the error location. Schmidt *et al.* [42], introduced a bounded distance decoding for *RS* codes based on syndrome extension technique in the frequency domain. This method provides practically the same decoding efficiency as the Sudan algorithm but exhibits a reduced complexity as compared with the Sudan algorithm.

Furthermore, KV [16] presented a polynomial-time soft-decision algorithm for RS codes to reduce the complexity of algebraic decoding. The KV allows a free choice between computational complexity and decoding performance. This is achieved by extending the GS algebraic interpolation techniques to a soft decision decoding algorithm, using distance metrics rather than the Hamming distance to form soft reliable symbols and multiplicities for distinct points. The algorithm obtains a mapping from later probabilities (soft information) to multiplicities and utilizes the GS algorithm's interpolation and factorization to decode. The KV algorithm outperforms all other algebraic decoding algorithms, but with a very large computational complexity.

Due to the high cost of computing the interpolation process, an algebraic Chase decoding based on module minimization approach [43], has lately been implemented to fix the interpolation problem with reduced computational complexity. The module minimization method formulates the interpolation test-vectors using the soft received information. This formulation allows a re-encoding transform to reduce the size of the module entries, resulting in a simpler module minimization. This algorithm has decreased computational complexity compared to the KV soft-decision list algorithm. Nonetheless, the algorithm allows each Chase decoding trial to be executed in the parallel, thus resulting in a high decoding latency for practical implementation.

2.4.2.5 Belief propagation decoding

Iterative SD decoding of long length, LDPC codes based on belief propagation was first introduced by Gallager [6, 44]. Belief propagation iterative decoding uses both the soft obtained information and channel property knowledge to derive log-likelihood ratios for the signal being transmitted. Assume that a binary signal is transmitted with a probability of $p = 1$, while the probability of sending a 0 is given as $1 - p$. These probabilities can be represented as the log-likelihood ratio [45],

$$LLR(p) = \log\left(\frac{1-p}{p}\right), \tag{2.36}$$

which only requires an addition operation, thus reducing the complexity of implementation.

BP focuses on computing a posterior probability (APP) of each bit in a codeword, $P_i = P(c_i = 1|E)$. The APP is the probability that the ith bit is a 1 given an event E, such that all parity-check equations are constrained. Let P_i^{int} represent the *intrinsic* (*a posterior probability*), which is the original bit probability without knowing the code constraints. Also, let P_i^{int} represent an *extrinsic* probability, which is the outcome of learning from the event E. For each bit of a codeword, the BP calculates an estimate of the value of the APP for each iteration. Note that the cycle free code produces the precise APP approximation [45]. Therefore, the extrinsic information acquired from the parity-check equations in the iteration becomes the intrinsic (inherent) information for the next iteration and does not depend on the initial intrinsic value for that bit at the initial iteration till the information returns through the process.

As described in [45], extrinsic probability of the ith bit in a codeword can be obtained from the jth PCEs by determining the probability of the other codeword bits being an odd number are a 1 that an odd number of the other codeword bits are a 1,

$$P_{i,j} = \frac{1}{2} + \frac{1}{2} \prod_{i' \in B_j, i' \neq i} (1 - 2P_{i'}^{int}), \tag{2.37}$$

where B_j is the set of column positions of the bits in the jth PCE of the code. From Section 2.4.2.5, we have

$$1 - 2p = \tanh\left(\frac{1}{2}\log\left(\frac{1-p}{p}\right)\right),$$

to produce the extrinsic LLR,

$$LLR(P_{i,j}^{ext}) = \log\left(\frac{1 + \prod_{i' \in B_j, i' \neq i} \tanh\left(LLR(P_{i'}^{int})/2\right)}{1 - \prod_{i' \in B_j, i' \neq i} \tanh\left(LLR(P_{i'}^{int})/2\right)}\right). \tag{2.38}$$

Therefore, LLR of the approximated APP for the ith bit is obtained as the combination of the sum of the extrinsic LLRs and the original LLR at each iteration, represented by:

$$LLR(P_i) = LLR(P_i^{int}) + \sum_{j \in A_i} LLR(P_{i,j}^{ext}), \tag{2.39}$$

where A_i represents the set of corresponding row positions of the PCEs that is satisfied on every ith bit of the code. The iteration terminates whenever a stopping condition is attained. At this point, either the decoder is converging to a valid codeword or it reaches the maximum number of defined iterations.

The relationship between the BP decoding and linear codes is linked to the graphical representation of the codes. For example, the analysis and performance of the BP decoding may be associated with the presence of cycles in the Tanner graph representation of the codes. Tanner [46] studied codes on graphs in 1981. The study expanded the parity-check limitations of Gallager's LDPC codes to random linear code challenges. Also, impact of the cycles on the error rate performance of the initial Gallager codes and the benefits of applying short cycle free graphs were described in [47]. It has been shown in [47] that the BP decoding algorithm is inappropriate for HDPC codes like the RS and BCH codes. This is due to a large number of short cycles in codes' factor graph. Thus, creating an undesirable association among the messages, which reduces the efficiency of BP decoding.

2.4.2.6 Adaptive Belief Propagation Algorithm

In an effort to implement the belief propagation algorithm as a decoding technique for high dense parity-check codes, Jiang in *et al.* [12] showed the option of using a stochastic shifting based iterative decoding algorithm for cyclic codes. The algorithm capitalized on the cyclic structure of the codes to effectively decode according to the graph using a different representation of the parity-check matrix at every decoding iteration. The study demonstrated that proper random cyclic shifting (scheduling) of the channel's reliability information from the channel avoids the BP algorithm from getting stuck at the update stages in the stochastic shifting algorithm. Some other iterative SD decoding algorithms that utilize the parity-check matrix adaptation at each iteration have been provided in [13, 14, 48]. At each iteration, the ABP algorithm reduces columns of the binary check matrix that matches the positions of the least reliable bits to unitary columns. Thus, making the matrix ideal for BP decoding. The ABP algorithm is described as follows.

Consider a binary cyclic code $\mathcal{C}(n,k)$ with rate $R = k/n$ over $\mathbb{F}_2$. Let $\underline{c} = [c_1, c_2, \ldots, c_n]$ be a codeword in $\mathcal{C}$. Each codeword can be associated with a polynomial $c(x)$ over GF_2 mod $(x^n + 1)$ such that all cyclic shifts of $c(x)$ yields a valid codeword. The generator polynomial of $\mathcal{C}$ exists as a unique monic polynomial of minimal degree $n-k$ in the set of the code polynomials in $\mathcal{C}$. Hence, every codeword can be uniquely expressed as

$$c(x) = m(x)g(x) \bmod x^n + 1, \tag{2.40}$$

where $m(x)$ is a polynomial in $GF_2[x]$ of degree less than k. Also, there exist a polynomial $h(x)$ of degree k such that $g(x)h(x) = x^n + 1$ since $g(x)$ is a factor of $x^n + 1$. Thus, cyclically shifting the binary coefficients of $h(x)$ yield a standard form of the parity-check matrix of the binary cyclic codes as:

$$\mathbf{H} = \begin{bmatrix} h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \end{bmatrix}. \tag{2.41}$$

The matrix $\mathbf{H}$ is dense and typically contains a lot of 4-cycle lengths, which reduces the decoding performance of the BP algorithm. ABP is intended to accommodate these high dense matrices, thus the decoding method transforms the matrix $\mathbf{H}$, according to the bit reliabilities based on Gauss reduction to eliminate cycles in the subgraph corresponding to the bits obtained with low reliability.

Suppose that every bit $c_i\underline{c}$, $i = 1, 2, \ldots, n$ is modulated using a BPSK modulation scheme, where 0 and 1 are mapped to constellation points +1 and −1 respectively.

The modulated vector is transmitted over an AWGN channel to produce a soft-decision received vector $\underline{r} = (r_1, r_2, \cdots, r_n)$. Consequently, the initial reliability (L^0) of $r_i \in \underline{r}$ is represented by the log-likelihood ratios,

$$L^0(c_i) = \log \frac{P(c_i = 0|r_i)}{P(c_i = 1|r_i)}, \tag{2.42}$$

as calculated from the channel output.

The ABP decoding algorithm consists of the update phase of the matrix and the bit reliability update phase. The matrix update stage involves sorting the magnitudes of the obtained LLRs, $|L(c_i)|$ in ascending order. Let $\{i_1, i_2, \cdots, i_n\}$ be the bit indices for the sorted LLRs. That is, the bits at the i_1th and i_nth positions are the least reliable and the most reliable respectively. Since the original $(n-k) \times n$ check matrix is full rank, there exists at least $n-k$ independent columns that can be reduced to weigh one. Therefore, the Gaussian elimination is implemented to reduce all the ith independent columns of $\mathbf{H}$ to identity columns as shown in *Equation* (2.43).

$$\hat{\mathbf{H}} = \begin{bmatrix} \cdot & 1 & \cdot & 0 & 0 & \cdot & \cdot & 0 & \cdot & 0 & \cdot \\ \cdot & 0 & \cdot & 1 & 0 & \cdot & \cdot & 0 & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & 0 & 1 & \cdot & \cdot & 0 & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & 0 & 0 & \cdot & \cdot & 1 & \cdot & 0 & \cdot \\ \cdot & 0 & \cdot & 0 & 0 & \cdot & \cdot & 0 & \cdot & 1 & \cdot \end{bmatrix} \tag{2.43}$$

The unitary columns in $\hat{\mathbf{H}}$ represent the unreliable positions. However, for non-MDS codes, it is not ensured that the $n-k$ identity columns will correspond to the least reliable bits [14].

In the bit reliability update stage (message passing stage), the extrinsic LLR vector L^l_{ext} for each bit is obtained in relation to the LLRs $L^l(c_i)$ for each lth iteration, using the sum-product algorithm (SPA) [49] based on the adapted parity-check matrix as:

$$L^l_{ext}(c_i) = \sum_{\substack{j=1 \\ H_{ji}=1}}^{n-k} 2\tanh^{-1}\left(\prod_{\substack{p=1 \\ p\neq i, H_{jp}=1}}^{a} \tanh\left(\frac{L(c_p)}{2}\right)\right). \tag{2.44}$$

The SPA updates the bit reliability so that at every iteration the channel LLR of every bit $L^0(c_i)$ converges to the *a posterior* LLR. This convergence is only possible if the graph was acyclic [7]. Hence, the bit reliability is updated as

$$L^{(l+1)} = L^{(l)} + \alpha L_{ext}^{(l)}, \tag{2.45}$$

where $0 < \alpha \leq 1$ is a damping factor. The iteration will proceed till the specified maximum number of iterations is reached or till the condition for parity checks has been satisfied. Note that transforming the parity-check matrix does not ensure that the graph is cycle free, rather it restricts the error-prorogation resulting in an approximated a posteriori LLR of the bits. However, the updated LLR of the bits is expected to have the correct sign so that there are no errors after a hard decision.

ABP effectively improves the performance of short length RS codes over hard-decision decoding and the traditional belief propagation algorithm. Unfortunately, the performance decreases as the length of the codeword increases. More so, an analysis of the ABP performance [50] has shown that the ABP algorithm is approximately similar to the OSD of order-1 at medium to high SNR. This means that the ABP achieves near maximum likelihood decoding only for high rate codes.

Moreover, Kamali *et al.* [51], discusses the finding of a sparse parity-check matrix based on Vardy decomposition [52], for the binary image of RS code, to apply the BP algorithm as a bit-level SD decoding technique. Despite the implementation of the Vardy decomposition to obtain a sparse binary parity-check matrix for the RS codes, there are limitations associated with this strategy. Sometimes there would not be a BCH subfield subcode. In other instances, the size of the BCH code becomes small that there are a lot of closely related vectors. As a result, the parity-check matrix will not be sparse. It was emphasized that the Vardy decomposition technique can not be used for high rate RS codes and that the total number of nonzero elements in the rows and columns of the parity-check matrix is closely associated to the BP performance.

Furthermore, the adaptive belief propagation was combined to enhance performances of the eHDD and algebraic soft decoding in the literature. In [53], an iterative decoding algorithm was developed based on ABP decoding and reliability-based decoding such as OSD and box and match algorithm (BMA) for linear block codes to improve the performance of moderate length codes. The ABP-OSD or ABP-BMA algorithms focus on reducing the number of errors in the most reliable basis as the bit reliabilities converge to the optimum codeword. This enables the errors to be corrected with a smaller OSD or BMA order compared to the order required when using only channel information. Similar to [12], the algorithm decodes on the graph described by a new parity-check matrix at each iteration, such that the initial parity-check matrix is adapted to reduce the generation of unreliable soft information at every decoding iteration. Besides, an algebraic SD list decoding algorithm for RS codes based on the ABP and KV algorithms was proposed in [54] to attain near ML performance for relatively short length, high rate codes. The ABP-KV algorithm utilizes the ABP decoder to enhance the soft-input information, which is then used by an interpolation multiplicity assignment algorithm. However, these hybrid decoding methods yield only hard decisions as output information and retain relatively high computational complexity compared to implementing only the ABP for bit-level SD decoding of linear block codes.

2.4.2.7 Redundant Parity-check Matrix Algorithm

Aside from the use of the ABP algorithm to iteratively enhance the parity-check matrix for BP decoding, it has been demonstrated that logically adding selected redundant rows can improve the minimum weight of codewords, and trapping sets of a given parity-check matrix [55], thereby improving the performance of BP decoding. The random redundant soft-decision decoding (RRD) algorithm [56] utilizes a transient redundant parity-check matrix that is obtained at each decoding stage based on the permutation group of the code $\mathcal{C}$, Per($\mathcal{C}$). The Per($\mathcal{C}$), also known as the automorphism group, is defined as the set of permutations of coordinate places that send $\mathcal{C}$ onto itself [21]. The automorphism group has been well studied for many block codes [21], while the method in [57, 58] can be used to obtain the permutation groups for short length codes. Therefore, RRD decodes with the permuted LLR vectors, which is similar to decoding over the permuted parity-check matrix. It also utilizes varying damping factor to scale the soft information vector values at different decoding iteration until a valid codeword is obtained.

Moreover, the multiple-bases belief-propagation (MBBP) algorithm based on the redundant parity-check matrix was implemented in [59, 60]. Unlike the RRD that utilizes the transient redundant parity-check matrix by modifying the initial parity-check matrix at each decoding stage, the MBBP algorithm describes $n \times n$ parity-check matrices derived from the minimum weight codewords of the dual code. The separate structures of the redundant matrices affect the decoding computational complexity of both algorithms. The MBBP uses a certain number l of BP decoders in parallel, where the input of the decoders are different parity-check matrix $\mathbf{H}_l$, and conducts joint output processing to evaluate the transmitted codeword. For cyclic algebraic code $\mathcal{C}$, $\mathbf{H}_l$ is formed by partitioning the set of codewords of the dual code $\mathcal{C}$ into sets of cyclic shifts of a single codeword. One of these codewords represents a cyclic orbit generator. Hence, the square parity-check matrices $\mathbf{H}_l$ are constructed using the orbit generators with Hamming weight equal to the minimum distance of the dual code. The size of this generator determines the possible number of the $\mathbf{H}_l$ matrix, which varies with the code. The MPPB decoding performs a maximum of i iterations to yield a decoded vector $\underline{\hat{c}}_l$. In a situation where none of the decoders converged to a valid codeword, all output codewords are sent through a least metric selector to decide the best codeword estimate using the decision rule

$$\underline{\hat{c}} = \arg\max_{v \in \mathcal{V}} Pr\{Y = \underline{y} | C = \underline{\hat{c}}_v\}, \tag{2.46}$$

where $\mathcal{V} = \{1, \ldots, l\}$. On the other hand, if l decoders converged to a valid codeword, the output codewords are also passed through the least metric selector using the decision rule (2.46), such that $\mathcal{V} \subseteq \{1, \ldots, l\}$.

In addition to the RRD and MBBP algorithms, a modified redundant iterative HDPC decoding (mRRD) algorithm was proposed in [61]. The mRRD algorithm uses l numbers of belief propagation decoders in parallel whereby each decoder utilizes the same $(n-k) \times n$-dimensional parity-check matrix, but with a random permutation. The initial permutation was randomly selected at the start of each decoding iteration. Unlike the RRD, a fixed damping factor was empirically selected and used during the entire decoding process. Each of the decoders performs I_2 number at the outer loop and a I_1 inner loop. Since the RRD and mRRD use the same parity-check matrix, the complexity of the decoder was obtained by either averaging the number of BP iterations performed before a valid codeword was reached, or till the iteration attains a set threshold in a situation that the decoder does not converge.

