Ph.D. Thesis
Doctor of Philosophy

Error Reconciliation Protocols for Continuous-Variable Quantum Key Distribution

Department of Physics
Technical University of Denmark

Hossein Mani

Supervisor: Professor Ulrik Lund Andersen
Co-supervisor: Associate professor Tobias Gehring
Co-supervisor: Senior researcher Christoph Pacher

Kongens Lyngby 2020

DTU Physics
Department of Physics
Technical University of Denmark

Fysikvej,
Building 311
2800 Kongens Lyngby, Denmark
Tel.: +45 4525 3344
info@fysik.dtu.dk
https://www.fysik.dtu.dk/english

Summary

Continuous-variable quantum key distribution (CV-QKD) utilizes an ensemble of coherent states of light to distribute secret encryption keys between two parties. One of the key challenges is the requirement of capacity-approaching error correcting codes in the low signal-to-noise (SNR) regime (SNR < 0 dB). Multi-level coding (MLC) combined with multi-stage decoding (MSD) can solve this challenge in combination with multi-edge-type low-density parity-check (MET-LDPC) codes which are ideal for low code rates in the low SNR regime due to degree-one variable nodes. However, designing such highly efficient codes remains an open issue. Here, we introduce the concept of generalized extrinsic-information transfer (G-EXIT) charts for MET-LDPC codes and demonstrate how this tool can be used to analyze their convergence behavior. We calculate the capacity for each level in the MLC-MSD scheme and use G-EXIT charts to exemplary find codes for some given rates which provide a better decoding threshold compared to previously reported codes. In comparison to the traditional density evolution method, G-EXIT charts offer a simple and fast asymptotic analysis tool for MET-LDPC codes.

A linear optimization approach to design highly efficient MET-LDPC codes at very low SNR, which is highly required by certain applications like CV-QKD will be discussed. The cascade structure is introduced in terms of three disjoint submatrices and a convex optimization problem is proposed to design highly efficient MET-LDPC codes based on cascade structure. Simulation results show that the proposed algorithm is able to design MET-LDPC codes with efficiency higher than 95%, especially at very low SNR.

Abstrakt

Kontinuert variable kvante-nøgledistribution (CV-QKD) bruger et ensemble af kohærent lysttilstande til at distribuere hemmelige krypteringsnøgler mellem to parter. Én af udfordringerne er kravet om fejlkorrektionskoder der nærmere sig Shannon-kapaciteten i detlave signal-til-støj (SNR) regime (SNR $<$ 0 dB). Flerniveaus-kodning (MLC) kombineret med flerstadie-afkodning (MSD) kan løse denne udfordring i kombination med flerkants-type lavdensitetsparitetscheck (MET-LDPC) koder, som er ideel for lave koderater i det lave SNR-regime pga. grad-1 variable knuder. Imidlertid er design af sådanne effektive koder stadig etåbent problem. Her introducerer vi konceptet af diagrammer for generaliseret overførelse af udefrakommende information (G-EXIT) for MET-LDPC-koder og demonstrerer, hvordan dette værktøj kan bruges til at analysere deres konvergensadfærd. Vi beregner kapaciteten for hvert niveau i MLC-MSD-ordningen og bruger G-EXIT-diagrammer i eksempler på at finde koder til givet rater, der giver en bedre afkodningstærskel sammenlignet med tidligere rapporterede koder. Sammenlignet med den traditionelle tæthedsudviklingsmetode (density evolution method)tilbyder G-EXIT-diagrammer et simpelt og hurtigt asymptotisk analyseværktøj til MET-LDPC-koder.

En lineær optimeringsmetode til at designe meget effektive MET-LDPC-koder ved meget lav SNR, hvilket kræves af visse applikationer som CV-QKD, bliver diskuteret. Kaskadestrukturenintroduceres i form af tre usammenhængende undermatricer, og der foreslås et konveks optimeringsproblem til at designe effektive MET-LDPC-koder baseret på kaskadestruktur. Simuleringsresultater viser, at den foreslåede algoritme er i stand til at designe MET-LDPC-koder med effektivitet højere end 95%, især ved meget lav SNR.

Preface

This PhD thesis was prepared at the department of Physics at the Technical University of Denmark in fulfillment of the requirements for acquiring a PhD degree in Physics.

Kongens Lyngby, August 31, 2020

Hossein Mani

Hossein Mani

Acknowledgements

This Ph.D thesis is the output of the effort and support of several people to whome I am deeply greatful. At first, I thank my supervisors professor **Ulrik Lund Andersen**, Associate professor **Tobias Gehring**, and associate professor **Christoph Pacher**. It has been a great advantage to work with you all. I could not have imagined having better advisors and mentors for my Ph.D study.

Ulrik, thanks for all your support and responsiveness that brought me to the Denmark. Your advices were of great help.
Tobias, I am sincerely greatful for your presence and attention during these years. Many thanks for your supports both technically and personally.
Christoph, thanks for facilitating my transition to Vienna and for your technical supports. I learned a lot from you.

Besides my advisors, I would like to thank the rest of my thesis committee: Professor Søren Forchhammer, Professor Vicente Martin, and Associate Professor. Jesús Martínez Mateo, for their insightful comments and encouragement.

I would like to thank all the people (fellow Ph.D students and postdoctoral researchers, professors, administration, guest researchers, ...) in both the Quantum Physics and information Technology (QPIT) center and the Austrian institute of Technology (AIT) who created a great work environment (Mikkel, Casper, Ulrich, Alexander, Jonas, Tine, Sepehr, Ilya, ...). Especially, I thank my fellow labmates both in DTU and AIT for the helpful discussions, for the great environment we were working together, and for all the fun we have had in these years: Dr. Nitin Jain, Dr. Hou-Man Chin, Dr. Dino Solar Nikolic, Dr. Arne Kordts, Dr. Fabian Laudenbach, Mr. Philipp Grabenweger, ...

I am greatly indebted to my ceremonial assistant **Mahtab Madani** who made the years my Ph.D run safe and sound.

Mahtab, my love, thank you very much for being by my side for 5 years. Could not have done it without you.

Also I thank my friends in the University of Idaho and Tarbiat Modares University. In particular, I am grateful to Prof. Saied Hemati and Associate professor Hamid Saeedi for all the supports.

Fortunately, I have also the privilege of having a lovely family who had a fundamental role in getting me through the PhD process successfully. Thank you all.

Thanks
谢谢
سپاس
Thanks
谢谢
धन्यवाद
سپاس
Gracias
धन्यवाद
धन्यवाद
Thanks
سپاس
Tak
谢谢
Tak
Tak
Thank you
ThanksVielen Dank

Acronyms

In this thesis we also use standard terminology and acronyms. Here we present list of acronyms.

ADC Analog-to-Digital Converter

AES Advanced Encryption Standard

AWGN Additive White Gaussian Noise

BEC Binary Erasure Channel

BER Bit Error Rate

BF Bit-Flipping

BI-AWGN Binary-Input Additive White Gaussian Noise

BIOSM Binary-Input Output-Symmetric Memoryless

BP Belief Propagation

BPSK Binary Phase Shift Keying

BSC Binary Symmetric Channel

CN Check-Node

CSI Channel Side Information

CV Continuous Variable

CV-QKD Continuous-Variable Quantum Key Distribution

DAC Digital-to-Analog Converter

DE Density Evolution

DES Data Encryption Standard

DV-QKD Discrete-Variable Quantum Key Distribution

EB Entanglement Based

EXIT Extrinsic-Information Transfer

FER Frame Error Rate

FR Forward Reconciliation

G-EXIT Generalized Extrinsic-Information Transfer

i.i.d Independent and identically distributed

KL Kullback-Leibler

LDPC Low-Density Parity-Check

LLR Log-Likelihood Ratio

LSB Least-Significant Bit

MET Multi-Edge-Type

MET-LDPC Multi-Edge-Type Low-Density Parity-Check

MLC Multi-Level-Coding

MLC-MSD Multi-Level-Coding Multi-Stage-Decoding

MP Message Passing

MS Min-Sum

MSB Most-Significant Bit

MSD Multi-Stage-Decoding

PC Polarization Controller

PDF Probability Density Function

P&M Prepare and Measure

QC-LDPC Quaci-Cyclic LDPC

QKD Quantum Key Distribution

QRNG Quantum Random Number Generator

RR Reverse Reconciliation

RSA Rivest Shamir Adleman

SCA Stochastic Chase Algorithm

SCE Socket Count Equality

SNR Signal to Noise Ratio

SNU Shot-Noise Unit

TMSVS Two-Mode-Squeezed Vacuum State

VN Variable-Node

Contents

Summary i

Abstrakt iii

Preface v

Acknowledgements vii

Acronyms ix

Contents xiii

List of Figures xv

List of Tables xvii

1 Introduction 1

1.1 Cryptography: History, Present, Future 1
1.2 Important challenges in post-processing of QKD 2
1.3 Thesis outline 4
1.4 Academic publications 5

2 Quantum Key Distribution with Continuous Variables 7

2.1 Introduction 7
2.2 CV-QKD with Gaussian modulation 8
2.3 Security Analysis and Secure Key Rate 14
2.4 Experimental setup 19