Despite the improved decoding performance of the RRD, mRRD and MBBP algorithms, the computational complexities of the algorithms remain high due to the large set of permutation groups and the amount of parallel decoders used for decoding compared to the ABP algorithm. Thus, a low complexity decoding algorithm that also utilizes a set of permutations from the automorphism group of the code during the decoding step was developed for short length linear cyclic codes in [62]. The set of permutations are obtained using the product replacement algorithm [63]. At each iteration, the permuted belief propagation (PBP) algorithm applies the BP decoding algorithm to the received codeword. The iteration terminates when a valid codeword is obtained. If not, a random permutation is chosen from the automorphism group and applied to the output from the previous iteration. The BP decoder is then iterated and the process repeated. It was demonstrated that incorporating the permutation into the message passing process yields faster convergence and produces the same result as using the damping factor in the ABP and mRRD. In addition, the computational cost of generating and implementing permutations has been decreased through suitable mapping of memory addresses during message passing. But the PBP method introduced an additional accumulate step with n summations compared to the ABP algorithm,, resulting in additional decoding complexity.

2.4.2.8 Parity-Check Transformation Algorithm

Until recently, the parity-check matrix transformation algorithm [15] was developed to enhance the performances of the BP decoding algorithm for HDPC codes. The PTA is a simple symbol level iterative SD decoder based on parity-check equations. For a (n, k) RS code, the algorithm utilizes the soft received information, $\underline{r}$ obtained from the channel output to derive the soft reliability information matrix, β. The reliability matrix is computed based on the Euclidean distance d between the signal constellation points of a given modulation scheme and each bit $r_i \in \underline{r}$. Suppose $\underline{s} = [s_0, s_1, \ldots, s_{m-1}]$ are the m-constellation points, the Euclidean distance is defined d as

$$d(s_\epsilon, r_i) = \sqrt{(s_\epsilon - r_i)^2},\ \epsilon = 0, 1, \ldots, m-1, \tag{2.47}$$

which forms the $m \times n$ distance matrix Π,

$$\Pi = \begin{bmatrix} d(s_0, r_1) & d(s_0, r_2) & \ldots & d(s_0, r_n) \\ d(s_1, r_1) & d(s_1, r_2) & \ldots & d(s_1, r_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(s_{m-1}, r_1) & d(s_{m-1}, r_2) & \ldots & d(s_{m-1}, r_n) \end{bmatrix}. \tag{2.48}$$

Let $\Pi(p, q)$ be the elements in the row indexed by s_ϵ and the column indexed by r_i, so Π is normalized along the columns as:

$$\mathcal{N}(p, q) = \frac{e^{-\Pi(p,q)}}{\sum_{q=1}^{n} e^{-\Pi(p,q)}} \mathrel{\hat{=}} P(r_i = \epsilon | c_i). \tag{2.49}$$

The normalization in (2.49) is similar to the softmax function [64], which ensures that the reliability of each bit in the received sequence $\underline{r}$ is observed as probabilities. Hence, the reliability matrix β is derived as:

$$\beta = \begin{bmatrix} \mathcal{N}_{0,1} & \mathcal{N}_{0,2} & \ldots & \mathcal{N}_{0,n} \\ \mathcal{N}_{1,1} & \mathcal{N}_{1,2} & \ldots & \mathcal{N}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{N}_{m-1,1} & \mathcal{N}_{m-1,2} & \ldots & \mathcal{N}_{m-1,n} \end{bmatrix}, \tag{2.50}$$

where the elements $\beta(p,q)$ are the APP values.

The reliability matrix is provided as an input to the PTA decoder to obtain the transformed parity-check matrix. The reliability of each bit c_i is obtained by selecting the more probable (highest) elements in the columns of β, which are then sorted in ascending order to yield:

$$\underline{\beta} = [\beta_1, \beta_2, \ldots, \beta_n], \tag{2.51}$$

where $\beta_1 > \beta_2 > \cdots > \beta_n$. Let the indices of sorting with respect to the order of $\underline{\beta}$ be represented as:

$$I = \{\underbrace{I_1, I_2, \ldots, I_{n-k}}_{U}, \underbrace{I_{(n-k)+1}, \ldots, I_n}_{R}\}. \tag{2.52}$$

The sets U and R are the $(n-k)$ LRPs and k Most reliable positions (MRPs) of the bit reliabilities respectively. Therefore, the initial systematic parity-check matrix is segmented into submatrices of the least reliable bits and the most reliable bits according to the sets U and R. Hence, the matrix $\mathbf{H}$ is transformed as:

$$\hat{\mathbf{H}} = \mathbf{H} \cdot \mathbf{H}_U^{-1}, \tag{2.53}$$

where $\mathbf{H}_U$ is the set of LRBs. The PTA is specifically designed for *RS* codes, which are MDS codes. That is, every set of $n-k$ columns of the parity check matrix of the code is linearly independent [5]. Thus, making it possible to identify a full-rank $(n-k) \times (n-k)$ submatrix $\mathbf{H}_U$ at all times.

The PTA shows enhanced performance in comparison to the KV algorithm and other recognized HD decoders, but it has not been demonstrated that the algorithm will generate better performance as compared to the ABP algorithm. For the class of non-MDS codes, it is not guaranteed that the submatrix $\mathbf{H}_U$ will be invertible since the set of unreliable bits may not necessarily coincide to a linearly independent column, consequently reducing the decoding performance of the PTA.

2.5 Conclusion

A linear code is defined in this chapter as a vector subspace of a n-dimensional binary vector space. This code is denoted as a (n, k) code, that is, a k-dimensional vector subspace of $\mathbb{F}_2^n$. The subspace is determined by specifying k linearly independent vectors as basis vectors from the row space of a $k \times n$ generator matrix $\mathbf{G}$ with rank k. In addition, the subspace is defined using dual vectors in this chapter, which is a very important tool for designing codes with a certain guaranteed minimum distance. More so, the cyclic codes and its properties are reviewed in Chapter 2. Furthermore, detailed literature about related soft-decision decoding of cyclic codes has been discussed in this chapter. The chapter highlights a significant gap in implementing the well-known BP algorithm for decoding HDPC codes. Also, related work that presents modifying the parity-check matrices of cyclic codes for BP decoding are discussed. Moreover, the limitations to these adaptive algorithms are investigated to enable the construction of a modified parity-check transformation algorithm in Chapter 3.

Chapter 3

Modified Parity-Check Transformation Algorithm for Iterative Soft-Decision Decoding

3.1 Introduction

This Chapter presents a relaxed condition for transforming the parity-check matrix of binary cyclic codes. The algorithm is inspired by the conventional parity-check transformation algorithm [15], which requires matrix inversion to transform the matrix of maximum distance separable codes such as RS codes. As discussed in Chapter 2, the parity-check matrices of the MDS codes have every set of redundancy columns to be linearly independent. However, the PTA fails for the class of binary cyclic codes as the vectors of the matrix may not be linearly independent. Therefore, an extended parity-check transformation algorithm EPTA is presented in this chapter to iteratively decode the class of binary cyclic codes. The EPTA avoids matrix inversion by computing the reduced row echelon form of the matrix according to the bit reliabilities.

Similar to the PTA, the proposed algorithm utilizes a soft reliable information matrix obtained from the output of the channel to transform the systematic parity-check matrix at each iteration. Results show a significant performance gain compared to the Berlekamp-Massey (BM) hard decision and belief propagation algorithms. However, the algorithm exhibit a similar BER performance in comparison to the adaptive belief propagation algorithm. An important feature of the proposed decoder is that it functions within a practical decoding time complexity, and can be generally implemented for the class of linear block codes.

3.2 Extended Parity-Check Transformation Algorithm for Binary Cyclic Code

A numerical example can easily demonstrate the execution of the EPTA. Consider the double error correcting (n, k) cyclic code, where $n = 7$ and $k = 3$ with designed distance $d = 4$, generated by the irreducible factor $g(x) = 1 + x^2 + x^3 + x^4$ of $x^7 + 1$ over $GF(2)$. The generator matrix G is constructed from the right cyclic shift of $g(x)$ and represented in its systematic form. Hence, the systematic parity-check matrix $\mathbf{H}$ of this code is given as:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{3.1}$$

Assume that every bit of the codeword $\underline{c} = [1, 0, 0, 1, 0, 1, 1]$ is modulated using BPSK, where 0 is mapped to $+1$ and 1 is mapped to -1. The modulated bits are transmitted over the AWGN channel to obtain a soft received vector, $\underline{r}$. Hence, the initial reliability matrix β is derived according to Equations (2.47)–(2.49) as:

$$\beta = \begin{bmatrix} 0.164 & 0.303 & 0.699 & 0.200 & 0.965 & 0.888 & 0.214 \\ 0.836 & 0.697 & 0.301 & 0.800 & 0.035 & 0.112 & 0.786 \end{bmatrix}. \tag{3.2}$$

Subsequently, the algorithm is allowed to directly operate on β at each iteration in the following steps:

1. *Sorting and Transformation*: An essential step of the EPTA is to determine the reliability of β, and transform the initial parity-check matrix based on the sorted reliabilities. The reliability of each bit $c_i \in \underline{c}$ is obtained by selecting the highest elements from the columns of β in (3.2) as:

$$\underline{L} = \begin{bmatrix} 0.836 & 0.697 & 0.699 & 0.800 & 0.965 & 0.888 & 0.786 \end{bmatrix}, \tag{3.3}$$

which are sorted in ascending order to obtain:

$$\underline{L}' = \begin{bmatrix} 0.697 & 0.699 & 0.786 & 0.800 & 0.836 & 0.888 & 0.965 \end{bmatrix}. \tag{3.4}$$

The indices of the ascending order given as:

$$l = \{2, 3, 7, 4, 1, 6, 5\}, \tag{3.5}$$

such that the $n - k$ indices correspond to the least reliable positions, $U = \{2, 3, 4, 7\}$, and the remaining k indices correspond to the most reliable positions, $R = \{1, 6, 5\}$. Thus, the columns of $\mathbf{H}$ are segmented according to the reliability positions to the least reliable bits and the most reliable bits. The columns of $\mathbf{H}$ that coincide with U generates the submatrix

$$\mathcal{U} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \subset \mathbf{H}, \tag{3.6}$$

while the remaining columns that coincide with R forms the submatrix

$$\mathcal{R} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \subset \mathbf{H}. \tag{3.7}$$

Recall from [5, Lemma 6.6 on p. 293] that the parity-check matrices of the non-MDS codes are not guaranteed to have every set of $n-k$ columns to be linearly independent. Thus, the submatrix $\mathcal{U}$ in Equation (3.6) will not be invertible, since the third and fourth columns are linearly dependent, causing the PTA to fail. To overcome this limitation with the PTA, we first concatenate both $\mathcal{U}$ and $\mathcal{R}$ horizontally to obtain:

$$\mathbf{H}^c = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{3.8}$$

Thereafter, $\mathbf{H}^c$ is reduced to its row echelon form,

$$\mathbf{H}^\star = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{3.9}$$

Hence, we reorder the columns of (3.9), based on the reliability index l to produce a transformed parity-check matrix $\hat{\mathbf{H}}$.

$$\hat{\mathbf{H}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{3.10}$$

Note that the transformed matrices of the ABP and PTA are formed in such a way that the unreliable bits will always participate once in each row of the parity-check matrix. This implies that the matrix will always contain unitary weighted columns at corresponding columns of unreliable bits. However, the column condition on $\hat{\mathbf{H}}$ has been relaxed such that more unreliable bits may participate in the rows of the transformed parity-check matrix. As a result, corresponding columns of unreliable bits do not necessarily contain unitary weighted columns.

2. *Message Passing*: The message passing stage is performed in two steps.

Step 1 - Parity-check Equations: Using $\hat{\mathbf{H}}$ and β, we compute the parity-check equations p_j by dot multiplying the jth row of the transpose of $\hat{\mathbf{H}}$ by the result of the hard decision detection on β to yield scalar values:

$$p_j = \underline{b} \cdot \underline{h}_j, \;\; j = 1, 2, \ldots, n-k, \tag{3.11}$$

where $\underline{h}_j$ are the row vectors of $\hat{\mathbf{H}}^T$, and $\underline{b}$ is the output of hard decision HD detection on β. From Equation (3.2), we obtain the HD detection as:

$$\underline{b} = [1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1]. \tag{3.12}$$

Therefore, each parity-check equation is calculated as

$$p_1 = [1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1] \cdot [0,\ 1,\ 0,\ 1,\ 1,\ 0,\ 0] = 0 \tag{3.13a}$$

$$p_2 = [1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1] \cdot [0,\ 0,\ 1,\ 1,\ 1,\ 1,\ 0] = 1 \tag{3.13b}$$

$$p_3 = [1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1] \cdot [0,\ 0,\ 0,\ 1,\ 0,\ 1,\ 1] = 0 \tag{3.13c}$$

$$p_4 = [1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1] \cdot [1,\ 0,\ 0,\ 0,\ 1,\ 1,\ 0] = 1. \tag{3.13d}$$

Equations (3.13a)–(3.13d) generate a syndrome vector $\underline{S} = \begin{bmatrix} p_1, & p_2, & p_3, & p_4 \end{bmatrix}$, which determines the stopping condition for the iterative decoder. The iteration terminates whenever $\underline{S}$ is equal to zero, thus $\underline{b}$ is returned as the correct codeword. On the other hand, if the syndrome vector is not equal to zero, β is refined based on the results of p_j in the next step.

Step 2 - Reliability update: An important characteristic of the PTA decoder is that the decoder will always have one participating unreliable symbol in the jth column of $\hat{\mathbf{H}}$, while the remaining symbols will participate in the corresponding MRPs. Thus, the highest column entry of β corresponding to the LRP is fully penalized/rewarded, while the remaining corresponding entries to the MRPs are partially penalized/rewarded based on an updating factor δ. However, the relaxed condition of the EPTA decoder allows the jth row of $\hat{\mathbf{H}}$ to contain more than one participating unreliable bits. Consequently, an equal updating factor is applied to the corresponding column entries of β to produce an updated reliability matrix $\hat{\beta}$. This implies that if $p_j = 0$, a selected δ is added to the highest entries in the columns of β that corresponds to the participating bits in the jth row of $\hat{\mathbf{H}}^T$. Otherwise, δ is subtracted from the corresponding column entries of β.

Considering Equations (3.13a)–(3.13d), the first parity-check equation checks, that is, $p_1 = 0$. Thus, the highest β-values at corresponding columns to the participating bits in h_1 are increased by $\delta = 0.001$ to form:

$$\beta_{p_1} = \begin{bmatrix} 0.164 & 0.303 & 0.699 & 0.200 & \boxed{0.966} & 0.888 & 0.214 \\ 0.836 & \boxed{0.698} & 0.301 & \boxed{0.801} & 0.035 & 0.112 & 0.786 \end{bmatrix}. \tag{3.14}$$

Moreover, $p_2 = 1$, meaning it does not check. Therefore, the corresponding β-values of participating bits in h_2 row are decreased by $\delta = 0.001$ to yield:

$$\beta_{p_2} = \begin{bmatrix} 0.164 & 0.303 & \boxed{0.698} & 0.200 & \boxed{0.965} & \boxed{0.887} & 0.214 \\ 0.836 & 0.698 & 0.301 & \boxed{0.800} & 0.035 & 0.112 & 0.786 \end{bmatrix}. \tag{3.15}$$

Similar checks are performed for p_3 and p_4 to obtain the respective refined reliability matrices:

$$\beta_{p_3} = \begin{bmatrix} 0.164 & 0.303 & 0.698 & 0.200 & 0.967 & \boxed{0.888} & 0.214 \\ 0.836 & 0.698 & 0.301 & \boxed{0.801} & 0.035 & 0.112 & \boxed{0.787} \end{bmatrix}. \tag{3.16}$$

$$\beta_{p_4} = \begin{bmatrix} 0.164 & 0.303 & 0.698 & 0.200 & \boxed{0.966} & \boxed{0.887} & 0.214 \\ \boxed{0.835} & 0.698 & 0.301 & 0.801 & 0.035 & 0.112 & 0.787 \end{bmatrix}. \tag{3.17}$$

The updated reliability matrix $\hat{\beta} = \beta_{p_4}$ is returned as input to the iterative decoder till the syndrome vector is equal to zero or a maximum number of iteration N is attained. The operations of the EPTA is summarized in Algorithm 1.

Algorithm 1: EPTA algorithm for iterative decoding

```
Input: $\underline{r}$, $\mathbf{H}$, $n$, $k$, $N$, $\delta$.
Output: Decoded vector $\hat{b}$.
Sorting and transformation
compute: $\beta$, $\underline{L}$, $\underline{L}'$, $l$, $U$, $R$ ;
obtain: $\mathcal{U}$, $\mathcal{R}$ ;
derive: $\mathbf{H}^c = \left[\mathcal{U} \quad \mathcal{R}\right]$;
form: $\mathbf{H}^*$ based on Gaussian elimination;
derive: $\hat{\mathbf{H}} = l[\mathbf{H}^*]$;
Message passing
repeat
    compute: $\underline{b}$, $p_j$ and $\underline{S}$;
    $\underline{S}$ = false || $N$= false;
    repeat
        perform: step 2;
    until $\underline{S}$ = true || $N$= true;
until $\underline{S}$ = true || $N$= true;
    $\hat{\underline{b}} = \underline{b}$
```

Furthermore, the simulation environment for the EPTA is discussed in Appendix A, while the iterative steps of the EPTA is outlined in a simple flow chart as shown in Figure 3.1.