3 Information Reconciliation of CV-QKD 21

3.1 Multidimensional Reconciliation 22
3.2 Multi-Level-Coding Multi-Stage-Decoding 25
3.3 Comparison between reverse and forward reconciliation 43
3.4 Randomized Reconciliation 48
3.5 Reconciliation for a wide range of SNRs 51

4 On the Design of Highly Efficient MET-LDPC codes 55

4.1 Multi-Edge-Type LDPC codes 55
4.2 Density evolution and other asymptotic analysis tools 60
4.3 G-EXIT charts for MET-LDPC codes 66
4.4 Code design optimization problem 74
4.5 Optimization for cascade structure 78
4.6 Highly efficient codes 83
4.7 Simulation Results 87

5 Conclusion and Future Directions 91

A Introduction to LDPC codes 95
A.1 Preliminaries 95
A.2 Channel entropy and channel capacity 97
A.3 Low-density parity-check (LDPC) codes 98
A.4 Belief Propagation 100
A.5 EXIT and G-EXIT charts 101
A.6 Plotting EXIT and G-EXIT Charts 102

B Reconciliation software for CV-QKD 105
B.1 About the Software 105
B.2 How to install the software 107

C Software to design highly efficient MET-LDPC codes 109
C.1 About the software 109

Bibliography 111

List of Figures

2.1 Simple illustration of QKD system 10
2.2 Phase-space representation 12
2.3 Ideal homodyne detection 13
2.4 Effect of reconciliation efficiency-Asymptotic 17
2.5 Effect of reconciliation efficiency-Finite-size 18
2.6 Experimental setup 19

3.1 Multidimensional reconciliation 24
3.2 Reconciliation: Multi-level coding 27
3.3 Binary set partitioning 27
3.4 Digitizer intervals and levels 29
3.5 Digitization effect 30
3.6 Reconciliation: One-level coding 31
3.7 Reconciliation: Two-level coding 36
3.8 Channel coding rates for individual levels 39
3.9 Mutual information for individual levels 40
3.10 Numerical joint PMF at SNR $= -5$ dB 41
3.11 Forward vs Reverse reconciliation 43
3.12 Admissible rate region 45
3.13 Differential entropies 45
3.14 Decoding by measured soft information 48
3.15 Reconciliation efficiencies for SNR range 52
3.16 Shortening process 53

4.1 Graphical representation of MET-LDPC code 58
4.2 KL divergence for Gaussian approximation 63
4.3 KL divergence for edge type two 64
4.4 KL divergence for edge type three 64
4.5 EXIT chart for rate 0.5 68
4.6 G-EXIT chart for MET-LDPC with rate 0.02 71
4.7 Convergence behavior on a BEC 72
4.8 G-EXIT chart for MET-LDPC with rate 0.02 on a BI-AWGN channel . . 75
4.9 EXIT chart for base code with additional noise 81
4.10 Optimization problem for the connector part 83

4.11 Optimization algorithm 84
4.12 The FER Comparison for rate 0.02 88
4.13 The efficiency Comparison for rate 0.02 89

A.1 The schematic representation of standard BIOSM channels 96
A.2 The capacity of the BEC and BSC channels 99
A.3 The capacity of the BI-AWGN and AWGN channels 99

C.1 EXIT chart for rate 0.8 irregular LDPC code 110

List of Tables

2.1 CV-QKD vs DV-QKD . . . 8

3.1 Digitization effect . . . 30
3.2 Decoders complexity . . . 51

4.1 Table format for MET-LDPC codes . . . 56
4.2 Rate 0.02 MET-LDPC code . . . 57
4.3 MET-LDPC code with cascade structure . . . 59
4.4 Complexity comparison for different algorithms . . . 65
4.5 G-EXIT curves on BE channel . . . 70
4.6 EXIT curves on BIOSM channel . . . 73
4.7 G-EXIT curves on BIOSM channel . . . 73
4.8 Table presentation for MET-LDPC code with rate 0.02 . . . 77
4.9 MET-LDPC code with Rate 0.01 . . . 84
4.10 MET-LDPC code with Rate 0.02 . . . 85
4.11 MET-LDPC code with Rate 0.05 . . . 85
4.12 MET-LDPC code with Rate 0.10 . . . 86
4.13 MET-LDPC code with Rate 0.25 . . . 86

5.1 Summary of designed codes . . . 92

A.1 The distribution of the L conditioned on the $X = 1$ . . . 97
A.2 The parametric representation of an EXIT chart . . . 103
A.3 The parametric representation of G-EXIT chart . . . 103

CHAPTER 1

Introduction

1.1 Cryptography: History, Present, Future

Historically, *cryptography* was used mainly for military purposes. The famous Caesar cipher or *scytale* of Sparta in ancient Greece are examples of historical cryptography systems. For *encryption* the Caesar cipher substitutes each letter in the message (*plain-text*) by another letter shifted by some fixed number (*key*). *Decryption* is the reverse operation which is applied to the encrypted message (*cipher-text*) to recover the plain-text. The number of fixed shifts for the Caesar cipher was allegedly 3. It is clear that the number of possible keys for this cipher is 26. Thus, one unauthorized person can test all the possible keys in a *brute-force* attack to recover the plain-text. In addition, some efficient attacks also exist based on frequency analysis of the cipher-text. The academic research in the field of cryptography was started in late 1970s. Today, the cryptography is an integral part of everyday life. It is impossible to imagine an internet based activity without cryptography. It can be a simple Web browsing, sending email or a money transfer in a bank system.

The science of *cryptography* and the science of *cryptanalysis* are considered with each other as the filed of *cryptology*. Classical cryptography after the computer age consists of two main categories, *symmetric* and *asymmetric* cryptography.

In symmetric cryptography two legitimate parties perform encryption and decryption with a shared secret key. The *Data Encryption Standard* (DES) is an example of symmetric cryptography, created by IBM in early 1970s. In 1977 with modifications from the United States National Security Agency (NSA) it was used as a government standard suitable for commercial applications. The private key of DES is 56-bit which makes it vulnerable for brute-force attack. In 1997 the United States' National Institute of Standards (NIST) organized an open competition and in 2000 they announced the Advanced Encryption Standard (AES) as a block cipher with three different key sizes 128, 192 and 256. Regardless of encryption standard the procedure to share the secret key plays an important role in the security of the system. One practical method to share the secret key is the use of smart cards which is not very promising and cost efficient solution in large scale networks. Another solution is the use of asymmetric or *public key cryptography*. Public key cryptography uses two different keys for encryption and decryption. The public key is used for encryption and the private key remains secret for decryption. In public key cryptography the encryption procedure corresponds to a public key and should be a simple task. The decryption task should be easy for everyone who has the private key, but should be very complicated for someone who does not have the private key. In principle, asymmetric cryptography algorithms like Rivest Shamir Adleman (RSA), Elgamal and ECDSA are based on mathematical problems (integer factorization, discrete logarithm and Elliptic curves). It implies that the security of these algorithms depends on the eavesdropper computational power.

In practice, if two parties want to send each other a long message, they use a hybrid encryption system. First they use an asymmetric cipher to share key between each other and then use a symmetric cipher to transmit the actual data. It is due to the fact that for large amounts of data the symmetric ciphers are faster than the asymmetric ciphers.

The security of today's asymmetric cryptography, e.g. the RSA protocol and the Diffie-Hellman key-exchange protocol, is based on mathematical complexity assumptions of basic problems like the discrete log problem and the factorization of large numbers [1]. These classical algorithms provide *computational security.* The advent of the quantum computer or an unexpected algorithmic innovation can compromise their security with drastic consequences for the internet.

One possible solution is *quantum key distribution* (QKD) [2, 3] which provides *information theoretical secure* cryptographic key exchange for two parties, Alice and Bob, based on the properties of quantum mechanics (no cloning theorem, superposition, entanglement and nonlocality). The idea of using quantum physics for cryptography was first introduced in the 1983. The idea was taken from the fact that in a quantum system cannot be measured without perturbing the system. Thus Alice and Bob can share a key based on transmission and measurement of quantum states and Eve cannot extract any information about the communication without introducing perturbations. Thus, thanks to the fundamental principles of quantum physics the QKD makes us able to detect illegitimate parties and provide a secure communication link.

1.2 Important challenges in post-processing of QKD

Several commercial QKD implementations have already exists worldwide. Apart from different variations of QKD systems which are based on *discrete-variable* (DV) or *continuous-variable* (CV) the post-processing of QKD protocols still remained challenging for long distance. Recent results show that for *continuous-variable quantum key distribution* (CV-QKD) using optical fibers the longest possible distance is 202.81 km with output key rate of 6.24 bits/s [4]. For *discrete-variable quantum key distribution* (DV-QKD) the maximum reported distance is 307 km with output key rate of 3.18 bits/s [5]. The motivation of this thesis is to address two of the key challenges that exists in post-processing of QKD system by focusing on CV-QKD:

1. High throughput reconciliation for CV-QKD: In the case of CV-QKD two different approaches can be used to extract binary information from Gaussian

variables. Currently the best known reconciliation method for CV-QKD uses a *multi-dimensional reconciliation* [6]. Though this method denotes high efficiency specifically at very low SNRs it is limited to extract just one bit from the Gaussian symbols. In principle for the CV-QKD in contrast to DV-QKD it is possible to extract more than one bit from Gaussian symbols. One of the best known methods to extract more than one bits from Gaussian symbols is *multi-level-coding multi-stage-decoding* (MLC-MSD) [7]. The MLC-MSD method has been used for reconciliation but particular advantage of MLC-MSD reconciliations remained unclear due to lack of study [8, 9].

- In this thesis we provide a detailed analysis of the MLC-MSD scheme and denote that it is possible to extract two-bits from Gaussian symbols. Specifically, we calculate the soft information for two levels which then can be used by soft decoders.
- We will show that the MLC-MSD scheme requires multiple encoders and decoders each working at specific rate. In addition we provide both analytical and numerical methods to calculate the code rates for each level. Details are provided in Section 3.2.5.
- In this thesis also we introduce the concept of randomized reconciliation which can be used to increase the throughput of the reconciliation task by sacrificing the frame error rate performance. The idea of randomized reconciliation is to use a fast hard-decoder instead of complicated soft-decoder and feed different error patterns to the hard-decoder. The error patterns should be generated randomly using the soft information that we extracted from the channel. More details about randomized reconciliation is provided in Section 3.4.

2. Long distance CV-QKD: Current implementations of CV-QKD shows that for efficient reconciliation an error correction task is required at very low signal-to-noise ratio (SNR). For instance in [10] an SNR of $-15.37\,\mathrm{dB}$ was reported for a transmission distance of $80\,\mathrm{km}$ and in [11] an SNR of $-16.198\,\mathrm{dB}$ for $100\,\mathrm{km}$. Designing highly efficient forward error correction codes (FECs) at such a low SNR is one of the core problems. Though *multi-edge-type low-density parity-check* (MET-LDPC) codes have been widely applied to QKD for error correction the characteristics of these codes and their design procedure are not widely understood. Few researchers have addressed the problem of designing highly efficient degree-distribution (DD) for MET-LDPC codes but unfortunately their works are limited to high SNR regime (SNR > 0) and require high computational complexity for their optimization algorithm [12, 13, 14]. To the best of our knowledge the only available DD for a rate 0.02 MET-LDPC code was presented in [10] and has an asymptotic efficiency of 98%.

    - In this thesis we analyse the charactristics of the MET-LDPC codes and focuse on MET-LDPC codes with cascade structure. More details about the cascade strucures and MET-LDPC codes can be found in Section 4.1.

- We propose a new approximation method for density evoution (DE) of MET-LDPC codes called semi-Gaussian approximation.
- In addition the concept of *extrinsic-infromation transfer chart* (EXIT) and generalized-EXIT (G-EXIT) chart are introduced for MET-LDPC codes. Then we will show how these tools can be used to analyse the performance of the MET-LDPC codes.
- Furthermore, in this thesis we provide an algorithmic approach to design highly efficient DD for MET-LDPC codes for low SNR.
- Using our proposed algorithm we design some new highly efficient DDs for MET-LDPC codes. For example we designed a new DD for rate 0.02 which has an asymptotic efficiency of 99.2%. Also we designed a new code for rate 0.01 which can works at SNR −18.48 dB.

.

1.3 Thesis outline

This thesis is divided into five chapters. Chapter 2, gives a brief overview of the CV-QKD. Some important steps in CV-QKD protocols are explained. Also we talk about the secure key rate and some of the important factors for long distance CV-QKD. Finally our the experimental setup is presented.

Chapter 3, focuses on the reconciliation process of CV-QKD. First, a brief overview of multi-dimensional reconciliation is represented. Secondly, we focus on reconciliation based on MLC-MSD and a detailed analysis is presented. We calculate the soft information for deocders at each level and denote how to find the optimum code rate in MLC-MSD scheme. Then, we propose a new reconciliation scheme called Randomized Reconciliation to increase the throughput of the reconciliation scheme. Finally we talk about existing tools and methods to provide a robust reconciliation for a wide range of SNRs.

In Chapter 4, a detailed analysis of MET-LDPC codes is presented. First, it starts by defining this class of error correction codes. Secondly, we talk about DE and other asymptotic analysis tools for MET-LDPC codes. We also propose our semi-Gaussian approximation method. Thirdly, we introduce the concept of generalized extrinsic-information transfer (G-EXIT) chart and explain how we can develop this concept for MET-LDPC codes. In addition we explain how the convergence behavior of MET-LDPC codes can be described by G-EXIT charts. Fourthly, we propose our new algorithmic optimization process to design highly efficient MET-LDPC codes. Then we present for the first time some new DDs for MET-LDPC codes designed by our new algorithm. These codes are specifically designed for low rate applications like CV-QKD. Finally we show the performance of the rate 0.02 code.

In Chapter 5 we conclude the thesis. In addition this thesis contains three appendices. In Appendix A, we explain requirements related to the LDPC codes. In addition the definition of EXIT and G-EXIT functions for irregulr LDPC codes are presented. In Appendix B, useful information can be found about our software tools related to the reconciliation of the CV-QKD. Appendix C explains how to use the software tools to design new degree distributions for MET-LDPC codes.

1.4 Academic publications

The results of this thesis have been presented in posters and presentations at the conferences [15, 16, 17] and also some of the results are submitted to peer-reviewed journals where the arXived version are also available [18]. Here is the list of our publications:

1. H. Mani, T. Gehring, C. Pacher, and U. L. Andersen, "*Multi-edge-type LDPC code design with G-EXIT charts for continuous-variable quantum key distribution*,"ArXiv, vol. abs/1812.05867, 2018. [Online]. Available: `https://arxiv.org/pdf/1812.05867.pdf`

2. H. Mani, T. Gehring, C. Pacher, and U. L. Andersen, "*An approximation method for analysis and design of multi-edge type LDPC codes*," 2018. [Online]. Available: `http://2018.qcrypt.net/others/accepted-posters/`

3. H. Mani, T. Gehring, C. Pacher, and U. L. Andersen, "*Algorithmic approach to design highly efficient MET-LDPC codes with cascade structure*," 2019. [Online]. Available: `http://2019.qcrypt.net/scientific-program/posters/`

4. H. Mani, B. Ömer, U. L. Andersen, T. Gehring, and C. Pacher, "*Two MET-LDPC codes designed for long distance CV-QKD*," 2020.[Online]. Available: `https://2020.qcrypt.net/accepted-papers/#list-of-accepted-posters`

CHAPTER 2 Quantum Key Distribution with Continuous Variables

This chapter is about the principles of the QKD. After a short introduction, Section 2.2 talks about the basic steps of CV-QKD. In addition important steps in QKD protocols will be discussed. In Section 2.3, we will talk about the security analysis and different attacks in CV-QKD. Finally, the experimental setup of our CV-QKD is presented in Section 2.4.

2.1 Introduction

Similar to classical information, quantum information uses two types of variables: discrete-variables and continuous-variables. One example of a discrete quantum variable is two polarization states of a single photon and the best-known example of continuous quantum information [19, 20] is the quantized harmonic oscillator described by continuous variables such as position and momentum. In this thesis, we tried to focus on the quantum information processing specifically for QKD using continuous variables. It means that instead of working with the properties of the single photons, the quadratures of the electromagnetic field can be used as continuous variables.

The first DV-QKD protocol was investigated by Bennet and Brassard in 1984, known as BB84 [21]. The main limitation of DV-QKD is the requirement of single-photon counting technology. In a major advance, almost fifteen years later, Grosshans and Grangier proposed the first and simplest CV-QKD protocol known as GG02 [22]. Nowadays, CV-QKD is attracting considerable interest due to its simpler practical implementation, thanks to the existence of many electro-optical components developed for optical telecommunication. For instance, the alternative of the single-photon counting technology for CV-QKD is coherent detection techniques, which are widely used in classical optical communications. Though, CV-QKD provides a convenient practical implementation, its security proofs are still not mature in comparison with DV-QKD protocol. Two important factors to evaluate practical performance of QKD systems are secure key rate and transmission distance. Table 2.1 briefly compares the recent results related to these two families of the protocols and some of their basic differences.

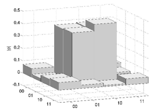
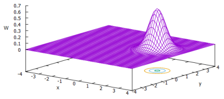
	DV-QKD	CV-QKD
Year	2019	2019
Light	Discrete Photon	Continuous Wave
Information carrier	Photon polarization/phase	Field phase or amplitude
State representation	Density matrix 	Wigner function 
Detector	single-photon detector	Homodyne/Heterodyne
Practical maximum range	104 km (307 km)	202.81 km
Output key rate	12.7 kbps (3.18 bps)	6.24 bps

Table 2.1: Respective comparison of DV-QKD and CV-QKD protocols. The results are taken from [5, 4].