3.3 Simulation Results

3.3.1 Error rate performance analysis for different δ

We analyze the choice of the updating factor δ as it determines the number of iterations and the bit error rate performance of the EPTA decoder. We assume different δ values and investigate the performance of a high rate $(15, 11)$ binary cyclic code for the same data using the BPSK constellation scheme with AWGN.

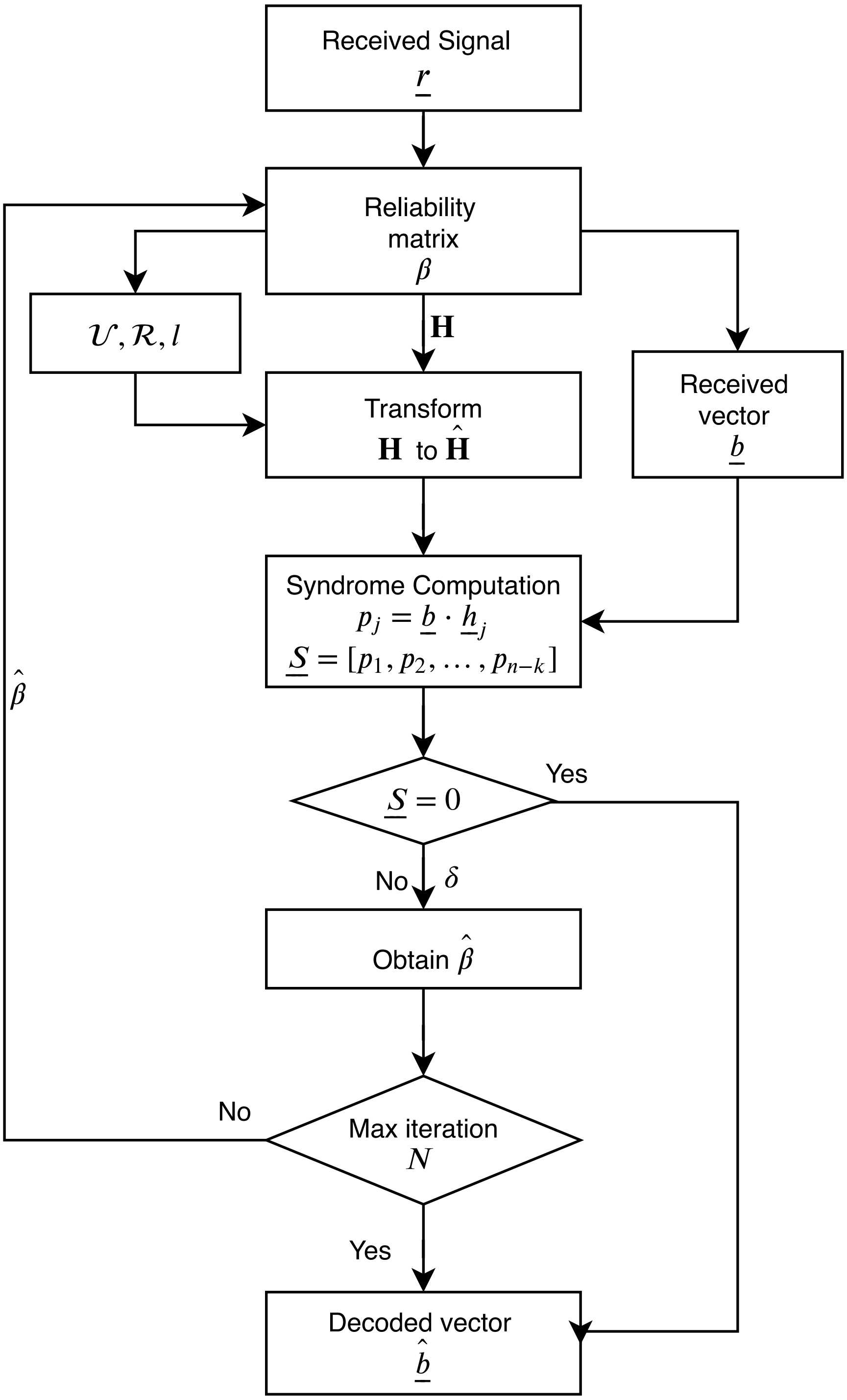

Figure 3.1: The EPTA Soft-Decision Decoder

Figure 3.2 shows that the smallest δ-value ($\delta = 0.001$) requires more iterative steps, while $\delta = 0.9$ goes through the least number of iterations. $\delta = 0.9$ saturates before properly refining the reliability matrix and thus, converges to a wrong codeword.

However, Figure 3.3 shows that $\delta = 0.05$ is computationally efficient, since it yields a similar decoding performance with the smallest δ value and requires moderate number of iterations. Thus, we assumed $\delta = 0.05$ as the incremental and decremental operator for the EPTA decoder.

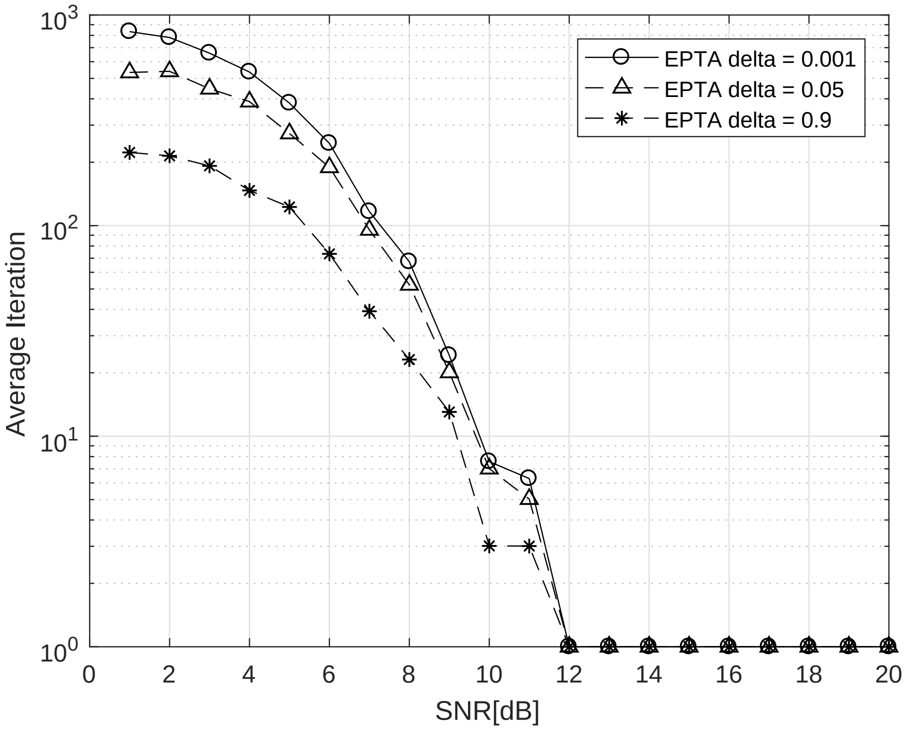

FIGURE 3.2: Average number of iterations of the EPTA with different δ values

3.3.2 Performance comparison of the EPTA with similar scheme

In Figure 3.4, the BER performance of the proposed EPTA is compared with the hard decision BM algorithm, traditional Belief Propagation BP, and the symbol-level PTA. We consider a medium rate $R = 0.4667$ BCH $(15, 7)$ code. The EPTA exhibits a performance gain of 1.5dB, 2.8dB, and 4.6dB in comparison with the BM, PTA, and BP decoders respectively. Basically, BP exhibits the least decoding performance since the parity-check matrices of binary cyclic codes contain cycles of length 4 in the associated Tanner graphs. Moreover, the PTA offers a performance loss of 1.3dB and 2.8dB in comparison with the conventional BM algorithm and the EPTA respectively.

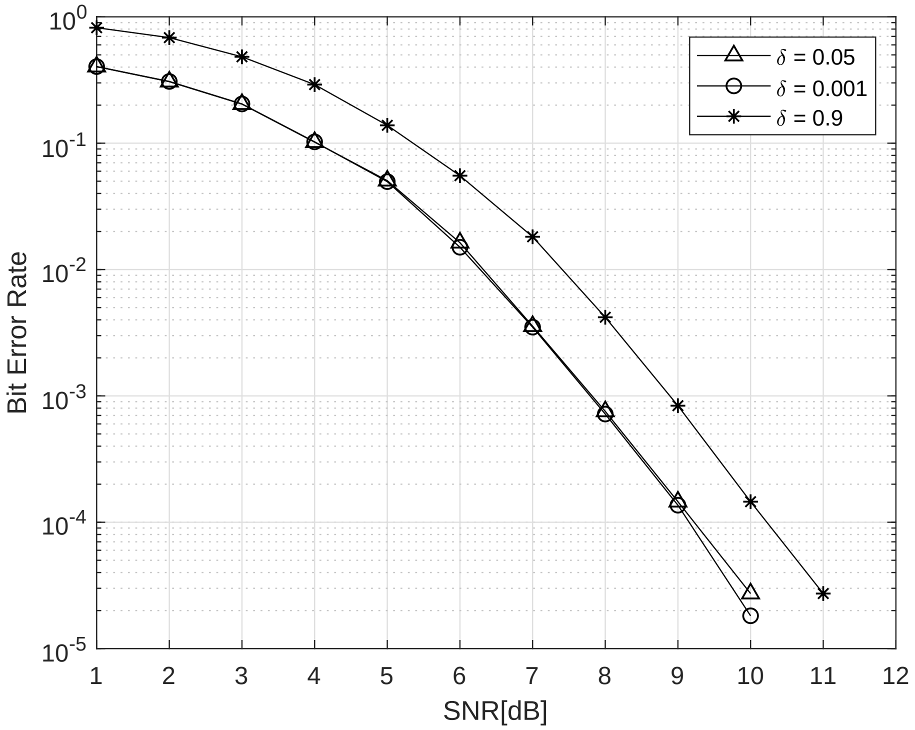

FIGURE 3.3: Performance comparison of the EPTA with different δ values

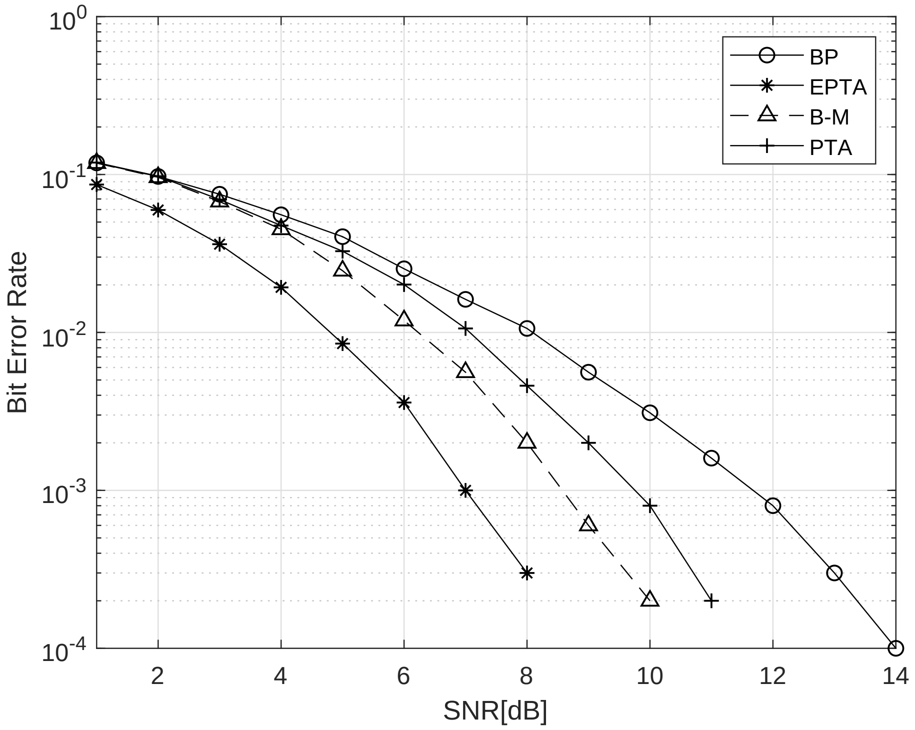

FIGURE 3.4: Performance comparison: BCH code $(15, 7)$, $R = 0.4667$

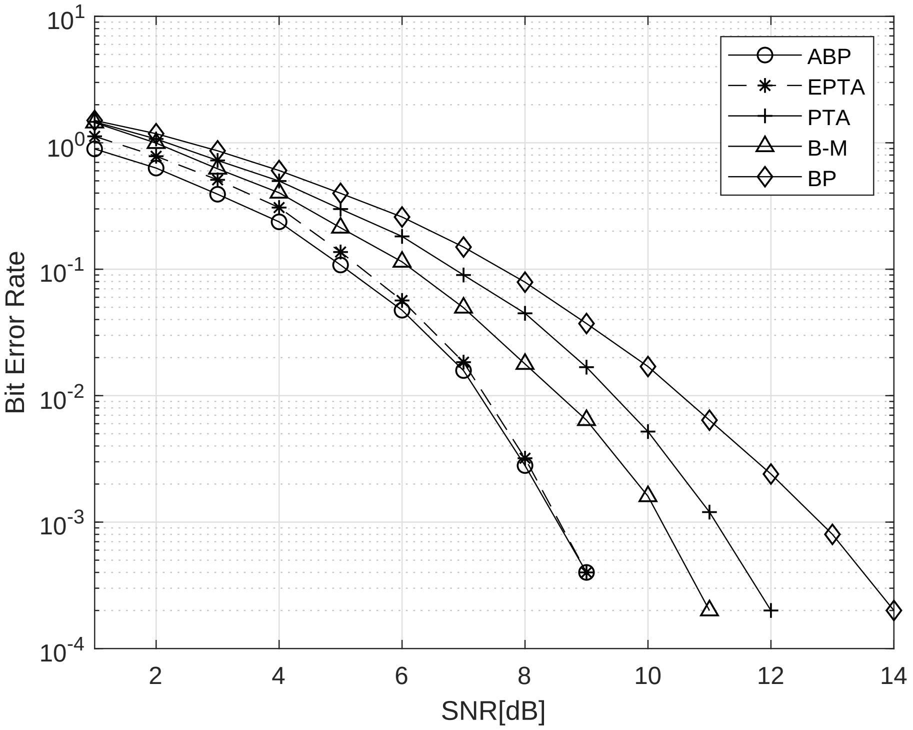

FIGURE 3.5: Performance comparison: BCH $(15, 11)$ code

This is due to the non-invertible submatrix that occurs at the matrix transformation stage, thus confirming Hypothesis 1.

Figure 3.5 shows the simulation result for a high rate $R = 0.7333$, BCH $(15, 11)$ code. The ABP is presented as a comparison benchmark for the proposed EPTA and the other algorithms. As compared to the BM and BP algorithms, the EPTA exhibits a performance gain of 3.2dB and 5.3dB at a BER of 10^{-3} respectively. Also, the algorithm yields a similar BER performance in comparison to the ABP algorithm at a BER of 10^{-3}. Moreover, the proposed algorithm exhibits a 2.7dB performance gain over the PTA, which is as a result of the non-invertible submatrix at the transformation stage.

Furthermore, a BCH$(31, 21)$ code of rate $R = 0.6774$ is considered in Figure 3.6. At a BER of 10^{-3}, the EPTA algorithm exhibits a similar performance compared to the ABP, and performance gain of 2.5dB in comparison with the PTA. Likewise, the EPTA produces about 1.5dB and 4.8dB performance gain compared to the BM and BP algorithms respectively. This depicts the proposed EPTA as a robust decoding

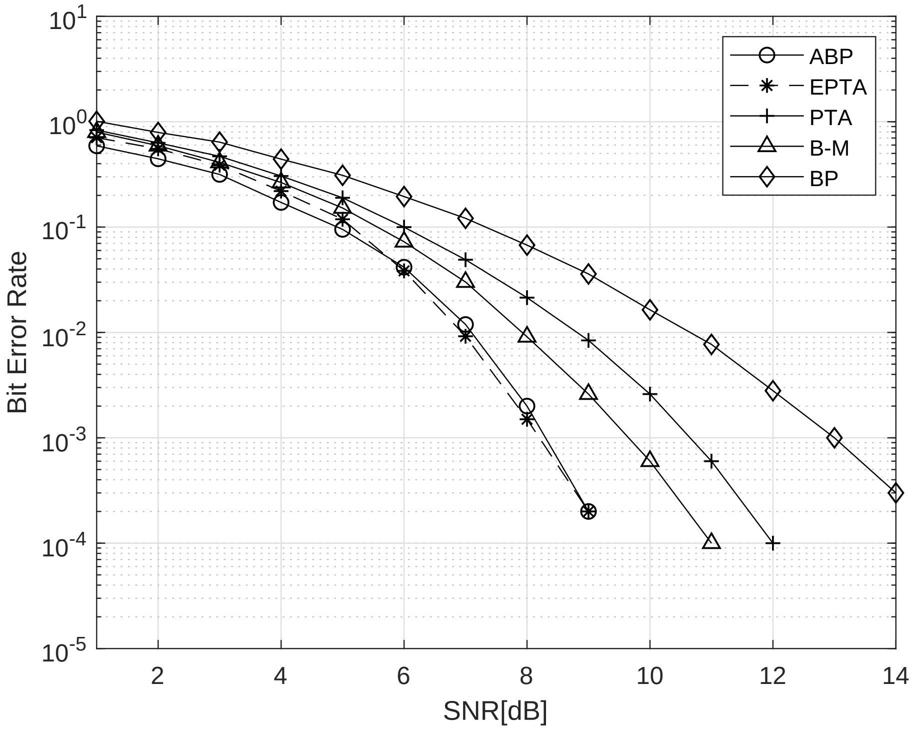

FIGURE 3.6: Performance comparison: BCH $(31, 21)$ code, $R = 0.6774$

algorithm for binary cyclic codes and can be realized on a real-time coding scheme. The algorithm can easily be generalized to codes with symbols from a non-binary field.