This chapter aims to provide a short review of CV-QKD protocols focusing in particular on protocols using Gaussian modulation (GM) of coherent states. Gaussian states are continuous variable states that have a representation in terms of Gaussian functions. A very comprehensive collection of literature about the Gaussian quantum information plus topics related to CV-QKD can be found in [20, 22, 23, 24, 25, 26, 27, 28].

2.2 CV-QKD with Gaussian modulation

There are different types of CV-QKD protocols where the simplest class are one-way protocols with Gaussian modulation. Other variants like two-way protocols [29, 30] or non-Gaussian modulation [31] fall outside the scope of this thesis. In general, the state-of-the-art in GM CV-QKD has two possible implementations: *prepare and measure* (P&M) and *entanglement based* (EB). In the P&M based protocol, Bob measures the quadrature components of the displaced coherent states where generated by Alice and transmitted through a Gaussian channel. In the case of EB protocol, Alice generates a two-mode squeezed state in her lab, performs the measurement on one mode, and sends the other mode to Bob. It has been proven [26] that for Gaussian protocols, as long as Alice's lab is trusted, these two implementations are equivalent. Thus, providing a security proof for the EB protocol is equivalent to the security proof of the P&M protocol which has simpler implementation.

A CV-QKD protocol contains multiple steps including:

- State preparation and measurement
- Information reconciliation
- Parameter estimation
- Privacy amplification.

These are the basic steps of any QKD protocol and different implementation choices for these steps provide a variety of different protocols. For example, using single-mode or two-mode states for state preparation, using homodyne or heterodyne detection, forward or reverse reconciliation would cause different implementations and protocols. There are different reasons for choosing different protocols, for instance, some of the protocols have simpler implementations, some of those are better for long distance transmission and some of those have better security proofs. In [28] a comprehensive comparison is done for variety of CV-QKD protocols including the final status of their security proofs.

Additionally, it is important to notice that the CV-QKD protocol consists of two different phases. The first phase is related to the transmission of the quantum states in a quantum channel and the second phase is related to the classical post-processing tasks. Thus, it is convenient to consider two types of channels including quantum channel and classical channel for these two phases.

Finally, the goal of the whole system is to generate a secure shared key between two distant parties using transmission of quantum states on an untrusted quantum channel and later by exchanging classical data through an authenticated classical channel as presented in Figure 2.1. The following subsections briefly describe each steps of a CV-QKD protocol.

2.2.1 State preparation and measurement

2.2.1.1 Quadrature operators

The definition of the electric field using the *annihilation* and *creation* operators ($\hat{a},\ \hat{a}^\dagger$) can be written for a single mode as:

$$\vec{E}(\vec{r},t) = E_0\ \vec{e}\ [\hat{q}\ \cos(\vec{k}\cdot\vec{r}-\omega\ t)+\hat{p}\ \sin(\vec{k}\cdot\vec{r}-\omega\ t)]\ , \tag{2.1}$$

where ω is the angular frequency, $\vec{e}$ is the polarization vector and operators $\hat{q}$ and $\hat{p}$ are called *quadratures* of the electromagnetic field

$$\hat{q} = \hat{a}^\dagger + \hat{a}\ , \tag{2.2}$$

$$\hat{p} = i(\hat{a}^\dagger - \hat{a})\ . \tag{2.3}$$

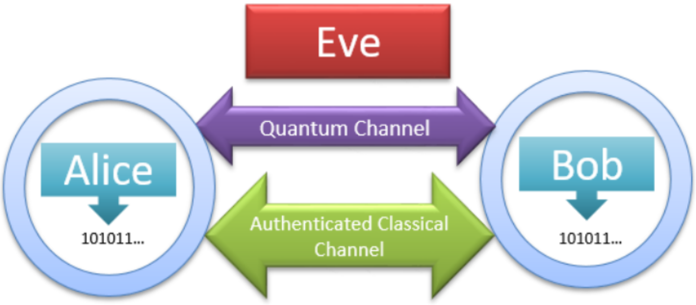


Figure 2.1: Simple illustration of the QKD system with two channels. The quantum channel and the classical channel.

These dimensionless operators are then measurable and satisfy the *commutation* relation

$$[\hat{q},\ \hat{p}] = 2i\ , \tag{2.4}$$

which then gives the Heisenberg uncertainty relation

$$\delta\hat{q}\ \delta\hat{p} \geq 1\ . \tag{2.5}$$

Finally, the photon number operator is defined as:

$$\hat{n} = \frac{1}{4}(\hat{q}^2 + \hat{p}^2) - \frac{1}{2}\ , \tag{2.6}$$

and all the above formulations are in shot-noise units (SNU).

2.2.1.2 Gaussian modulation

Let us consider a Gaussian modulated scenario. In this case, Alice prepares a sequence of coherent states $|\alpha_1\rangle\,,\ \cdots,\ |\alpha_j\rangle\,,\ \cdots, |\alpha_N\rangle$ where

$$|\alpha_j\rangle = |q_j + ip_j\rangle\,. \tag{2.7}$$

Then, the two quadrature components q and p can be considered as real valued outcomes of two independent and identically distributed (i.i.d) normal random variables $\mathcal{Q}$ and $\mathcal{P}$ with zero mean and variance $\frac{V_{\text{mod}}}{4}$,

$$\mathcal{Q} \sim \mathcal{P} \sim \mathcal{N}(0,\ \frac{V_{\text{mod}}}{4}). \tag{2.8}$$

Since, coherent states $|\alpha_j\rangle$ are eigenstates of the annihilation operator, thus

$$\hat{a}\,|\alpha_j\rangle = \alpha_j\,|\alpha_j\rangle \ , \tag{2.9}$$

$$\frac{1}{2}(\hat{q}+\hat{p})\,|\alpha_j\rangle = (q_j + ip_j)\,|\alpha_j\rangle \, . \tag{2.10}$$

Furthermore, the corresponding variance for the quadrature operators are equal and related to the modulation variance by

$$\mathrm{Var}(\hat{q}) = \mathrm{Var}(\hat{p}) = V = V_{\mathrm{mod}} + 1 \ . \tag{2.11}$$

It is clear that even with $V_{\mathrm{mod}} = 0$, the modulation variance of quadrature operators equal to $V_0 = 1$, which is referred to as shot-noise. In addition, the corresponding phase-space representation of a coherent state $|\alpha_j\rangle$ can be visualized by using the Wigner function, which serves as a quasi-probability distribution in phase space. The marginal probability distribution of a quadrature measurement can be obtained from the Wigner function by integration over the other conjugate quadrature as presented in Figure 2.2. More details about the continuous variables and phase-space representation can be found in [19, 20, 23, 24].

2.2.1.3 The covariance matrix

After Alice's preparation of coherent states $|\alpha_j\rangle$, she sends them through a quantum channel to Bob. Then, Bob uses a homodyne or heterodyne detection to measure the eigenvalues of one or both quadrature operators (P&M protocol). It has been shown that for Gaussian modulation, this is equivalent to the case when Alice generates a two-mode-squeezed vacuum state (TMSVS), performs a measurement on one mode, and sends the other mode to the Bob (EB protocols). The *covariance matrix* of this two mode bosonic system can be described as:

$$\Sigma_{\mathrm{A,B}} = \begin{pmatrix} V & 0 & \sqrt{V^2-1} & 0 \\ 0 & V & 0 & -\sqrt{V^2-1} \\ \sqrt{V^2-1} & 0 & V & 0 \\ 0 & -\sqrt{V^2-1} & 0 & V \end{pmatrix} , \tag{2.12}$$

where $V = V_{\mathrm{mod}} + 1$ denotes the variance of the quadrature operators. It can be written in a more compact form by considering the Pauli matrix $\sigma_Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$

$$\Sigma_{\mathrm{A,B}} = \begin{pmatrix} V\,\mathbb{I}_2 & \sqrt{V^2-1}\,\sigma_Z \\ \sqrt{V^2-1}\,\sigma_Z & V\,\mathbb{I}_2 \end{pmatrix} , \tag{2.13}$$

where $\mathbb{I}_2$ is the identity matrix with dimension 2. In terms of the V_{mod} the covariance matrix is:

$$\Sigma_{\mathrm{A,B}} = \begin{pmatrix} (V_{\mathrm{mod}}+1)\,\mathbb{I}_2 & \sqrt{V_{\mathrm{mod}}^2 + 2V_{\mathrm{mod}}}\,\sigma_Z \\ \sqrt{V_{\mathrm{mod}}^2 + 2V_{\mathrm{mod}}}\,\sigma_Z & (V_{\mathrm{mod}}+1)\,\mathbb{I}_2 \end{pmatrix} . \tag{2.14}$$

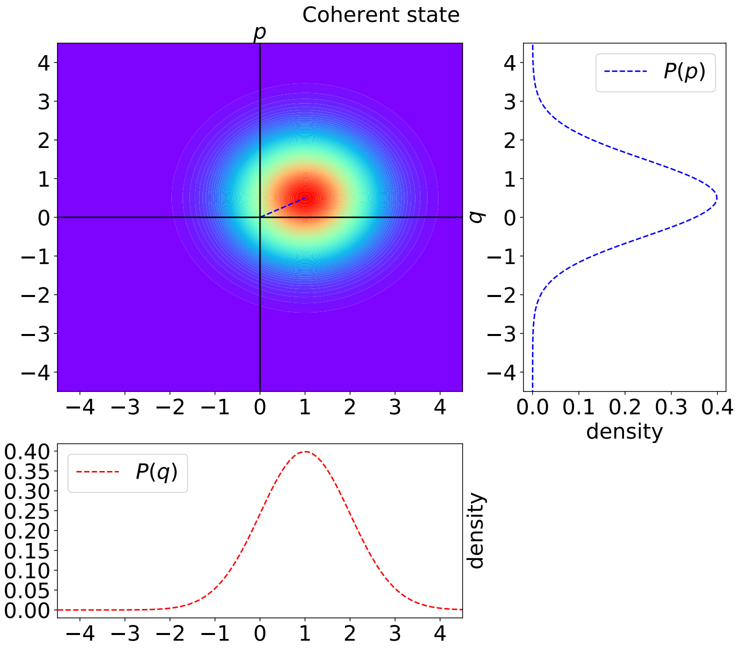


Figure 2.2: Phase-space representation of a coherent state $|\alpha_j\rangle$. For any coherent state, the quadratures have the same variance.