TABLE 3.1: Worst-case time complexity analysis

Steps	**Big O notation**		
	EPTA	**PTA**	**ABP**
Sorting reliability	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
Matrix Transformation	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
Extrinsic information generation	-	-	$\mathcal{O}(n^2)$
Bit-level reliabilities update	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Hard-decision	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

3.4 Worst-case complexity analysis

The worst-case time complexity of the PTA, EPTA, and ABP algorithms is presented using the big-$\mathcal{O}$ notation in Table 3.1. Some of the most used sorting algorithms include the bubble sort, selection sort, insertion sort, and quick sort algorithms. Therefore, applying any of these sorting algorithms to determine the bit reliability during the matrix transformation processes of the PTA, EPTA and ABP result in worst-case time complexity of $\mathcal{O}(n^2)$. Note that MATLAB uses the quick sort algorithm that depends on the system and the numerical data used in the sort.

Moreover, computing the reduced row echelon form of the parity-check matrix takes a large component of the per-iteration time complexity. For instance, solving a system of n equation with n unknowns by carrying out row operations on the matrix till it is reduced to its echelon form, and thereafter solving each unknown in reverse order involves $n(n+1)/2$ divisions, $(2n^3+3n^2-5n)/6$ multiplications, and $(2n^3+3n^2-5n)/6$ subtractions [65, p. 12]. Therefore, a maximum of approximately $2n^3/3$ operations is required for a Gaussian elimination. Hence, the PTA, EPTA, and ABP have a worst-case computational complexity of $\mathcal{O}(n^3)$ at the matrix transformation stage.

Nevertheless, the ABP algorithm generates the extrinsic information vector by applying the SPA using the transformed parity-check matrix and the ordered LLR vector. The number of summations and multiplications required by applying the SPA is quadratic complexity, that is, of order $\mathcal{O}(n^2)$ [66]. Compared to the PTA and EPTA, there is no extrinsic information generation step. Thus, the main computational difference between the EPTA and the ABP algorithm is the extrinsic information generation step.

3.5 Conclusion

An iterative soft-decision decoder based on the transformed parity-check matrix was designed for the class of cyclic codes. The algorithm is an extension of the symbol-level parity-check transformation algorithm, which fails for non-MDS codes. The proposed extended parity-check transformation algorithm introduced a relaxed transformed matrix at each iteration to enhance the performance of the PTA for non-MDS codes. Furthermore, a new method of refining the reliability matrix from the channel's output was implemented based on the relaxed condition. Results showed that carefully selecting the updating factor reduces the number of decoding iteration as compared to the PTA. But, more work must be done to further decrease the required number of iterations. Moreover, despite that the proposed EPTA offer the same error performance as the ABP, the algorithm exhibits a reduced computational complexity at the message passing stage. More so, the algorithm exhibited an improved decoding performance as compared to the traditional HD algorithm and other soft decision decoding algorithms. The performance of the algorithm can be further improved by developing a more efficient matrix transformation technique for the HDPC codes as shown in Chapter 4.

Chapter 4

A Generalized Parity-check Transformation Algorithm for Iterative Soft-Decision Decoding

4.1 Introduction

The conventional parity-check matrix transformation algorithm requires a matrix inversion to transform matrices of MDS codes. However, such transformation does not always hold for the non-MDS codes. As a result, the extended parity-check transformation algorithm was presented in Chapter 3 to resolve the PTA limitation. The EPTA algorithm in Chapter 3 utilizes a moderate updating factor to reduce the number of iterations needed for decoding. This is contrary to the PTA that implements a very small updating factor, thus requires a very large number of iterations. Nevertheless, the number of iterations of the EPTA exceeds that of the ABP. Hence, a generalized parity-check matrix transformation algorithm is developed in this chapter to enhance the computational efficiency of the EPTA.

Similar to the EPTA, the GPT avoids the matrix inversion of the PTA. Rather, it permutes the columns of the parity-check matrix based on the reliability information from the channel's output. Results show a reasonable performance gain as compared to the other SD decoding algorithms. In addition, the decoder functions within a practical decoding time complexity at both the matrix transformation stage and the message passing stage. The BER performance of the algorithm is detailed by way of computer simulations over the AWGN channel.

4.2 Parity-Check Transformation Algorithm for Binary Cyclic Code

We consider a binary cyclic code $\mathcal{C}(n, k)$, of length n and dimension k. From Chapter 2, the standard form of the parity-check matrix is given in terms of the binary coefficients of $h(x)$ as:

$$\mathbf{H} = \begin{bmatrix} h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \end{bmatrix}. \tag{4.1}$$

This matrix contains many length 4-cycles, thus resulting in poor performance of the BP decoding [67]. Consequently, the ABP utilizes Gaussian elimination, while the PTA uses matrix inversion based on the bit-reliability to reduce the number of length 4-cycles. Since $\mathbf{H}$ is full rank, there are at least $(n-k)$ independent columns in $\mathbf{H}$ to be reduced to weight one column. These reduced (sparse) columns form an identity submatrix $\mathcal{U}$, such that each of the columns correspond to an unreliable bit. However, for the non-MDS codes, it is not always guaranteed that all the columns associated to the unreliable bits will be reduced during matrix adaptation [14]. In this case, the performance of the iterative SD decoder decreases. Specifically, the PTA inverts $\mathcal{U}$ and multiplies the inverted matrix by $\mathbf{H}$ to generate the transformed matrix. This method of transformation fails whenever $\mathcal{U}$ is not invertible. Therefore, a variation of the PTA was presented in Chapter 3. The EPTA produces a transformed matrix by way of concatenation, row reduction, and matrix column rearranging based on the bit reliabilities. Compared to the PTA, the algorithm showed an improved performance but goes through more decoding iterations than the ABP. Hence, we propose the generalized parity-check transformation technique algorithm in this chapter, intended to reduce computational complexity and enhance the decoding performance of the EPTA.

4.3 GPT algorithm for binary cyclic codes

The proposed GPT consists of two stages, namely; the matrix transformation stage and the message passing stage.

4.3.1 Matrix transformation.

Let $\underline{c} = [c_1, c_2, \cdots, c_n]$ be a codeword of a binary cyclic code $\mathcal{C}$. Assume the bits of the codeword are modulated using the BPSK, with 0 mapped to $+1$ and 1 mapped to -1. The modulated vector is transmitted over an AWGN channel to obtain vector $\underline{r}$:

$$\underline{r} = (-2\underline{c} + 1) + \phi, \tag{4.2}$$

where ϕ is the vector of statistically independent Gaussian random variables with zero mean and variance $N_0/2$. Thus, we represent the initial reliability of each bit in the received vector $\underline{r}$ as a reliability matrix β:

$$\beta = \begin{bmatrix} P(c_1 = 0|r_1) & P(c_2 = 0|r_2) & \cdots & P(c_n = 0|r_n) \\ P(c_1 = 1|r_1) & P(c_2 = 1|r_2) & \cdots & P(c_n = 1|r_n) \end{bmatrix}. \tag{4.3}$$

Each column of the reliability matrix contains a pair of normalized APP values $P(c_i = 0|r_i)$ and $P(c_i = 1|r_i)$. Note that for the ABP decoding, the APP values are used to determine the lLLR values of c_i as:

$$L(c_j) = \log \frac{P(c_i = 0|r_i)}{P(c_i = 1|r_i)}. \tag{4.4}$$

The vector of the LLRs is given as

$$\underline{L} = [L(c_1), L(c_2), \ldots, L(c_n)], \tag{4.5}$$

such that the magnitude of $\underline{L}$ is sorted in ascending order. Hence, the parity-check matrix is reduced to a transformed form based on Equation (4.5).

However, the GPT directly operates on the reliability matrix in (4.3) to transform $\mathbf{H}$. The highest APP value in the columns of β are selected and sorted in ascending order of reliability to yield $\underline{\beta}$:

$$\underline{\beta} = [\beta_1, \beta_2, \ldots, \beta_n], \tag{4.6}$$

with $\beta_1 > \beta_2 > \cdots > \beta_n$ and $\beta_i \in \beta,\ i = 1, 2, \ldots, n$. Thus, the index of sorting are recorded in order of reliability as:

$$\mathcal{I} = \underbrace{i_1, i_2, \ldots, i_{n-k}}_{\rho}, \underbrace{i_{n-k+1}, \ldots, i_n}_{\sigma}, \tag{4.7}$$

where segments ρ and σ correspond to the least reliable values and the most reliable values of $\underline{\beta}$ respectively. To reduce the density and remove part of the short cycles in (4.1), we introduce the process of permuting columns of $\mathbf{H}$ based on $\mathcal{I}$ as:

$$\mathbf{H}^{\iota} = \mathcal{I}[\mathbf{H}] = [\mathbf{h}_1^{\iota}\ \mathbf{h}_2^{\iota}\ \cdots\ \mathbf{h}_n^{\iota}], \tag{4.8}$$

where $\mathbf{h}_i^{\iota}$ represents the ith column vector of $\mathbf{H}^{\iota}$. Since each reliability value β_i of $\underline{\beta}$ is associated with the ith column $\mathbf{h}_i^{\iota}$ of $\mathbf{H}^{\iota}$, we refer to β_i as the corresponding reliability value of $\mathbf{h}_i^{\iota}$. Hence, we perform elementary row operations on $\mathbf{H}^{\iota}$ to obtain a systematic matrix $\mathbf{H}^s$:

$$\mathbf{H}^s = [I_{(n-k)}P], \tag{4.9}$$

where $I_{(n-k)}$ is the $(n-k) \times (n-k)$ identity matrix and P is the $(n-k) \times k$ parity-check matrix. The columns of $\mathbf{H}^s$ are then rearranged according to $\mathcal{I}$, resulting to a transformed parity-check matrix $\hat{\mathbf{H}}$:

$$\hat{\mathbf{H}} = \mathcal{I}[\mathbf{H}^s] = [\hat{\mathbf{h}}_1 \ \hat{\mathbf{h}}_2 \ \cdots \ \hat{\mathbf{h}}_n]. \tag{4.10}$$

4.3.2 Message passing.

Let h_{ji} be the entry of matrix $\hat{\mathbf{H}}$. The ABP generates extrinsic information for each bit using the SPA based on the transformed parity-check matrix $\hat{\mathbf{H}}$ as:

$$L_{ext}(c_i) = \sum_{\substack{j=1 \\ h_{ji}=1}}^{n-k} 2\tanh^{-1}\left(\prod_{\substack{p=1 \\ p \neq i, h_{jp}=1}}^{a} \tanh\left(\frac{L(c_p)}{2}\right)\right). \tag{4.11}$$

Thereafter, the bit reliability is updated as discussed in Chapter 2. However, the GPT algorithm directly updates the bit reliability in β based on the information provided by the participating bits in the i-th row of $\hat{\mathbf{H}}$. The process is summarized in three steps.

Step 1: Suppose $a_{w\mu}$ are the bit reliability entries of β in (4.3). Let b represent the corresponding index of the highest entry $a'_{w\mu}$ at the μth column of β. For a BPSK scheme, b can either be 0 or 1. Hence, an estimate vector $\underline{b}$ of the received codeword $\underline{c}$ is obtained as:

$$\underline{b} = \left[b\widehat{=}a'_{w1},\ b\widehat{=}a'_{w2},\ \ldots,\ b\widehat{=}a'_{wn}\right]. \tag{4.12}$$

Thus, we calculate each parity-check equation as:

$$p_j = \underline{b}\hat{\mathbf{H}}^T, \ j = 1, 2, \dots, n-k. \tag{4.13}$$

Step 2: Since each entry $a'_{w\mu}$ corresponds to a participating bit $\hat{\mathbf{h}}_i$ in the jth row of $\hat{\mathbf{H}}$, we update the bit reliability based on results from Equation (4.13). If p_j yields zero, the highest bit reliability $a'_{w\mu}$ in each column of β is updated by adding a updating factor δ. Otherwise, β is updated by subtracting δ from $a'_{w\mu}$.

Step 3: The GPT algorithm is iterative and a stopping rule is given such that the syndrome vector equal zero, that is, $\underline{S} = [p_1 = 0, p_2 = 0, \dots, p_{(n-k)} = 0]$, then the decoding process terminates and produces $\hat{\underline{b}}$ as its correct codeword. Otherwise, the updated bit reliability matrix $\hat{\beta}$ returns as input to the decoder until a maximum number of iteration (N) is attained. The operations of the GPT is summarized in Algorithm 2 and Figure 4.1. More so, the simulation environment of the GPT is shown in Appendix A

Algorithm 2: GPT algorithm for iterative decoding

```
Input: $\underline{r}$, $\mathbf{H}$, $n$, $k$, $N$, $\delta$.
Output: Decoded vector $\hat{\underline{b}}$.
Matrix transformation
compute: $\beta$, $\underline{\beta}$, $\mathcal{I}$ ;
compute: $\mathbf{H}^{\iota}$, $\mathbf{H}^{s}$ based on (4.8) and (4.9);
derive: $\hat{\mathbf{H}} = \mathcal{I}[\mathbf{H}^{s}]$;
Message passing
repeat
    compute: $\underline{b}$, $p_j$, and $\underline{S}$;
    $\underline{S}$ = false || $N$= false;
    repeat
        perform: step 2;
    until $\underline{S}$ = true || $N$= true;
until $\underline{S}$ = true || $N$= true;
    $\hat{\underline{b}} = \underline{b}$
```

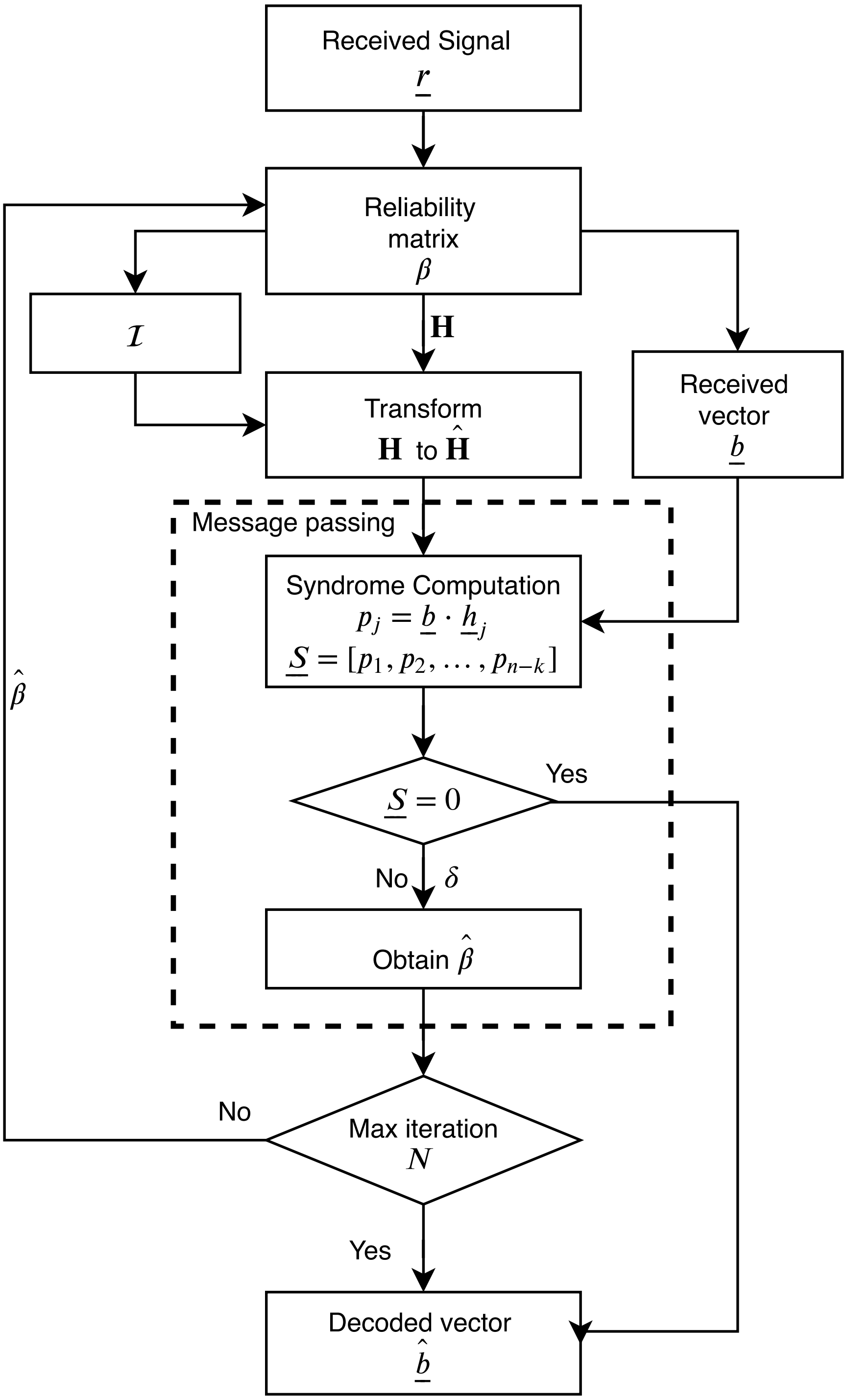

FIGURE 4.1: The GPT Soft-Decision Decoder

4.4 Simulation Results

We present simulation results for the BCH codes using BPSK modulation with different rates R, length n and dimension k, over an AWGN channel. The BER performance of the proposed GPT is compared to other iterative soft-decision decoding algorithms for the class of binary cyclic codes. The following representations are used in the legends. GPT(N_1) symbolizes the proposed generalized parity-check transformation algorithm, where N_1 is the maximum number of iterations. ABP(N_1, N_2) represents the adaptive belief propagation in [14], where N_2 is the number of regrouping of unreliable bits. PTA(N_1) refers to the traditional parity-check matrix transformation for MDS codes in [15]. mRRD(N_1, N_3) refers to the modified random redundant decoding in [61], where N_3 is the maximum number of iterations per outer iteration with each iteration utilizing a different permutation. PBP represents the permuted belief propagation algorithms in [62].