Finally, according to [32] the corresponding covariance matrix after transmission through a Gaussian channel and Bob's homodyne detection is:

$$\Sigma_{\mathrm{A,B}} = \begin{pmatrix} (V_{\mathrm{mod}} + 1)\ \mathbb{I}_2 & \sqrt{T(V_{\mathrm{mod}}^2 + 2V_{\mathrm{mod}})}\ \sigma_Z \\ \sqrt{T(V_{\mathrm{mod}}^2 + 2V_{\mathrm{mod}})}\ \sigma_Z & (TV_{\mathrm{mod}} + 1 + \xi)\ \mathbb{I}_2 \end{pmatrix} . \tag{2.15}$$

where T stands for the transmittance and ξ denotes the total excess noise which is the combination of the excess noise, electric noise etc. A comprehensive calculation of the covariance matrix and the relation between the P&M based protocol and EB protocol can be found in [32].

2.2.1.4 Homodyne detection

As shown in Figure 2.3, one of the quadratures of an electromagnetic field can be measured using an ideal balanced homodyne detector. It contains a local oscillator and a balanced beamsplitter. The input mode is combined with the local oscillator and then the intensity of the outgoing modes are measured using two photodiodes. Let us denote the quadratures of the local oscillator by $(x_{\mathrm{LO}}, 0)$ where for each mode

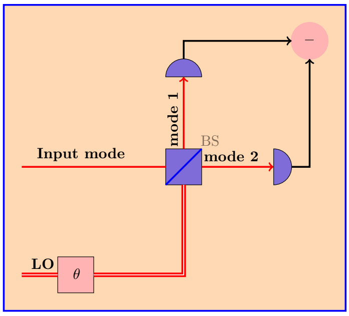


Figure 2.3: Simple block diagram of ideal homodyne detector.

$$I_i = \frac{k}{2}(\hat{q}_i^2 + \hat{p}_i^2 - 1) \ , \quad \text{for} \quad i = 1, 2 \ , \tag{2.16}$$

where k is a prefactor contains all the dimensional, and

$$\hat{q}_1 = \frac{\hat{q}_{\mathrm{in}} + x_{\mathrm{LO}}}{\sqrt{2}} \ , \tag{2.17}$$

$$\hat{p}_1 = \frac{\hat{p}_{\mathrm{in}}}{\sqrt{2}} \ , \tag{2.18}$$

$$\hat{q}_2 = \frac{\hat{q}_{\mathrm{in}} - x_{\mathrm{LO}}}{\sqrt{2}} \ , \tag{2.19}$$

$$\hat{p}_2 = \frac{\hat{p}_{\mathrm{in}}}{\sqrt{2}} \ . \tag{2.20}$$

Then, to obtain an estimation of the $\hat{q}$, one can write

$$I_1 - I_2 = \frac{k}{2}\left((\hat{q}_{\mathrm{in}} + x_{\mathrm{LO}})^2 - (\hat{q}_{\mathrm{in}} - x_{\mathrm{LO}})^2\right) = k \ \hat{q}_{\mathrm{in}} x_{\mathrm{LO}} \ . \tag{2.21}$$

To measure the other quadrature, it is enough to apply a phase shift of $\frac{\pi}{2}$ to the local oscillator and then, follow the same procedure. The realistic implementation of the homodyne detection is not addressed here. Detailed and relevant information for homodyne and heterodyne detection can be found in [23, 32].

2.2.2 Post-processing

This section gives a brief overview of how Alice and Bob can generate a secure key from their raw data. There are three main tasks during the post-processing, but the order of these steps might be different according to the protocol.

2.2.2.1 Information reconciliation

After state preparation and measurement, it can be assumed that Alice and Bob have access to their list of real-valued data. If Bob uses heterodyne detection, then he has a list of $n = 2N$ real valued data denoted by $\vec{X}_B = (X_{B,i})_{i=0}^{n-1}$ and Alice has access to her own list of data denoted by $\vec{X}_A = (X_{A,i})_{i=0}^{n-1}$. If Bob uses homodyne detection, then $n = N$ and he informs Alice about the choice of his quadratures and Alice then ignores half of her data accordingly. If Alice and Bob agree to remove all uncorrelated data before starting post-processing, an extra step known as *sifting* is necessary. For the case of heterodyne detection no sifting is required.

In Chapter 3, a detailed description of different reconciliation schemes are presented. Two possible options for the reconciliation are *forward reconciliation* (direct reconciliation) and *reverse reconciliation.* In forward reconciliation (FR), Bob corrects his data according to Alice's data and in reverse reconciliation (RR), Alice tries to correct her data to estimate the Bob's data. In general, RR provides better performance in comparison to FR [25].

2.2.2.2 Parameter estimation

The goal of this step is to obtain an upper bound on Eve's information. This can be done by accurate estimation of the covariance matrix. Considering RR, the upper bound for Eve's possible information about the Bob's key is denoted by $\mathcal{X}_{\mathrm{EB}}$. A more detailed description about the Holevo information and the security of the CV-QKD are presented in Section 2.3. In general, in the case of ideal post processing, it is possible to extract a secure key, if the mutual information between Alice and Bob $I(A; B)$ (sometimes shown as I_{AB}) is higher than the Holevo bound between Eve and Bob, which is known as Devetak-Winter formula [33].

2.2.2.3 Privacy amplification

Finally, Alice and Bob have access to a same bit string which they can use as a secret key, but it might be possible that Eve also has some correlation with this data. Thus, Alice and Bob extract a shorter string from their common string by applying a hashing function.

2.3 Security Analysis and Secure Key Rate

There are three kind of attacks in the CV-QKD:

- Individual attack,
- Collective attack,
- Coherent attack.

In all of these attacksit is assumed that Eve has full access to the quantum channel, but in the case of classical channel she is not able to manipulate the classical channel. In another words, it is assumed that the protocol has an *authenticated* classical channel to exchange some classical information between Alice and Bob. All the information transferred in this classical channel is considered available to Eve.

In the case of individual attack, Eve performs her measurement right after Bob reveals his quadrature before doing reconciliation. In this case, Eve's information is limited by the Shannon information and it has been shown in [34] that the raw Shannon key rate is:

$$K^{\text{Raw}}_{\text{ind, RR}} = I_{\text{AB}} - I_{\text{BE}} \ , \tag{2.22}$$

which is secure for Gaussian and non-Gaussian individual attacks, even with finite size length for reverse reconciliation. I_{AB} is the mutual information between Alice and Bob, and I_{BE} is the classical mutual information between Eve and Bob's data.

For collective attack, it has been shown that the asymptotic secret key rate for reverse reconciliation is:

$$K^{\text{Asymp}}_{\text{coll, RR}} \geq (1 - \text{FER})(1 - \nu)(\beta I_{\text{AB}} - \mathcal{X}_{\text{EB}}) \ , \tag{2.23}$$

where $\mathcal{X}_{\text{EB}}$ is upper bound for the Eve's information on the Bob's data, FER $\in [0,\ 1]$ is the frame error rate of the reconciliation, $\beta\ \in [0,\ 1]$ denotes the efficiency of the reconciliation process, and ν denotes the fraction of the data used for the estimation of the covariance matrix [35, 36, 37].

If $f_{X_B}(x_B)$, denotes the probability distribution of the Bob's measured outcomes, then the value of the Holevo quantity $\mathcal{X}_{\text{EB}}$ can be calculated as

$$\mathcal{X}_{\text{EB}} = \mathcal{S}(\rho_{\text{E}}) - \int f_{X_B}(x_B)\mathcal{S}(\rho_{\text{E}}^{x_B})\ dx_B \ , \tag{2.24}$$

where $\mathcal{S}(\rho)$ denotes the von Neumann entropy of the quantum state ρ. For the n-mode Gaussian state ρ, the von Neumann entropy is

$$\mathcal{S}(\rho) = \sum_{i}^{n} G\left(\frac{\lambda_i - 1}{2}\right) \ , \tag{2.25}$$

where $G(x) = (x+1)\log_2(x+1) - x\log_2(x)$, and λ_i denote the eigenvalues of the covariance matrix related to the ρ. A more detailed calculation of asymptotic secure key rate for collective attack can be found in [23, 35, 36]. To read more about von Neumann entropy and smooth min and max entropy see [38].