As shown in Chapter 3, the number of iterations and BER performance of the GPT algorithm is determined by δ. Similarly, we simulate the BCH$(31, 16)$ code to select an appropriate δ-value for the algorithm. Figure 4.2 shows that the number of iterations in the GPT algorithm increases as the δ-values become smaller. As shown, $\delta = 0.5$ requires the minimum number of iterations in comparison to the other δ-values, but it offers the worst BER performance in Figure 4.3. However, $\delta = 0.1,\ 0.05,\ 0.01,\ 0.005$, and 0.001 yield the same BER performance. Nevertheless, considering Figure 4.3, $\delta = 0.1$ requires the minimum number of iterations in comparison to $\delta = 0.05,\ 0.01,\ 0.005$, and 0.001. Thus, we carefully select $\delta = 0.1$ as the updating factor for the proposed algorithm since it is computationally efficient.

We present the results for medium rate, $R = 0.5161$, BCH $(31, 16)$ code in Figure 4.4. For this code, the ML decoding curve is presented as a comparison benchmark for the proposed GPT and other iterative SD decoders. The GPT(10) approaches the ML curve within 0.5dB at a BER of 10^{-3}. Also, it reasonably compares to the performances of the mRRD$(15, 50)$ with fifteen parallel decoders and the PBP(5) algorithms. The mRRD utilizes 50 iterations to attain a 0.1dB performance gain over the GPT with 10 iterations and the PBP with 5 iterations. The decoding performance of the GPT algorithm and the EPTA are compared to show the difference between the two algorithms. The GPT utilizes 10 iterations compared to the EPTA with 100 iterations. More so, the GPT exhibits a performance gain of about 0.5dB at a BER of 10^{-4}. As compared to the algebraic HD decoder, PTA(10), ABP$(20, 1)$, and the regrouping ABP$(20, 3)$, the GPT(10) exhibits a performance gain of 2.7dB, 1.7dB, 0.4dB, and 0.1dB at a BER of 10^{-4} respectively. The PTA exhibits the worst SD decoding performance due to the formation of a non-invertible matrix at the transformation stage. Thus, it converges to a wrong codeword. Note that the ABP$(20, 3)$ offers a performance gain of about 0.4dB in comparison to the ABP$(20, 1)$. This performance gain comes with increased decoding computational complexity because of the increase in the number of regroupings.

In Figure 4.5, we consider a medium rate $R = 0.5714$, BCH $(63, 36)$ code. At a BER of 10^{-4}, the GPT(10) algorithm matches the performances of the mRRD$(15, 50)$ and the PBP(5) algorithms. Note that the ML is not included since it becomes impracticable as the code length increases. Again, the GPT exhibits about 0.4dB, 0.8dB and 1.8dB performance gain in comparison to the ABP$(20, 3)$, ABP$(20, 1)$ and the PTA respectively.

Figure 4.6 shows the simulation results for a high rate $R = 0.9059$, BCH $(255, 231)$

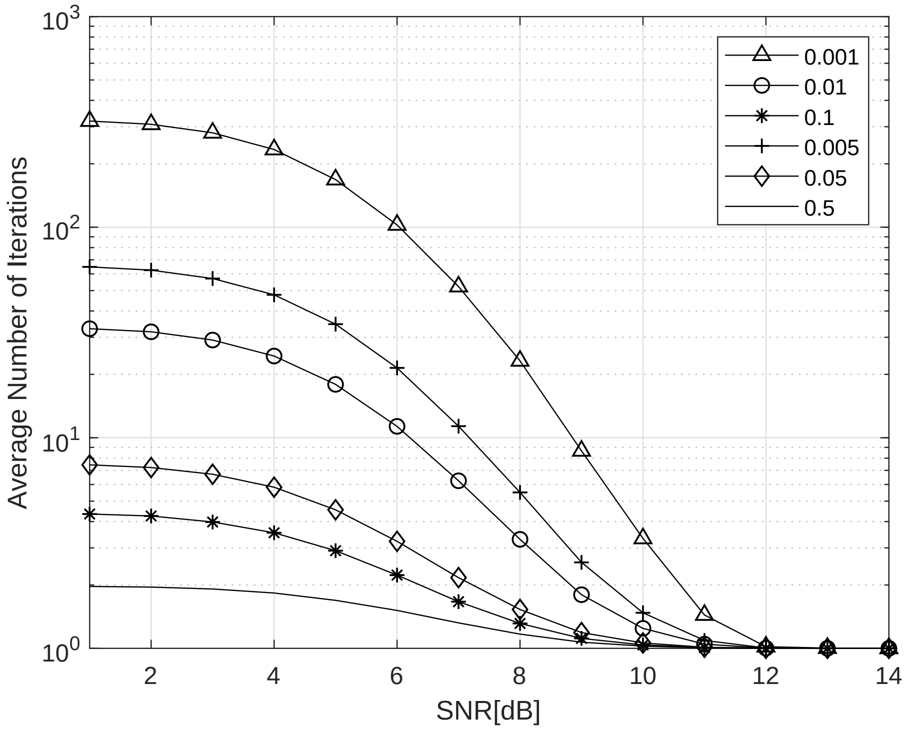

FIGURE 4.2: Average number of iterations for different δ values.

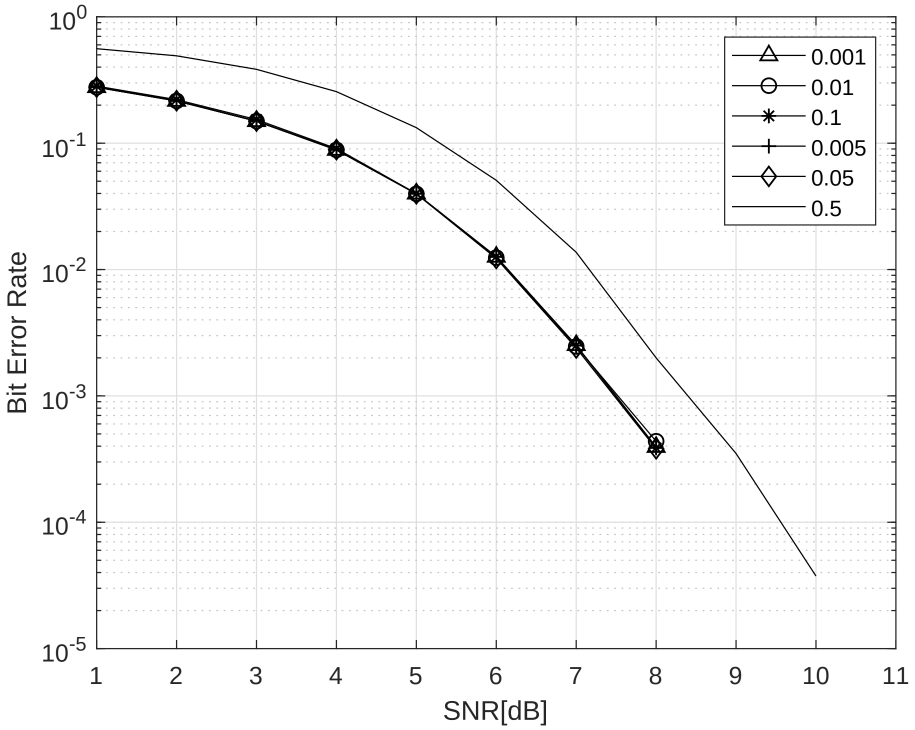

FIGURE 4.3: Performance comparison of the GPT for different δ values.

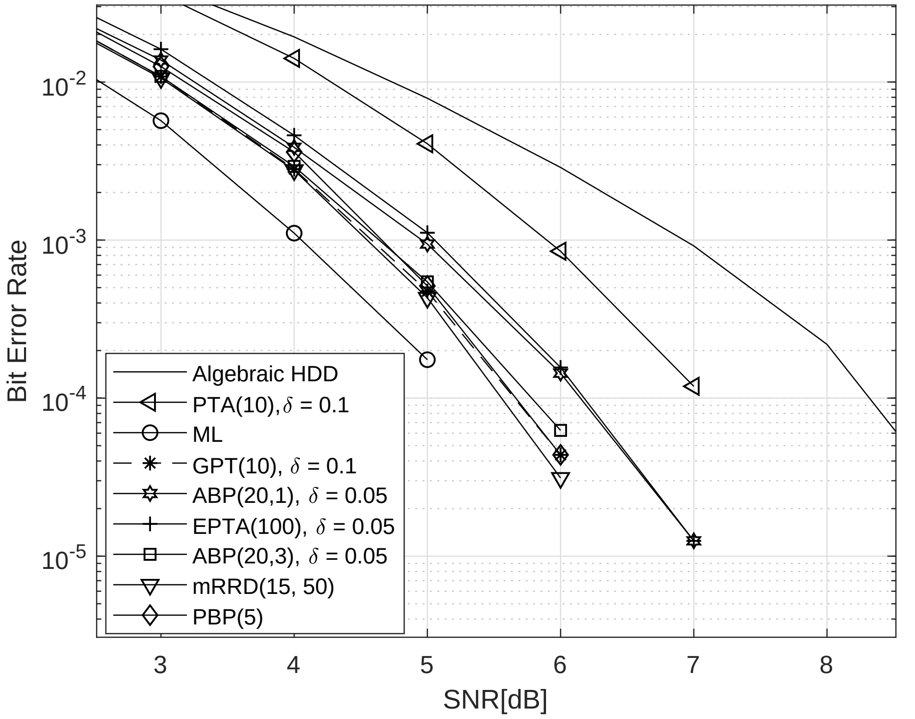

FIGURE 4.4: BER performance comparison: BCH code $(31, 16)$, $R = 0.5161$.

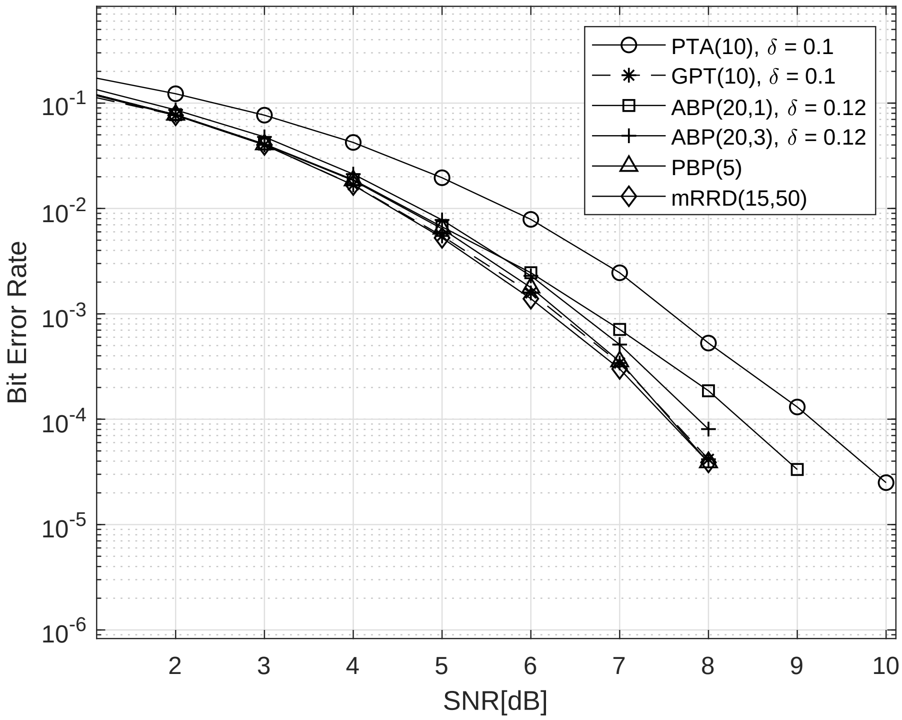

FIGURE 4.5: BER performance comparison: BCH code $(63, 36)$, $R = 0.5714$.

code. In comparison to the mRRD$(15, 50)$ and the PBP(5), the proposed GPT algorithm utilizes the maximum of 10 iterations to yield a performance gain of 0.6dB, at a BER of 10^{-4}. Likewise, the algorithm exhibits a performance gain of about 0.5dB, 1.1dB, 1.4dB and 2.3dB in comparison to the ABP$(80, 50)$, ABP$(20, 3)$, ABP$(20, 1)$, and the conventional PTA respectively. This portrays the GPT as a performance efficient decoding algorithm. Therefore, it can be used in real-time coding system as it exhibits reasonable decoding time complexity as shown in Section 4.5.

4.5 Worst-case complexity analysis

As compared to the permuted belief propagation algorithm and the modified random redundant decoding, the GPT only permutes the columns of the parity-check matrices based on the bit reliability at each iteration. The complexity of the PBP algorithm increases by computing n required summations in the accumalate step to replace the updating factor of the GPT. Also, the PBP generates a permutation

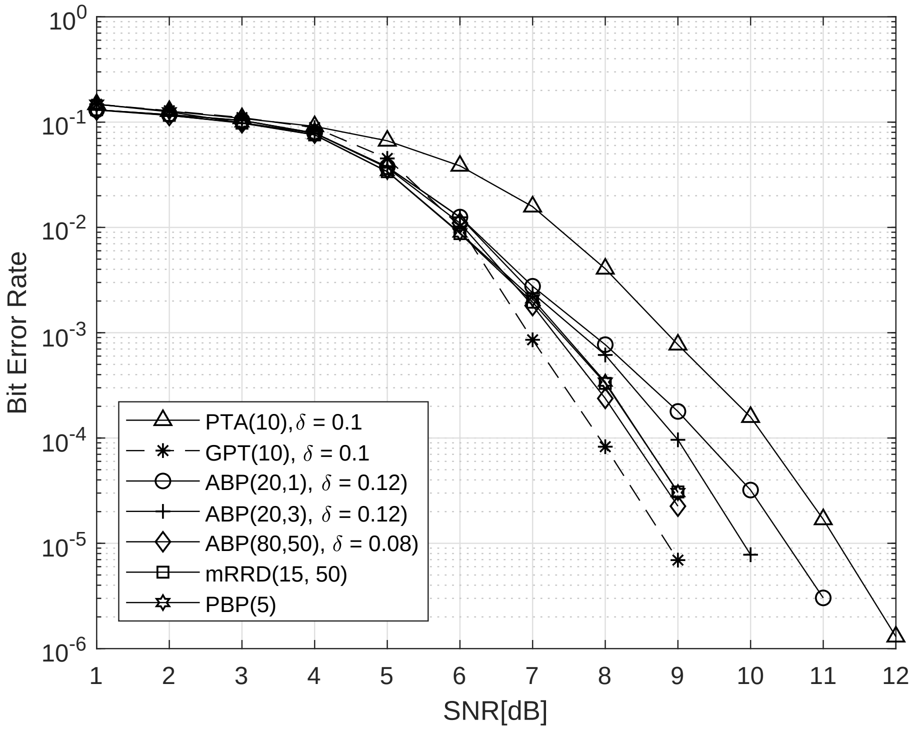

FIGURE 4.6: BER performance comparison: BCH code $(255, 231)$, $R = 0.9058$.

step, which are added to the belief propagation algorithm. On the other hand, the mRDD's complexity is assessed based on the number of parallel decoders and the number of BP iterations required for the decoder to converge to a codeword. The mRRD requires 15 parallel decoders and 50 iterations on average to achieve the same decoding performance as the GPT algorithm.

In addition, the worst-case time complexity analysis of the GPT, PTA and ABP algorithms is presented in Table 4.1 using the big O notation. As shown, ABP has more computational complexity as compared to the GPT and PTA. This is largely attributed to the computation of extrinsic information during the check-node update, which is not required by the PTA and GPT algorithms. Thus, aside from obtaining a better BER performance, the GPT also exhibits a reduced computational time complexity in comparison to the other considered.

TABLE 4.1: Worst-case time complexity analysis

Steps	**Big O notation**			
	GPT	**EPTA**	**PTA**	**ABP**
Sorting reliability	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
Transformation	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
Extrinsic information generation	-	-	-	$\mathcal{O}(n^2)$
Bit-level reliability update	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Hard-decision	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

4.6 Conclusion

In this chapter, we developed a generalized parity-check transformation algorithm for iterative soft-decision decoding of binary cyclic codes. As compared to the mRRD and PBP, the proposed GPT does not utilize a wider set of permutations of the complete automorphism group. However, it transforms the initial parity-check matrix of a code by permuting the columns according to the bit reliability. The GPT algorithm carefully selects the updating factor, which significantly reduces the number of decoding iterations. Thus, the algorithm has a low time complexity, which is applicable for real-time communication systems. In addition, simulation results show that the proposed GPT algorithm significantly outperforms the conventional PTA and it offers a reasonably BER performance gain over the remaining SD decoders.

Despite the improved decoding performance of the GPT alforithm as compared with other SD algorithms, the BER performance of the GPT and other similar algorithms fall below the ML curve. This is due to the transformed parity-check matrix used at each decoding iteration. Hence, a perfect knowledge model is developed in Chapter 5 to analyze the performance of the transformed matrices.

Chapter 5

A Baseline Parity-Check Matrix for Iterative Soft-Decision Decoding of Binary Cyclic Codes

5.1 Introduction

The results from Chapters 3 and 4 indicate the need to further construct the best parity-check matrix to decode the class of cyclic codes in order to produce near-ML decoding performances. Due to the rapid interest in the idea of reliability based parity-check matrix transformation for BP decoding, the scope of this chapter is centered at analyzing the performances of the transformed parity-check matrices used in the ABP and GPT. The ABP and GPT algorithms are used as test beds for this analysis. Therefore, the performances of the transformed matrices are examined based on the proposed perfect knowledge model.

This chapter verifies that the iterative SD decoders utilize sub-optimal parity-check matrices for decoding in Hypothesis 4. To do this, we introduce the perfect knowledge model PKM to analyze the performance of the transformed parity-check matrix for iterative SD decoders. The model obtains the most suitable parity-check matrix of a code, which validates Hypothesis 5. Here, all the possible parity-check matrices of the code are computed according to a given channel condition, and an optimal parity-check matrix is chosen based on a minimum distance criteria. Also, we show by using a numerical example, that the optimal matrix from the PKM does not necessarily contain unit vectors at corresponding columns of the most unreliable received bits. This contradicts the assumptions of the transformed parity-check matrix in [14] and [15]. Thus, we emphasize that the iterative SD decoding algorithm of cyclic codes can still be improved by finding the appropriate transformed parity-check matrix. In addition, the PKM can be used as a benchmark for evaluating the performance of iterative decoders based on the transformed parity-check matrix.

5.2 Perfect Knowledge Model (PKM)

We introduce an empirical perfect knowledge model to determine the baseline parity-check matrix for iterative soft decision decoding and to show that the matrix does not necessarily contain unitary weight at the corresponding columns of the unreliable bits. The PKM is based on a distance metric, which determines the optimality of the parity-check matrix from a list of all possible matrices. Also, numerical results are presented to demonstrate the empirical model.

5.2.1 A baseline parity-check matrix

Let $\mathcal{C}^{\perp}$ be the dual of the code $\mathcal{C}(n,k)$ over $GF(2)$. A codebook ζ of all 2^{n-k} possible codewords is generated from $\mathcal{C}^{\perp}$ in a similar way to the ML exhaustive search table [5]:

$$\zeta = \{\underline{c}_1, \underline{c}_2, \cdots, \underline{c}_{2^{n-k}}\}, \ \underline{c} \in \mathcal{C}^{\perp}. \tag{5.1}$$

Also, a matrix $\mathcal{H}$, which rows are all permutations of choosing $(n-k)$ codewords of ζ with order and without repetitions, is formed from (5.1). The matrix contains a list of all the possible $(n - k \times n)$-dimensional parity-check matrices, $\mathbf{H}_\tau \in \mathcal{H}$.

$$\mathbf{H}_\tau = \{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_\iota\},\ \iota = \frac{\eta!}{(\eta - (n-k))!}, \tag{5.2}$$

where η is the number of codewords of ζ.

Furthermore, the initial reliability matrix (β_0) of the transmitted codeword is derived according to Equation (2.50). Thus, a hard decision detection is performed on β_0 to obtain the vector,

$$\underline{b} = \left[y \hat{=} \mathcal{N}'_{p,1},\ y \hat{=} \mathcal{N}'_{p,2},\ \ldots,\ y \hat{=} \mathcal{N}'_{p,n}\right], \tag{5.3}$$

where y is the index of the highest entry $\mathcal{N}'_{p,q}$ at the qth column of β_0. Hence, syndrome checks are performed on the rows of each matrix $\mathbf{H}_\tau \in \mathcal{H}$ as:

$$\underline{S}_\tau = \underline{b} \cdot \mathbf{H}_\tau^T,\ \tau = 1, 2, \cdots, \iota, \tag{5.4}$$

where $\mathbf{H}_\tau^T$ is the transpose of individual matrix $\mathbf{H}_\tau$. Moreover, the bit reliabilities of β_0 are updated based on the outcomes from the set of syndrome vectors $\underline{S}_\tau$. If a parity-check equation of $\mathbf{H}_\tau$ is equal to zero, $\mathcal{N}'_{p,q}$ in the qth column of β_0, which coincides to a participating bit in the parity-check equation, is rewarded by adding an updating factor δ. Otherwise, these entries would be penalized by subtracting the same δ value. Therefore, a corresponding set of reliability matrices is generated as:

$$\beta_\tau = \{\beta_1, \beta_2, \cdots, \beta_\iota\}, \tag{5.5}$$

based on β_0, $\mathbf{H}_\tau$ and $\underline{S}_\tau$.

Note that a basic assumption in the development of this model is that the transmitted codewords $\underline{c}$ are known. Thus, the codeword can be interpreted as a sequence of observations, which are subsequently converted to a matrix of observations,

$$\mathcal{A} = \begin{bmatrix} P(c_1|s_0) & P(c_2|s_0) & \cdots & P(c_n|s_0) \\ P(c_1|s_1) & P(c_2|s_1) & \cdots & P(c_n|s_1) \\ \vdots & \vdots & \ddots & \vdots \\ P(c_1|s_{m-1}) & P(c_2|s_{m-1}) & \cdots & P(c_n|s_{m-1}) \end{bmatrix}. \tag{5.6}$$

Matrix $\mathcal{A}$ signifies the probability of observing a bit b from a row indexed by the signal points s_ϵ, $\epsilon = 0, 1, \ldots, m-1$. Since the model has a prior knowledge of $\underline{c}$, the probability of observing b at the row indexed by $s_b \in \underline{s}$ will be equal to one, that is, $P(b|s_b) = 1$. Therefore, we can assume without loss of optimality that the PKM selects the baseline candidate parity-check matrix from the list of matrices $\mathbf{H}_\tau \in \mathcal{H}$, according to a defined distance metric $\mathcal{D}$. The metric is characterized by the matrix of observations $\mathcal{A}$ and the list of reliability matrices β_τ, $\tau = 1, 2, \ldots, \iota$. Let ϖ and π denote the row indices of $\mathcal{A}_{j,i} = 1$ and $\mathcal{A}_{j,i} = 0$ respectively. The set of distances $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_\iota\}$ between $\mathcal{A}$ and β_τ is given as:

$$\mathcal{D}_\tau = \sqrt{\sum_{i=1}^{n} (\mathcal{A}(\varpi, i) - \beta_\tau(\varpi, i))^2} + \sqrt{\sum_{i=1}^{n} (\mathcal{A}(\pi, i) - \beta_\tau(\pi, i))^2}. \tag{5.7}$$

Hence, the optimum parity-check matrix is one, which minimizes $\mathcal{D}$ as:

$$\mathbf{H}_{\gamma\in\tau} = \underset{\tau}{\operatorname{argmin}}\ \mathcal{D}(\mathcal{A}, \beta_\tau). \tag{5.8}$$

Moreover, the least of the distances may be indexed at more than one places. This implies that the model produces multiple candidate parity-check matrices. Therefore, any of the candidate parity-check matrices could be chosen as the baseline matrix. This matrix has a low probability that the erroneous bits will participate in the PCEs since the PKM has a prior knowledge of the transmitted codeword.

Furthermore, the optimality of the transformed parity-check matrix $\hat{\mathbf{H}}$ is analyzed based on the following scenarios with respect to the frequency of occurrence of $\hat{\mathbf{H}}$ in the set of selected baseline matrices $\mathbf{H}_\gamma$. Consider the case where one or more members of $\mathbf{H}_\gamma$ has weight one columns at the corresponding least reliable positions. As a result, the transformed parity-check matrix, $\hat{\mathbf{H}}$, of the iterative soft-decision decoder is categorized as the best matrix for BP decoding. There are other cases where the selected baseline matrices $\mathbf{H}_\gamma$ do not have unitary weighted columns at corresponding LRPs. This implies that $\hat{\mathbf{H}}$ is not a member of $\mathbf{H}_\gamma$, thus it is considered to be suboptimal for the BP decoding. Consequently, we infer that the iterative SD decoding algorithms either utilize the suboptimal matrix or not, depending on the frequency of occurrences of $\hat{\mathbf{H}}$ in the set of optimal matrices $\mathbf{H}_\gamma$. The PKM process is hereby demonstrated using a numerical example.

5.2.2 Numerical Example

Consider a $(7, 4)$ cyclic code, C, over $GF(2)$, with generator polynomial $x^3 + x + 1$. The codebook ζ of all possible linear combination of codewords, $\underline{c}$, (except the all zero codewords) is obtained as:

$$\zeta = \begin{array}{c} {\scriptstyle 1} \\ {\scriptstyle 2} \\ {\scriptstyle 3} \\ {\scriptstyle 4} \\ {\scriptstyle 5} \\ {\scriptstyle 6} \\ {\scriptstyle 7} \end{array} \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5.9}$$

The indices (i, j, k) of choosing $n - k$ codewords of ζ with order and without repetitions is permuted to obtain the matrix of all $(n - k \times n)$ possible parity-check matrices:

$$\mathcal{H} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 4 \\ \vdots \\ 7 \\ 6 \\ 5 \end{array} \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{5.10}$$

Each sequence $\{i, j, k\}$ of row combination represents a matrix $\mathbf{H}_\tau \in \mathcal{H}$, $\tau = 1, 2, \cdots, 210$,

$$\mathbf{H}_\tau = \begin{pmatrix} \mathbf{H}_{\{1,2,3\}} \rightarrow \mathbf{H}_1 \\ \mathbf{H}_{\{1,2,4\}} \rightarrow \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_{\{7,6,5\}} \rightarrow \mathbf{H}_{210} \end{pmatrix}. \tag{5.11}$$

Assume that the bits of $\underline{c} = [0\ 0\ 1\ 1\ 1\ 0\ 1]$ are modulated using the BPSK and transmitted over the AWGN channel. The initial reliability matrix from the channel's output is derived from Equation (2.50) as:

$$\beta_0 = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0.8229 & 0.8808 & 0.2806 & 0.1192 & 0.1192 & 0.3616 & 0.2169 \\ 0.1771 & 0.1192 & 0.7194 & 0.8808 & 0.8808 & 0.6384 & 0.7831 \end{bmatrix} \end{array}. \tag{5.12}$$

Thus, we obtained the hard decision vector $\underline{b}$ by selecting the indices of the highest elements $\mathcal{N}'_{p,q}$ at the qth columns of β_0 as:

$$\underline{b} = [0,\ 0,\ 1,\ 1,\ 1,\ 1,\ 1]. \tag{5.13}$$

Hence, syndrome checks are performed on the rows of each matrix $\mathbf{H}_\tau$ as:

$$\underline{S}_\tau = \begin{pmatrix} \underline{b}\mathbf{H}^T_{\{1,2,3\}} \rightarrow \underline{S}_1 = [1,\ 1,\ 0] \\ \underline{b}\mathbf{H}^T_{\{1,2,4\}} \rightarrow \underline{S}_2 = [1,\ 1,\ 0] \\ \vdots \\ \underline{b}\mathbf{H}^T_{\{7,6,5\}} \rightarrow \underline{S}_{210} = [1,\ 0,\ 0] \end{pmatrix}. \tag{5.14}$$

Since each syndrome vector $\underline{S}_\tau$ contains the results of the $n-k$ parity-check equations $(p_j, j = 1, 2, \ldots, n-k)$ from the individual matrices $\mathbf{H}_\tau$. Therefore, the syndromes are used to update the initial reliability matrix β_0 based on the results of each p_j from $\mathbf{H}_\tau$.

Starting with $\tau = 1$, that is, $\underline{S}_1 = [1,\ 1,\ 0]$. The first PCE (p_1) of $\mathbf{H}_{\{1,2,3\}}$ is equal to one, thus $\delta = 0.1$ is subtracted from $\mathcal{N}'_{p,q}$ at corresponding qth columns to the participating bits in the first row of $\mathbf{H}_{\{1,2,3\}}$. Similarly, p_2 of $\mathbf{H}_{\{1,2,3\}}$ yields one, therefore $\delta = 0.1$ is subtracted from $\mathcal{N}'_{p,q}$ at the corresponding qth columns to the participating bits in the second row of $\mathbf{H}_{\{1,2,3\}}$. However, the p_3 of $\mathbf{H}_{\{1,2,3\}}$ yields zero, so $\delta = 0.1$ is added to $\mathcal{N}'_{p,q}$ at corresponding qth columns to the participating bits in the third row of $\mathbf{H}_{\{1,2,3\}}$. The process of updating β_0 is performed for all $\underline{S}_\tau$ to generate the list of updated reliability matrices β_τ,

$$\beta_\tau = \left\{ \begin{matrix} \beta_1 \\ \begin{bmatrix} 0.8129 & 0.8708 & 0.2806 & 0.1192 & 0.1192 & 0.3616 & 0.2169 \\ 0.1771 & 0.1192 & 0.7294 & 0.8808 & 0.8708 & 0.6184 & 0.7831 \end{bmatrix} \\ \beta_2 \\ \begin{bmatrix} 0.8129 & 0.8708 & 0.2806 & 0.1192 & 0.1192 & 0.3616 & 0.2169 \\ 0.1771 & 0.1192 & 0.7194 & 0.8908 & 0.8808 & 0.6184 & 0.7731 \end{bmatrix} \\ \vdots \\ \beta_{210} \\ \begin{bmatrix} 0.8329 & 0.9008 & 0.2806 & 0.1192 & 0.1192 & 0.3616 & 0.2169 \\ 0.1771 & 0.1192 & 0.7194 & 0.8908 & 0.8908 & 0.6284 & 0.7831 \end{bmatrix} \end{matrix} \right\}. \tag{5.15}$$

Nevertheless, let the transmitted codeword $\underline{c} = [1\ 0\ 0\ 1\ 1\ 1\ 0]$ be represented as a sequence of bit observations. The bit sequence is converted to a matrix of observations $\mathcal{A}$, such that the probability of observing a bit b at the row indexed by s_b is 100%.