Here, we quickly provide a short calculation of the secure key rate for a reverse reconciliation CV-QKD with homodyne detection. All the parameters are in SNU, specifically ξ_{ch} denotes the excess channel noise, v_{el} denotes the electronic noise and η denotes the efficiency of the homodyne detection. Assuming a single-mode fiber with transmission loss $\alpha = 0.2$ dB/km, the transmittance of such channel is $T_{\text{ch}} = 10^{-\alpha d/10}$, where d denotes the distance between the two parties. As presented in [35], total noise between Alice and Bob is:

$$\xi_{\text{total}} = \xi_{\text{line}} + \frac{\xi_{\text{hom}}}{T_{\text{ch}}} \ ,$$

where $\xi_{\text{hom}} = \dfrac{1+v_{el}}{\eta} - 1$ is the homodyne detector noise and $\xi_{\text{line}} = (\dfrac{1}{T_{\text{ch}}} - 1) + \xi_{\text{ch}}$. Then, according to [35] and as presented in (2.15), the variance of Bob's data after homodyne detection, in SNU is:

$$\begin{aligned} V_B = &\ \eta T_{\text{ch}}(V + \xi_{\text{total}}) = \eta T_{\text{ch}} \left(V + \xi_{\text{line}} + \frac{\xi_{\text{hom}}}{T_{\text{ch}}} \right) \\ = &\ \eta T_{\text{ch}} \left(V + (\frac{1}{T_{\text{ch}}} - 1) + \xi_{\text{ch}} + \frac{\dfrac{1+v_{el}}{\eta} - 1}{T_{\text{ch}}} \right) \\ = &\ \eta T_{\text{ch}} V - \eta T_{\text{ch}} + \eta T_{\text{ch}} \xi_{\text{ch}} + 1 + v_{el} \\ = &\ T(V-1) + T\xi_{\text{ch}} + 1 + v_{el} \\ = &\ T V_{\text{mod}} + 1 + \xi \ , \end{aligned}$$

where $T = \eta T_{\text{ch}}$, $V = V_{\text{mod}} + 1$ and $\xi = T_{\text{ch}}\xi_{\text{ch}} + v_{el}$ as shown in (2.15). In addition, the mutual information between Alice and Bob is:

$$I_{\text{AB}} = \frac{1}{2}\log_2(1 + \text{SNR}) = \frac{1}{2}\log_2\left(1 + \frac{V + \xi_{\text{tot}}}{1 + \xi_{\text{tot}}}\right) \ . \tag{2.26}$$

According to [35], Eve's classical mutual information from Bob's data is:

$$I_{\text{BE}} = \frac{1}{2}\log_2 \frac{V_B}{V_{B|E}} \ , \tag{2.27}$$

where $V_{B|E} = \eta \left[\dfrac{1}{T(1/V) + \xi_{\text{line}}} + \xi_{\text{hom}} \right]$. Finally, Holevo bound $\mathcal{X}_{\text{EB}}$ is calculated as:

$$\mathcal{X}_{\text{EB}} = G\left(\frac{\lambda_1 - 1}{2}\right) + G\left(\frac{\lambda_2 - 1}{2}\right) - G\left(\frac{\lambda_3 - 1}{2}\right) - G\left(\frac{\lambda_4 - 1}{2}\right) \ , \tag{2.28}$$

where the eigenvalues are calculated from:

$$\begin{aligned} \lambda_{1,2} = &\ \sqrt{\frac{1}{2}\left(A \pm \sqrt{A^2 - 4B}\right)} \ , \\ \lambda_{3,4} = &\ \sqrt{\frac{1}{2}\left(C \pm \sqrt{C^2 - 4D}\right)} \ , \end{aligned}$$

where

$$\begin{aligned}
A =&\ V^2(1-2T) + 2T + T^2(V+\xi_{\text{line}})^2,\\
B =&\ T^2(V\xi_{\text{line}}+1)^2,\\
C =&\ \frac{V\sqrt{B} + T(V+\xi_{\text{line}}) + A\ \xi_{\text{hom}}}{T(V+\xi_{\text{tot}})},\\
D =&\ \sqrt{B}\ \frac{V+\sqrt{B}\xi_{\text{hom}}}{T(V+\xi_{\text{tot}})}.
\end{aligned}$$

From this we can plot the the asymptotic secret key rate for reverse reconciliation. In Figure 2.4, the asymptotic secure key rate is plotted against distance for different reconciliation efficiencies where all other parameters are assumed to be fixed. It can be observed that the overall transmittable distance between the two parties depends significantly on the reconciliation efficiency.

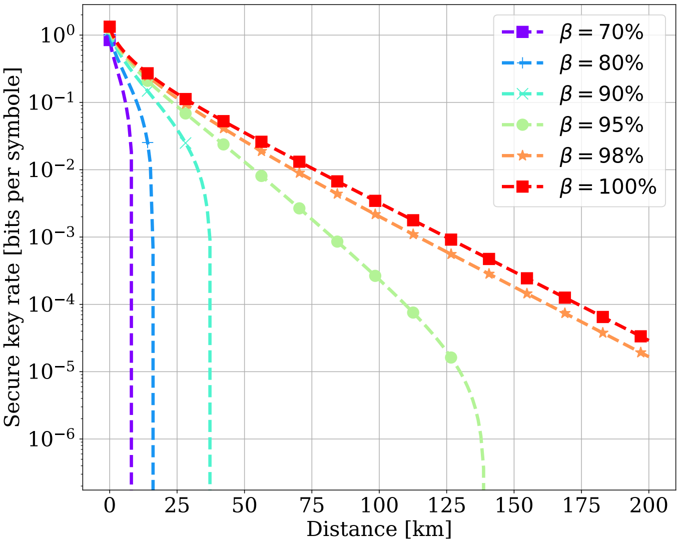


Figure 2.4: Effective asymptotic secure key rates against collective attack. The reconciliation efficiency $\beta = \{0.7, 0.8, 0.9, 0.95, 1.00\}$. The parameters used in the calculations are $V_{\text{mod}} = 8.5$, $\xi_{\text{ch}} = 0.015$, $\eta = 0.6$, $v_{el} = 0.041$, and the fiber loss is assumed to be 0.2 dB/km.

To read more about the security proof of the CV-QKD for finite length and most general coherent attack see [39, 40]. Let us repeat a similar simulation by considering the finite size effects. It is assumed that the length of the data for the privacy amplification is $n_{\text{privacy}} = 10^{10}$ bits. The number of the quantum symbols used for reconciliation is $n_{\text{quantum}} = 2 \times n_{\text{privacy}}$ ($\nu = 0.5$). We assumed that the security parameter $\epsilon_{\text{security}} = 10^{-10}$. From [39], the secure key rate considering the finite-size effect is:

$$K^{\text{Finite}}_{\text{coll, RR}} \geq (1 - \text{FER})(1 - \nu)(\beta I_{\text{AB}} - \mathcal{X}_{\text{EB}} - \Delta(n_{\text{privacy}})) \, , \tag{2.29}$$

where $\Delta(n_{\text{privacy}})$ is the finite-size offset factor and for $n_{\text{privacy}} > 10^4$ can be approximated by:

$$\Delta(n_{\text{privacy}}) \approx 7\sqrt{\frac{\log_2(2/\epsilon_{\text{security}})}{n_{\text{privacy}}}} \, .$$

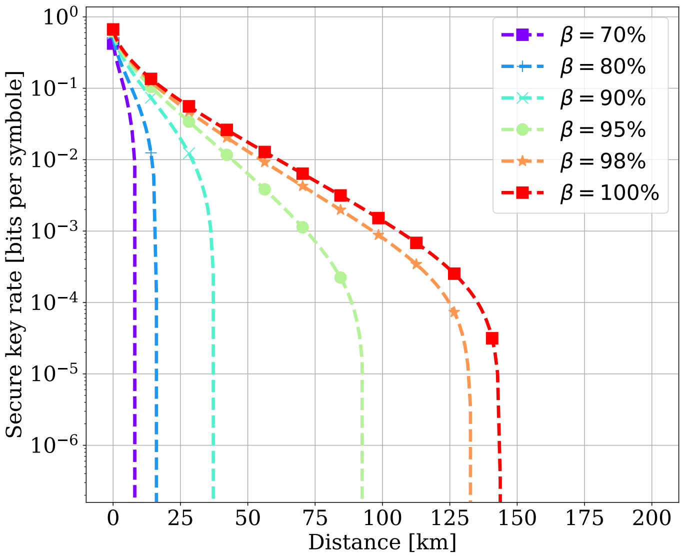


Figure 2.5: Effective finite secure key rate against collective attack. The reconciliation efficiency $\beta = \{0.7, 0.8, 0.9, 0.95, 1.00\}$. The parameters used in the calculations are $V_{\text{mod}} = 8.5$, $\xi_{\text{ch}} = 0.015$, $\eta = 0.6$, $v_{el} = 0.041$, $\nu = 0.5$, $n_{\text{privacy}} = 10^{10}$, $\epsilon_{\text{security}} = 10^{-10}$, and the fiber loss is assumed to be 0.2 dB/km.