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{5.16}$$

Thus, the set of distances between the observation of a bit in $\mathcal{A}$ and the corresponding reliability values in β_τ is derived from (5.7) as:

$$\mathcal{D}_\tau = \begin{pmatrix} \mathcal{D}_1 = 1.7628 \\ \mathcal{D}_2 = 1.7164 \\ \vdots \\ \mathcal{D}_{210} = 1.6921 \end{pmatrix} \tag{5.17}$$

In this case, $\mathcal{D}$ attains its smallest value, that is, $\mathcal{D}_\tau = 1.5888$ at indices $\gamma_{\{2,3,4\}}$, $\gamma_{\{2,3,6\}}$, $\gamma_{\{2,5,6\}}$, $\gamma_{\{4,5,6\}}$. Therefore, the parity-check matrices at indices corresponding to $\gamma_{\{i,j,k\}}$ are selected as the candidate matrices. This implies that

$$\mathbf{H}_{\gamma_{\{i,j,k\}}} = \left\{ \begin{array}{l} \mathbf{H}_{\{2,3,4\}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \\ \mathbf{H}_{\{2,3,6\}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \\ \mathbf{H}_{\{2,5,6\}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \\ \mathbf{H}_{\{2,3,4\}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array} \right\}, \tag{5.18}$$

where $\mathbf{H}_\gamma$ contains all the optimal parity-check matrices, which are suitable for BP decoding.

For this example, the frequency of occurrences of $\hat{\mathbf{H}}$ in the list of optimal matrices, $\mathbf{H}_\gamma$ is observed to show that the iterative SD decoders utilize a suboptimal transformed parity-check matrix. Here, the transformed parity-check matrix $\hat{\mathbf{H}}$ of the GPT is obtained as:

$$\hat{\mathbf{H}}_{\{2,4,5\}}(GPT) = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \qquad (5.19)$$

while that of the ABP is given as:

$$\hat{\mathbf{H}}_{\{2,4,5\}}(ABP) = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \qquad (5.20)$$

From Equations (5.19) and (5.20), it is observed that both the GPT and ABP utilize the same transformed matrix with the unit columns of $\hat{\mathbf{H}}_{\{2,4,5\}}$ coinciding with the $n-k$ LRPs ($U = \{3,6,7\}$ in β_0). In addition, it is found that $\hat{\mathbf{H}}_{\{2,4,5\}}$ does not belong to $\mathbf{H}_\gamma$. Hence, we deduce that $\hat{\mathbf{H}}_{\{2,4,5\}}$ is suboptimal for decoding the $(7,4)$ BCH code.

TABLE 5.1: Frequency of occurrences (f) of the GPT's $\hat{\mathbf{H}}$ matrix in the set of optimal $\mathbf{H}_\gamma$ matrices

SNR	l_{max}	BCH (7,4)		BCH (15,11)	
		f	$f_o(\%)$	f	$f_o(\%)$
1	10000	1065	10.65	165	1.65
2	10000	1085	10.85	180	1.80
3	10000	1099	10.99	177	1.77
4	10000	1195	11.95	178	1.78
5	10000	1079	10.79	180	1.80
6	10000	1678	16.78	180	1.91
7	10000	1877	18.77	195	1.95
8	10000	2080	20.80	199	1.97
9	10000	2091	20.91	295	2.26
10	10000	2165	21.65	295	2.65

TABLE 5.2: Frequency of occurrences (f) of the ABP's $\hat{\mathbf{H}}$ matrix in the set of optimal $\mathbf{H}_\gamma$ matrices

SNR	l_{max}	BCH (7,4)		BCH (15,11)	
		f	$f_o(\%)$	f	$f_o(\%)$
1	10000	1065	10.65	165	1.65
2	10000	1026	10.26	177	1.77
3	10000	1095	10.95	177	1.77
4	10000	1197	11.97	178	1.78
5	10000	1078	10.78	180	1.80
6	10000	1677	16.77	191	1.91
7	10000	1877	18.77	195	1.95
8	10000	2080	20.80	197	1.97
9	10000	2091	20.91	226	2.26
10	10000	2165	21.65	265	2.65

Furthermore, Tables 5.1 and 5.2 show the number of times that the $\hat{\mathbf{H}}$ matrices of the GPT and ABP occur in the list of $\mathbf{H}_\gamma$ matrices, using a maximum of $10,000$ iterations. Also, the percentage of the frequency occurrence (f_o) is given in Tables 5.1 and 5.2, such that the average percentage of occurrences (O_{avg}) is computed as:

$$O_{avg} = \frac{\sum f_o}{\#SNR}, \tag{5.21}$$

where $\#SNR$ is the number of SNR used. The observation results show that both the GPT and ABP have a similar frequency of occurrences, which increases with respect to the SNR.

For the medium rate $(7,4)$ BCH codes, $\hat{\mathbf{H}}(GPT)$ and $\hat{\mathbf{H}}(ABP)$ both occurred more frequently in the list of $\mathbf{H}_\gamma$ as compared to the high rate $(15,11)$ codes. In the case of the GPT algorithm, Table 5.1 shows that the medium rate code has 15.414% of $\hat{\mathbf{H}}$ in $\mathbf{H}_\gamma$ compared to the high rate code, which has 1.951% of $\hat{\mathbf{H}}$ occurring in the list of optimum matrices $\mathbf{H}_\gamma$. Similarly, the ABP algorithm in Table 5.2 indicates that the medium rate $(7,4)$ code has 15.3% occurrence of $\hat{\mathbf{H}}$ in $\mathbf{H}_\gamma$. On the other hand, the high rate $(15,11)$ code is observed to contain 1.7% of $\hat{\mathbf{H}}$ in $\mathbf{H}_\gamma$. Consequently, we infer from Tables 5.1 and 5.2 that the baseline parity-check matrix for the iterative decoding does not necessarily contain unitary weighted columns at the corresponding columns of the unreliable bits. More so, the frequency of occurrences shows that the iterative SD decoders frequently utilize the suboptimal transformed matrices for iterative soft-decision decoding the class of cyclic codes.

5.3 Simulation Results

We present the simulation results for iterative decoding of binary cyclic codes with length n and dimension k based on the GPT and ABP algorithms. The BPSK modulation is used and the modulated vector is sent over the AWGN channel. The simulation environment is presented in Appendix A. In the legend, GPT($\mathbf{H}_t$) and ABP($\mathbf{H}_t$) utilizes the matrix $\hat{\mathbf{H}}$, while GPT($\mathbf{H}_\gamma$) and ABP($\mathbf{H}_\gamma$) are results from adopting the baseline parity-check matrix derived from the PKM. The δ value of 0.1 is used and specified on the plot, unless otherwise indicated. In all cases, the iteration runs until $\underline{S}_\gamma = [p_1, p_2, \ldots, p_{n-k}]$ is equal to zero or till a fixed number of iterations ($l_{max} = 10,000$ iterations) is attained.

We present result for the BCH $(15, 11)$ code in Figure 5.1. Here, the standard algebraic HD decoding outperforms the traditional BP algorithm. It is clear that the parity-check matrix of the BCH codes is highly dense. Thus, the matrix is transformed to $\hat{\mathbf{H}}$ at each decoding iteration, which is utilized by both the GPT($\mathbf{H}_t$) and ABP($\mathbf{H}_t$). The GPT($\mathbf{H}_t$) and ABP($\mathbf{H}_t$) yield performances close to the ML curve and exhibit approximately 2.2dB and 3.4dB performance improvement in comparison to the HD decoder and BP performances at a BER of 10^{-4}. The result shows that the transformed parity-check matrix is crucial to attaining a near ML decoding.

Moreover, it suffices to show that the transformed matrix $\hat{\mathbf{H}}$, used by the GPT and ABP in Figure 5.1 is sub-optimal. This can be proved by substituting the $\hat{\mathbf{H}}$ with the baseline parity-check matrix $\mathbf{H}_\gamma$, derived from the PKM. The BER performance of utilizing the $\mathbf{H}_\gamma$ is shown in Figure 5.2. Figure 5.2 indicates that using the baseline parity-check matrix, the GPT($\mathbf{H}_\gamma$) and ABP($\mathbf{H}_\gamma$) offers a performance gain of 1.3dB, 1.4dB, and 3.5dB in comparison to the PTA($\mathbf{H}_t$), ABP($\mathbf{H}_t$), and algebraic HD decoder respectively at a BER of 10^{-4}. Also, adopting the baseline

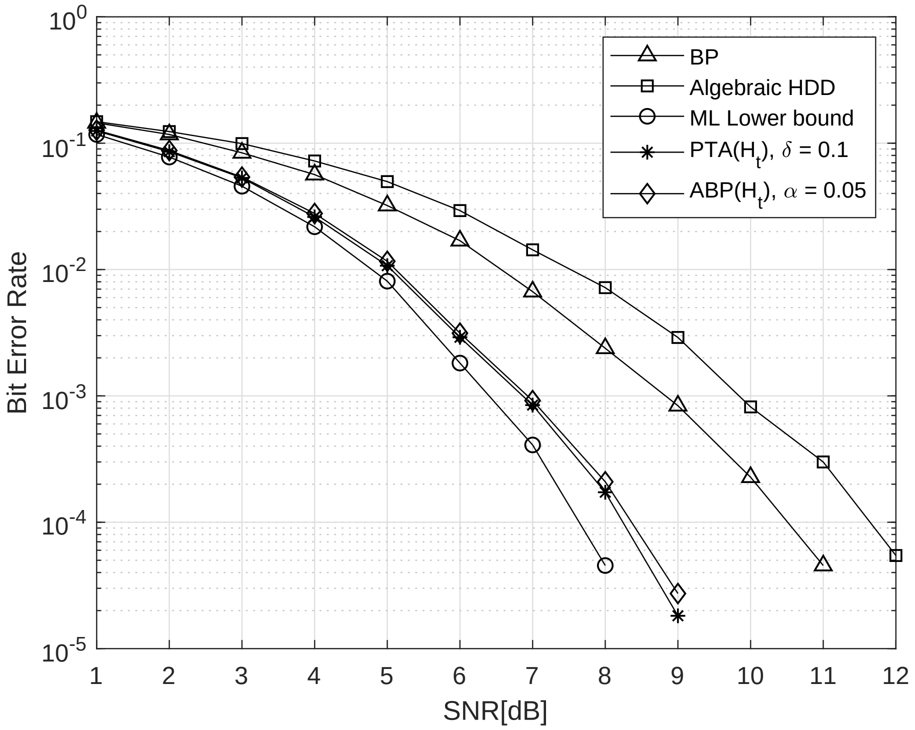

FIGURE 5.1: Performance comparison of the $(15, 11)$ binary BCH code

.

matrix yields a performance increase of 0.7dB in comparison to optimal lower bound ML decoding. Consequently, instead of the ML decoding, the PKM can be used as a baseline to verify performances of newly developed iterative SD decoders based on parity-check equations.

Furthermore, we note that the baseline parity check matrix $\mathbf{H}_\gamma$ is optimal, which signify that there is a possibility of obtaining an enhanced parity-check matrix for BP decoding. It is not certain that this desired or unknown matrix will achieve the same performance as the baseline matrix, since $\mathbf{H}_\gamma$ is obtained from a perfect knowledge model. However, this matrix can produce a performance that will be closer to the baseline parity-check matrix and potentially exceeds the ML decoding.

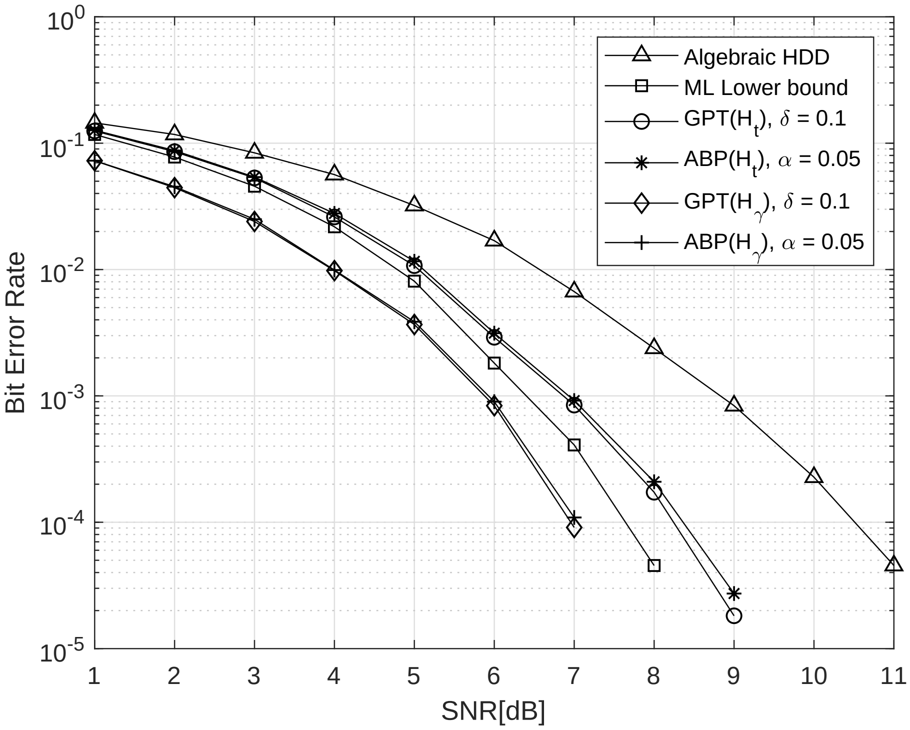

FIGURE 5.2: Performance comparison of the $(15, 11)$ binary BCH code

.

5.4 Conclusion

We investigated the existing parity-check transformation of the iterative soft decision decoders for the class of binary cyclic codes. We showed that the ABP and GPT utilizes a suboptimal transformed parity-check matrix for error corrections. In order to analyze the transformed parity-check matrix, we introduced a perfect knowledge model PKM. The model determines the optimal parity-check matrix based on prior knowledge of transmitted codewords. Computer search results on the $(15, 11)$ BCH code indicate that the choice of an optimal parity-check matrix guarantees an improved decoding performance in comparison with the ML decoder.

The PKM involves large computations since the number of parity-check matrices grows exponentially. Also, it is not suitable for practical decoding due to the assumption that it has prior knowledge of the transmitted codewords. Future research can strive to construct the optimal parity-check matrix without generating the list of possible matrices. More so, it is important to have a model that selects the baseline parity-check matrix without the knowledge of the transmitted codeword.

Chapter 6

Conclusion

Chapter 6 discusses the general contribution of this thesis. First, a description of the study's aim and achievement is summarized. Subsequently, the results of each chapter are summarized to give an overview of the precise contribution of each chapter. Also, this chapter provide recommendations and ideas for future research possibilities.

6.1 Summary and Key Results

This thesis entitled Soft Decision Decoding of Moderate Length Binary Cyclic Codes based on Parity-Check Transformation illustrates the general objective of this research. Specifically, the study aims at enhancing the efficiency of the iterative soft decision decoding algorithm, particularly the PTA, for the class of binary cyclic codes with coding performance close to the ML decoding. Chapter 1 gives the general introduction to the study and addresses the challenges faced by iterative SD decoding of cyclic codes. Also, Chapter 1 highlights the main hypotheses used to investigate and answer the research question of the thesis. Moreover, the relevance of this study and its applications are emphasized in Chapter 1.

A detailed background of fundamental principles and related literature is presented in Chapter 2. In this chapter, the linear block codes are described in terms of the generator and parity-check matrices. Also, the chapter addresses the relationship between the minimum distance of the code and the parity-check matrix. Besides, extensive related work on the soft decision decoding algorithms of cyclic codes is discussed in Chapter 2 to support the aim and achievement of this thesis.

In Chapter 3, An iterative soft decision decoding algorithm is developed for decoding binary cyclic codes based on the relaxed parity-check transformation. The modified parity-check transformation algorithm is an extension of the symbol-level PTA, which avoids the parity-check matrix inversion method of the PTA. The results in Chapter 3 show the importance of carefully selecting the updating factor since it reduces the number of decoding iterations of the EPTA. Other results in Chapter 3 indicate that the EPTA exhibits an enhanced decoding performance in comparison to the conventional HD decoder and other iterative soft decision decoding algorithms. Moreover, the time complexity analysis of the algorithm is presented using the big-$\mathcal{O}$ notation to allow a fair comparison of the message passing stage between the algorithm and the well-known ABP algorithm.

A generalized parity-check matrix transformation algorithm for iterative soft decision decoding of binary cyclic codes is developed in Chapter 4. The GPT algorithm permutes the columns of the parity-check matrix based on the reliability information from the channel's output. The results in Chapter 4 shows that the developed GPT exhibits better performance compared to the algebraic hard decision decoding algorithm, the conventional BP algorithm, and other soft decision decoders. In Chapter 4, the worst-case time complexity analysis of the GPT, PTA, and ABP is performed using the big-$\mathcal{O}$ notation, showing that the ABP has the highest complexity. Besides, the GPT goes through a reduced number of iteration as compared to other SD decoding algorithms. This portrays the GPT as an efficient SD decoder for the class of cyclic codes that can be implemented in real-time applications.

In Chapter 5, a baseline parity-check matrix for iterative soft-decision decoding of binary cyclic codes is developed based on the perfect knowledge model PKM. The model computes all the possible parity-check matrices according to the channel condition and selects the best matrix based on minimum distance criteria as detailed in Chapter 5. Also, a numerical example is given in Chapter 5 showing that the matrices obtained from the proposed PKM are optimal. Results show that selecting an optimal parity-check matrix enhances decoding performance of the ABP and proposed GPT algorithms compared to the maximum likelihood decoder.