Finally, a very comprehensive comparison for the CV-QKD protocols and their best available security proof until 2015 is available in [28]. In this thesis we focus on the reconciliation protocols for the CV-QKD and represent how it is possible to design highly efficient reconciliation scheme with lowest FER. A detailed description of different reconciliation schemes will be discussed in Chapter 3.

2.4 Experimental setup

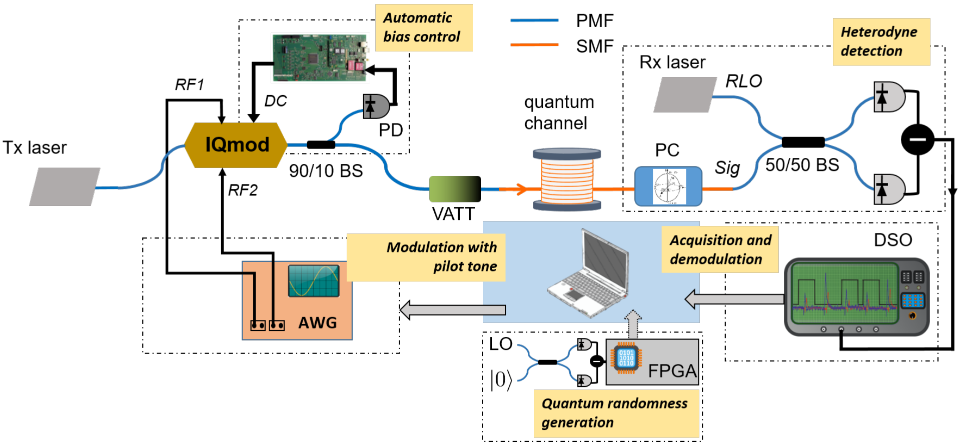


Figure 2.6: The experimental setup of our CV-QKD. The schematic was generated by Dr. Nitin Jain.

The experimental setup for the Gaussian modulated CV-QKD that we have established in our lab is shown in Figure 2.6. It has a continuous-wave laser (Tx laser) operating at 1550 nm and contains standard fiber optics and telecommunication components. On the transmitter side, an in-phase and quadrature (I-Q) electro-optical modulator is driven by Tx laser. It produces coherent states at a single side-band of the optical field at a rate of 100 MSymbols/s. In addition, to generate a complex coherent state amplitudes of the quantum signal, a quantum random number generator (QRNG) with a security parameter $\epsilon_{\text{qrng}} = 10^{-10}$ is used. The QRNG delivered Gaussian distributed symbols for discrete Gaussian modulation of coherent states. After attenuation to the desired mean photon number count, the coherent states were transmitted through a 20 km standard single mode fiber. The receiver side contains a frequency detuned laser (local oscillator) of the same type (Rx laser), a balanced beam splitter and a balanced receiver for radio-frequency heterodyne detection. As shown in Figure 2.6, a polarization controller (PC) is used to manually tune the polarization to match with Rx laser's polarization. Then, the signal and the local oscillator were interfered on a balanced beam splitter and detected by a balanced receiver. In addition, the I-Q-modulator was driven by two digital-to-analog converters (DACs) with 16 bit precision and 1 GSample/s and the receiver's output was sampled by a 16 bit analog-to-digital converter(ADC) at 1 GSample/s. Finally, the data was stored on a hard drive for offline post processing. For more details about our experimental setup, machine learning based approach for carrier phase recovery and the QRNG see [41, 42].

The secret key was established by the following protocol:

1. Gaussian distributed random numbers were generated by the random number generator and stored in a file.
2. The transmitter and receiver were calibrated.
3. Using the 20 km fiber, we transmitted the coherent states using frames prerecorded in a file and played out using the arbitrary waveform generator.
4. After the experiment was conducted, digital-signal-processing was performed offline to obtain the raw key.
5. Information reconciliation was performed using a multi-level coding, multi-stage decoding scheme by using multi-edge-type low-density-parity-check codes. More details about the reconciliation scheme, will be discussed in Chapter 3.
6. After error correction the entropy of the received symbols was estimated as well as the channel parameters for bounding the Holevo information.
7. By using a randomly chosen Toeplitz hash function for privacy amplification the final secret key is generated.

CHAPTER 3

Information Reconciliation of CV-QKD

Information reconciliation is a method by which two parties, each possessing a sequence of numbers, agree on a *common* sequence of bits by exchanging one or more messages. In CV-QKD with Gaussian modulation, the two sequences of numbers are joint instances of a bi-variate random variable that follows a bi-variate normal distribution. Physically, in a prepare-and-measure CV-QKD setup, these sequences are generated by one party modulating coherent states and the other party, measuring these states. The amplitude of each modulated coherent state, which can be visualized by a point in the quadrature-phase space, is determined by the values in the sequence. In other words, in QKD, two parties share correlated random variables and wish to agree on a common bit sequence. However, imperfect correlations introduced by the inherent shot noise of coherent states and noise in the quantum channel and the receiver, give rise to discrepancies in the two sequences of numbers which have to be corrected by exchanging additional information. As discussed in Section 2.3, the efficiency and performance of the error correction codes for reconciliation strongly affects the secure key rate of CV-QKD, which makes it one of the most crucial stages in the protocol (See Figure 2.4 and Figure 2.5).

This chapter is about the problem of reconciliation of Gaussian variables for CV-QKD. Two different reconciliation methods will be discussed. First, in Section 3.1, the *multi-dimensional reconciliation* will be introduced. Then, Section 3.2, talks about the reconciliation based on *multi-level-coding multi-stage-decoding* (MLC-MSD) and different variations of the method will be discussed (See also our arXived article [18]). In Section 3.4, the *randomized reconciliation* scheme, which is a modified version of the MLC-MSD approach for high throughput reconciliation will be introduced. Finally, Section 3.5 talks about reconciliation for a continuous range of SNRs.

3.1 Multidimensional Reconciliation

The idea of multidimensional reconciliation was first introduced in [6]. The authors apply a d-dimensional rotation to the Gaussian data from the physical additive white Gaussian noise (AWGN) channel, to convert the reconciliation problem to an equivalent channel coding problem, on a virtual binary-input AWGN (BI-AWGN) channel. As presented in [6], the quality of this approximation increases with d, but it is not possible to increase d arbitrarily. Practical values are $d \in \{1, 2, 4, 8\}$.

More precisely, by applying a d-dimensional rotation, a nonuniform Gaussian distribution on $\mathbb{R}^d$ transforms into a uniform distribution on the unit sphere $\mathcal{S}^{d-1}$ of $\mathbb{R}^d$. Then, it is possible to apply the same reconciliation scheme as in DV-QKD protocols. In DV-QKD, the reconciliation problem is based on the coset coding. The coset coding method was first introduced for wiretap channels in [43]. In this scheme, Alice and Bob agree to use a linear code $\mathcal{C}$ with parity check matrix H. Then, after the quantum phase, where Alice and Bob have access to two correlated strings $\vec{X}_A$ and $\vec{X}_B$, Alice sends the syndrome $\vec{s} = \vec{X}_A \times H$ to Bob through the authenticated classical channel. Then, Bob can apply decoding to this information using a coset code. The difference in this scheme to a channel coding problem is that Alice's data can generate a non-zero syndrome. In another words, Alice cannot restrict her data to only valid codewords. But as long as she transmits the syndrome $\vec{s}$ to Bob, the reconciliation problem with a coset coder is feasible. Another equivalent solution is that Alice randomly selects a codeword $\vec{u}$, and shares $\vec{r} = \vec{u} \oplus \vec{X}_A$ with Bob, using an authenticated classical channel. Then, Bob does the same action and calculates $\vec{r} \oplus \vec{X}_B$, then, he can apply this data to a decoder to estimate $\vec{u}$.

As presented in [6], binary codes designed for DV-QKD can be converted to a binary spherical code with the following mapping :

$$\mathbb{F}_2^d \to \mathcal{S}^{d-1} \subset \mathbb{R}^d, \qquad (b_1, \cdots, b_d) \to \left(\frac{(-1)^{b_1}}{\sqrt{d}}, \cdots, \frac{(-1)^{b_d}}{\sqrt{d}} \right) . \tag{3.1}$$

The noise analysis of the multi-dimensional reconciliation scheme is important. It shows that the virtual channel obtained by suitable mapping can be approximated by a BI-AWGN channel. Assume that after the quantum phase, Alice and Bob have shared two n-dimensional real valued correlated data $\vec{X}_A$ and $\vec{X}_B$. In addition, let us denote by $\vec{x}$ and $\vec{y}$, two d-uplets corresponding to correlated Gaussian vectors $\vec{X}_A$ and $\vec{X}_B$, respectively. Then, as presented in [6], $\vec{y} = \vec{x} + \vec{z}$ with $\vec{x} \ \sim \mathcal{N}(0,1)^d, \vec{z} \ \sim \mathcal{N}(0, \sigma^2)^d$.