This thesis basically provides efficient decoding algorithms for binary cyclic codes in terms of error rate performance and computational complexity. While binary cyclic codes are used as the testbeds in this research, most finding in this study can be extended to linear block codes. The results provided in Chapters 3–5 answer the research question in this thesis. The following major points are generally inferred from the findings of this study:

- The PTA developed for symbol-level decoding in [15] fails for the class of non-MDS codes whenever the parity-check matrix of the code is not invertible.
- The PTA decoder can be extended to a bit-level decoding algorithm for non-MDS codes if the condition for matrix transformation is relaxed during the decoding process.
- The parity-check matrix transformation method of the PTA and EPTA decoders can be formed to a generalized parity-check matrix transformation algorithm based on permuting columns of the matrix according to the bit reliabilities. Thus, enhancing the performance and reducing the complexity of the PTA compared to the ABP decoding algorithm in [14].
- The perfect knowledge model can be used as a baseline instead of the ML decoding algorithm to evaluate the performance of iterative SD decoders for binary cyclic codes based on parity-check equations PCEs. The model selects the best parity-check matrix, which is optimal and does not necessarily contain unit weighted columns in the corresponding columns of unreliable bits. A major point shown by the PKM is that the transformed parity-check matrices of the ABP, EPTA, and GPT algorithms are suboptimal.

- The performance of the iterative soft decision decoding algorithms can be enhanced depending on a carefully selected updating factor, a well-refined bit reliability, and the transformed parity-check matrix employed.

6.2 Recommendation and Future Research Possibilities

The PKM involves large computations as there is an exponential increase in the number of parity-check matrices. Also, because it has previous understanding of the transmitted codewords, it is not suitable for practical decoding. Therefore, the following recommendations can be regarded as useful insights for future research potentials to make full use of the study and findings provided in this thesis.

1. Research can be carried out to construct of the optimal parity-check matrix without creating a list of possible matrices.
2. It is also essential to have a practicable model that chooses the baseline parity-check matrix without knowing the codeword that has been transmitted.

Appendix A

Simulation Model

In order to test the different algorithms in Chapters 3–5 simulation was performed based on MATLAB scripts, which operates on arrays and matrices. One of the problems encountered with emperical work is to ensure that the end results are credible and realistic. This Section provides the important parameters needed to develop a realistic simulation environment and properly test the developed algorithms to prevent any false positive outcomes.

For the proposed algorithms in this thesis, the following outline can be used to simulate an (n, k) binary cyclic code having rate $R = k/n$.

1. Find all the $n - k$ degree cyclic polynomials p. MATLAB Communications Toolbox, *cyclpoly*.

2. Obtain the systematic parity-check matrix **H** and generator matrix **G** from any of the generating polynomial. MATLAB Communications Toolbox - *cyclgen*.

3. Create a modulator system object (bpskModulator) that modulates the input signal using the BPSK method and creates the ideal signal constellation. MATLAB Communications Toolbox - *comm.BPSKModulator*.

4. Create a demodulator system object (bpskDemodulator) that demodulates the input signal using the binary phase shift keying (BPSK) method. MATLAB Communications Toolbox - *comm.BPSKDemodulator*.

5. Obtain a binary message of length k by generating integer values from the uniform distribution on the set $0 : 1$. MATLAB Communications Toolbox - *randi*.

6. Encode the binary message to a codeword of length is n using the cyclic code. The operation requires vector/matrix multiplication over $GF(2)$, that is modulo 2 operations. MATLAB Communications Toolbox - *encode*.

7. Modulate encoded n bit codewords using the BPSK modulation scheme where 0 is mapped to $+1$ and 1 is mapped to -1.

8. Pass the modulated signal through an AWGN channel to obtain a soft received information $\underline{r}$. MATLAB Communications Toolbox - *awgn*. The SNR is in dB. The signal power is measured before adding noise.

9. Compute the reliability matrix β based on the Euclidean distance d between the signal constellation and the received vector from the channel's output as in Equations (2.47)–(2.50)

10. Decode using the BM, BP, PTA, ML, RRD, PBP, ABP, EPTA, GPT, and PKM algorithms.

11. Compare the BER performance of all the simulated algorithms.

12. Compare the time taken for each iterative algorithm to run (iteration).

13. Compare the number of iterations required for each iterative decoders to run.

Algorithm 3: Simulation environment for (n, k) binary cyclic code

```
Require: $p$, $\mathbf{H}$, $\mathbf{G}$, signal constellation
  Initilize: $iteration$, Fix $E_b = 1$
  foreach $SNR$ do
      for $runtime = 1 : iteration$ do
          Compute $N_0 = E_b/SNR$ and $\sigma^2 = N_0/2$
          Obtain binary message of length $k$
          Encode message to codeword of length $n$
          Modulate encoded $n$ bit codewords using BPSK modulation to obtain signals $\underline{s}$
          Generate a vector $\phi$ of statistically independent Gaussian random variables with zero mean and variance $\sigma^2 = N_0/2$
          Pass the modulated signal over the AWGN channel to obtain $\underline{r} = \underline{s} + \phi$
          Compute $\beta$ from $\underline{r}$ as in Equations (2.47)–(2.50)
          Decode using EPTA as in Algorithm 1
          Decode using GPT as in Algorithm 2
          Decode using baseline parity-check matrix from PKM
      end
  end
```

Bibliography

[1] D. J. Lin, Shu; Costello. *Error Control Coding: Fundamentals and Applications*. Saddle River, NJ 07458: Pearson Education Inc., Pearson Prentice Hall, 2004.

[2] C. E. Shannon. "A Mathematical Theory of Communication." *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[3] M. Tuchler, R. Koetter, and A. Singer. "Turbo equalization: principles and new results." *IEEE Transactions on Communications*, vol. 50, no. 5, pp. 754–767, May 2002.

[4] T. Clevorn, J. Brauers, M. Adrat, and P. Vary. "Turbo decodulation: iterative combined demodulation and source-channel decoding." *IEEE Communications Letters*, vol. 9, no. 9, pp. 820–822, Sep. 2005.

[5] T. K. Moon. *Error Correction Coding: Mathematical Methods and Algorithms*. John Wiley & Sons, Inc., 2005.

[6] R. Gallager. "Low-density parity-check codes." *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[7] R. McEliece, D. MacKay, and J.-F. Cheng. "Turbo decoding as an instance of Pearl's "belief propagation" algorithm." *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 140–152, 1998.

[8] M. Baldi and F. Chiaraluce. "A Simple Scheme for Belief Propagation Decoding of BCH and *RS* Codes in Multimedia Transmissions." *International Journal of Digital Multimedia Broadcasting*, vol. 2008, pp. 1–12, 2008.

[9] S. B. Wicker and V. K. Bhargava. *Algorithms and Architectures for the Design of a VLSI Reed-Solomon Codec*. IEEE, 1994.

[10] J. Yedidia. “Representing codes for belief propagation decoding.” In *IEEE International Symposium on Information Theory, 2003. Proceedings.*. IEEE, Jun. 2003.

[11] S. Sankaranarayanan and B. Vasic. “Iterative Decoding of Linear Block Codes: A Parity-Check Orthogonalization Approach.” *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3347–3353, Sep. 2005.

[12] J. Jiang and K. Narayanan. “Iterative soft decoding of Reed-Solomon codes.” *IEEE Communications Letters*, vol. 8, no. 4, pp. 244–246, Apr. 2004.

[13] J. Jiang and K. Narayanan. “Iterative soft decision decoding of Reed Solomon codes based on adaptive parity check matrices.” In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, p. 261. Chicago, IL, USA: IEEE, Jun. 2004.

[14] J. Jiang and K. Narayanan. “Iterative Soft-Input Soft-Output Decoding of Reed–Solomon Codes by Adapting the Parity-Check Matrix.” *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.

[15] O. Ogundile, D. Versfeld, and Y. Genga. “Symbol level iterative soft decision decoder for Reed-Solomon codes based on parity-check equations.” *Electronics Letters*, vol. 51, no. 17, pp. 1332–1333, Aug. 2015.

[16] R. Koetter and A. Vardy. “Algebraic soft-decision decoding of reed-solomon codes.” *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[17] E. R. Berlekamp. *Algebraic Coding Theory: Revised Edition*. Aegean Park Press,, 1984.

[18] T. Richardson and R. Urbanke. “The capacity of low-density parity-check codes under message-passing decoding.” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[19] "ETSI EN 302 307 V1.4.1 Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications.", Mar. 2014.

[20] "3GPP2 CMDA2000 High Rate Broadcast-Multicast Packet Data Air Interface Specification.", Feb. 2006.

[21] F. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes*. North-Hollard publishing company, 1981.

[22] J. Meggitt. "Error correcting codes and their implementation for data transmission systems." *IEEE Transactions on Information Theory*, vol. 7, no. 4, pp. 234–244, oct 1961.

[23] W. Peterson. "Encoding and error-correction procedures for the Bose-Chaudhuri codes." *IEEE Transactions on Information Theory*, vol. 6, no. 4, pp. 459–470, Sep. 1960.

[24] R. Chien. "Cyclic decoding procedures for Bose- Chaudhuri-Hocquenghem codes." *IEEE Transactions on Information Theory*, vol. 10, no. 4, pp. 357–363, Oct. 1964.

[25] G. Forney. "On decoding BCH codes." *IEEE Transactions on Information Theory*, vol. 11, no. 4, pp. 549–557, Oct. 1965.

[26] E. Berlekamp. "On decoding binary Bose-Chadhuri- Hocquenghem codes." *IEEE Transactions on Information Theory*, vol. 11, no. 4, pp. 577–579, Oct. 1965.

[27] J. L. Massey. "Shift-register synthesis and BCH decoding." *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.

[28] H. Burton. "Inversionless decoding of binary BCH codes." *IEEE Transactions on Information Theory*, vol. 17, no. 4, pp. 464–466, Jul. 1971.

[29] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. "A method for solving key equation for decoding goppa codes." *Information and Control*, vol. 27, no. 1, pp. 87–99, Jan. 1975.

[30] C. Hartmann and L. Rudolph. "An optimum symbol-by-symbol decoding rule for linear codes." *IEEE Transactions on Information Theory*, vol. 22, no. 5, pp. 514–517, Sep. 1976.

[31] A. Ashikhmin and S. Litsyn. "Simple MAP Decoding of First-Order Reed–Muller and Hamming Codes." *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1812–1818, Aug. 2004.

[32] A. Thangaraj. "Simple MAP Decoding of Binary Cyclic Codes." In *2006 IEEE International Symposium on Information Theory*, pp. 464–468. Seattle, USA: IEEE, Jul. 2006.

[33] G. Forney. "Generalized minimum distance decoding." *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, Apr. 1966.

[34] D. Chase. "Class of algorithms for decoding block codes with channel measurement information." *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, Jan. 1972.

[35] H. Tang, Y. Liu, M. Fossorier, and S. Lin. "On combining Chase-2 and GMD decoding algorithms for nonbinary block codes." *IEEE Communications Letters*, vol. 5, no. 5, pp. 209–211, May 2001.

[36] H. Tanaka and K. Kakigahara. "Simplified correlation decoding by selecting possible codewords using erasure information (Corresp.)." *IEEE Transactions on Information Theory*, vol. 29, no. 5, pp. 743–748, Sep. 1983.

[37] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa. "An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder." *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 320–327, Mar. 1994.

[38] M. Fossorier and S. Lin. “Soft-decision decoding of linear block codes based on ordered statistics.” *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[39] M. Sudan. “Decoding of Reed Solomon Codes beyond the Error-Correction Bound.” *Journal of Complexity*, vol. 13, no. 1, pp. 180–193, Mar. 1997.

[40] V. Guruswami and M. Sudan. “Improved decoding of Reed-Solomon and algebraic-geometric codes.” In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, vol. 45, pp. 1757–1767. IEEE Comput. Soc, 1999.

[41] Y. Wu. “New List Decoding Algorithms for Reed–Solomon and BCH Codes.” *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3611–3630, Aug. 2008.

[42] G. Schmidt, V. R. Sidorenko, and M. Bossert. “Syndrome Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis.” *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5245–5252, Oct. 2010.

[43] L. Chen and M. Bossert. “Algebraic chase decoding of Reed-Solomon codes using module minimisation.” In *2016 International Symposium on Information Theory and Its Applications (ISITA)*, pp. 305–309. Oct. 2016.

[44] R. G. Gallager. *Low-Density Parity-Check Codes*. M.I.T. Press, 1963.

[45] S. J. Johnson. *Introducting Low-Density Parity-Check Codes*. University of Newcastle, Australia, 2006.

[46] R. Tanner. “A recursive approach to low complexity codes.” *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[47] D. MacKay. “Good error-correcting codes based on very sparse matrices.” *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[48] A. Kothiyal and O. Takeshita. “A comparison of adaptive belief propagation and the best graph algorithm for the decoding of linear block codes.” In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*. IEEE, 2005.

[49] J. Hagenauer, E. Offer, and L. Papke. “Iterative decoding of binary block and convolutional codes.” *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[50] A. Ahmed, R. Koetter, and N. Shanbhag. “Performance analysis of the adaptive parity check matrix based soft-decision decoding algorithm.” In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*. IEEE.

[51] B. Kamali and A. Aghvami. “Belief Propagation Decoding of Reed-Solomon Codes; a Bit-Level Soft Decision Decoding Algorithm.” *IEEE Transactions on Broadcasting*, vol. 51, no. 1, pp. 106–113, Mar. 2005.

[52] A. Vardy and Y. Be’ery. “Bit-level soft-decision decoding of Reed-Solomon codes.” *IEEE Transactions on Communications*, vol. 39, no. 3, pp. 440–444, Mar. 1991.

[53] A. Kothiyal, O. Takeshita, W. Jin, and M. Fossorier. “Iterative reliability-based decoding of linear block codes with adaptive belief propagation.” *IEEE Communications Letters*, vol. 9, no. 12, pp. 1067–1069, Dec. 2005.

[54] M. El-Khamy and R. J. McEliece. “Iterative Algebraic Soft Decision Decoding of Reed-Solomon Codes.” In *International Symposium on Information Theory and its Applications*, pp. 1456–1461. Oct. 2004.

[55] S. Laendner, T. Hehn, O. Milenkovic, and J. B. Huber. “CTH02-4: When Does One Redundant Parity-Check Equation Matter?” In *IEEE Globecom 2006*. IEEE, Nov. 2006.

[56] T. Halford and K. Chugg. “Transactions Letters - Random Redundant Iterative Soft-in Soft-out Decoding.” *IEEE Transactions on Communications*, vol. 56, no. 4, pp. 513–517, Apr. 2008.

[57] J. Leon. "Computing automorphism groups of error-correcting codes." *IEEE Transactions on Information Theory*, vol. 28, no. 3, pp. 496–511, May 1982.

[58] N. Sendrier and G. Skersys. "On the computation of the automorphism group of a linear code." In *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252)*. IEEE, Jun. 2001.

[59] T. Hehn, J. B. Huber, S. Laendner, and O. Milenkovic. "Multiple-Bases Belief-Propagation for Decoding of Short Block Codes." In *2007 IEEE International Symposium on Information Theory*. IEEE, Jun. 2007.

[60] T. Hehn, J. Huber, O. Milenkovic, and S. Laendner. "Multiple-bases belief-propagation decoding of high-density cyclic codes." *IEEE Transactions on Communications*, vol. 58, no. 1, pp. 1–8, Jan. 2010.

[61] I. Dimnik and Y. Be'ery. "Improved random redundant iterative HDPC decoding." *IEEE Transactions on Communications*, vol. 57, no. 7, pp. 1982–1985, Jul. 2009.

[62] M. Ismail, S. Denic, and J. Coon. "Efficient Decoding of Short Length Linear Cyclic Codes." *IEEE Communications Letters*, vol. 19, no. 4, pp. 505–508, Apr. 2015.

[63] F. Celler, C. R. Leedham-Green, S. H. Murray, A. C. Niemeyer, and E. O'brien. "Generating random elements of a finite group." *Communications in Algebra*, vol. 23, no. 13, pp. 4931–4948, Jan. 1995.

[64] E. Alpaydin. *Introduction to Machine Learning: Adaptive Computation and Machine Learning*. 212. The MIT Press, Cambridge, 2010.

[65] Farebrother. *Linear Least Squares Computations (Statistics: A Series of Textbooks and Monographs)*. CRC Press, 1988.

[66] N. Noorshams and M. J. Wainwright. "Stochastic Belief Propagation: A Low-Complexity Alternative to the Sum-Product Algorithm." *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 1981–2000, Apr. 2013.

[67] M. Baldi, G. Cancellieri, and F. Chiaraluce. "Iterative Soft-Decision Decoding of Binary Cyclic Codes." *Journal of Communications Software and Systems*, vol. 4, no. 2, p. 142, Jun. 2008.