In the case of FR, Alice generates a uniform random vector $\vec{u}$ and sends $\vec{r} = \vec{u}.\vec{x}^{-1}$ to Bob. Then Bob computes $\vec{v} = \vec{r}.\vec{y}$. The analysis of the noise for the virtual channel

$(\vec{w} = \vec{v} - \vec{u})$ shows that:

$$\begin{aligned}\vec{w} &= \vec{v} - \vec{u} \\ &= \vec{r}.\vec{y} - \vec{u} \\ &= \vec{u}.\vec{x}^{-1}(\vec{x} + \vec{z}) - \vec{u} \\ &= \vec{u}\frac{\vec{z}}{\vec{x}} \sim \vec{u}\frac{\vec{z}}{||\vec{x}||} \ ,\end{aligned}$$

where as presented in [6, 10], the last equality holds due to the fact that $\vec{x}$ and $\vec{z}$ are independent. It means that the virtual channel can be considered as a Fading channel with known channel side information (CSI) [44]. The fading coefficient is the norm of $\vec{x}$. When d goes to infinity, the distribution of the norm of $\vec{x}$ becomes closer to a Dirac distribution, and the virtual channel becomes closer to a BI-AWGN channel. The maximum possible value for d is 8, because the division operator does not exists for $d > 8$.

Now, let us consider the case of RR. In this case, $\vec{x} = \vec{y} - \vec{z}$ with $\vec{y} \sim \mathcal{N}(0, 1 + \sigma^2)^d, \vec{z} \sim \mathcal{N}(0, \sigma^2)^d$. It is important to notice that in RR, $\vec{y}$ and $\vec{z}$ are not independent (actually, they are highly correlated). Thus, accurate noise analysis is necessary to obtain the distribution of the virtual channel. More precisely, in RR, Bob sends $\vec{r} = \vec{u}.\vec{y}^{-1}$, and Alice calculates $\vec{v} = \vec{r}.\vec{x}$. Thus, the noise of the virtual channel is

$$\begin{aligned}\vec{w} &= \vec{v} - \vec{u} \\ &= \vec{r}.\vec{x} - \vec{u} \\ &= \vec{u}.\vec{y}^{-1}(\vec{y} - \vec{z}) - \vec{u} \\ &= -\vec{u}\frac{\vec{z}}{\vec{y}} \ ,\end{aligned}$$

where in the last equation $\vec{z}$ and $\vec{y}$ are not independent. It shows that there is still a need for discussion on the precise definition of the virtual channel in the RR case. Despite of the lack of understanding (only in the case of RR), this method has been widely used in CV-QKD, and most studies tend to focus on the high reconciliation efficiency for this algorithm [4, 45, 46]. There is still significant concern over the security of RR.

Finally, let us consider the practical implementation of the multi-dimensional reconciliation. First, Alice and Bob normalize their data to have a uniform distribution on the unit sphere $\mathcal{S}^{n-1}$ of $\mathbb{R}^n$. Then, Bob generates a binary sequence $\vec{u}$ of length k with uniform distribution using a quantum random number generator (QRNG) [47], and uses a linear code to generate a codeword $\vec{c}$ of length n. In the next step, this binary code is mapped to a binary spherical code $\vec{c'}$ using (3.1). Then Bob calculates a mapping function $M(\vec{X'}_B, \vec{c'})$ and sends it back to Alice. The mapping function $M(\vec{X'}_B, \vec{c'})$ should satisfy

$$M(\vec{X'}_B, \vec{c'}).\vec{X'}_B = \vec{c'} \ ,$$

where $\vec{X'}_B$ denotes the normalized data of Bob. Then, Alice can use the mapping function and calculate a new sequence of $\vec{e} = M(\vec{X'}_B, \vec{c'}).\vec{X'}_A$. Alice is allowed to use a decoder designed for an BI-AWGN channel to recover estimation of $\vec{u}$. The following block diagram shows the RR protocol, using the multi-dimensional reconciliation approach.

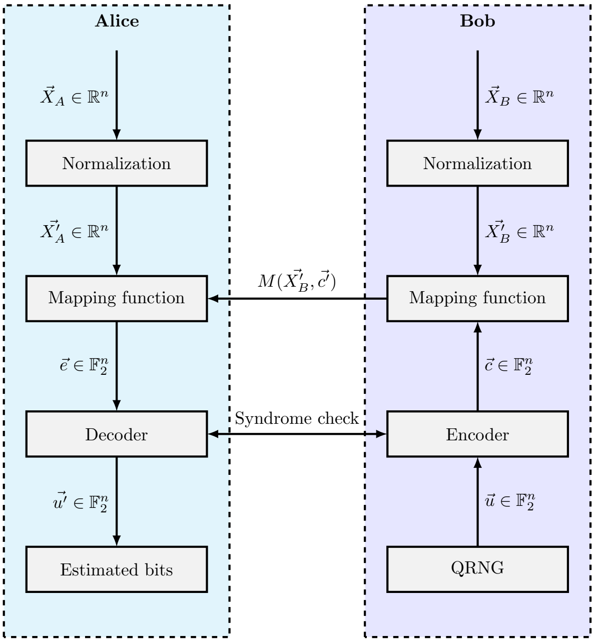

Alice
Bob
$\vec{X}_A \in \mathbb{R}^n$
$\vec{X}_B \in \mathbb{R}^n$
Normalization
Normalization
$\vec{X'_A} \in \mathbb{R}^n$
$\vec{X'_B} \in \mathbb{R}^n$
$M(\vec{X'_B}, \vec{c'})$
Mapping function
Mapping function
$\vec{e} \in \mathbb{F}_2^n$
$\vec{c} \in \mathbb{F}_2^n$
Syndrome check
Decoder
Encoder
$\vec{u'} \in \mathbb{F}_2^n$
$\vec{u} \in \mathbb{F}_2^n$
Estimated bits
QRNG

Figure 3.1: RR using multidimensional reconciliation scheme. $\vec{X'_A}$ and $\vec{X'}_B$ are two correlated sequences of real valued data. $\vec{X'_A}$ and $\vec{X'}_B$ denote the corresponding normalized sequences. $M(\vec{X'}_B, \vec{c'})$ represents the mapping function. The raw key is generated by a QRNG is denoted by $\vec{u}$. The encoded key using a linear code is denoted by $\vec{c}$.

3.2 Multi-Level-Coding Multi-Stage-Decoding

This Section introduces a reconciliation scheme based on MLC-MSD scheme. In the following, we start by representing the block diagram of RR protocol based on MLC-MSD scheme. Then we talk about digitization of continuous sources and try to show the effect of digitization levels. Then we talk with more details about the MLC-MSD scheme for RR. Starting with the *one-level* reconciliation method, which is a specific type when just the *most-significant bit* (MSB) is used for the reconciliation, we define and calculate the soft information for decoders. Then, the extension for *two-level* reconciliation and *multi-level* will also be explained. In addition, we determine how to calculate the code rates for individual levels. We represent both analytical and numerical methods to calculate the individual rates and mutual information. Finally, a comprehensive comparison for FR and RR is presented for the MLC-MSD reconciliation.

It is assumed that readers are familiar with the concept of error correction codes and *low density parity check* (LDPC) codes. Details about the definition of LDPC codes, their parity check matrix and their standard decoder can be found in Appendix A. For a more detailed introduction into these concepts see [44].

Through this thesis we denote a vector with real valued symbols of length n by $\vec{x} = (x_i)_{i=1}^{n} = (x_1, \cdots, x_n)$, where $x_i \in \mathbb{R}$. In addition a m-bit digitization of a real valued symbol at j^{th} index is denoted by $\mathcal{Q}(x_j) = (x_j^i)_{i=0}^{m-1} = (x_j^0, x_j^1, \cdots, x_j^{m-1})$. For example, a vector of Bob's real valued symbols is denoted by $\vec{x_B} = (x_{B,i})_{i=1}^{n} = (x_{B,1}, \cdots, x_{B,n})$ and the quantized version of its j^{th} symbol is $\mathcal{Q}(x_{B,j}) = (x_{B,j}^i)_{i=0}^{m-1} = (x_{B,j}^0, x_{B,j}^1, \cdots, x_{B,j}^{m-1})$. Sometimes, it is more convenient to consider Bob as a continuous Gaussian source denoted by a random variable X_B and its instant value is then denoted by x_B. Then the discretized source is denoted by $\mathcal{Q}(X_B) = (X_B^i)_{i=0}^{m-1}$. In addition $I(X;Y)$ denotes the mutual information between two random variables X and Y, where X and Y could be continuous or discrete random variables. Besides $H(X)$ denotes the entropy of secrete random variable and $h(X)$ denotes the differential entropy for a continuous random variable. Sometimes we are interested to find the entropy or mutual information at specific SNR s, then $I^s(X;Y) = I(X;Y)|_s$, $H^s(X) = H(X)|_s$.

3.2.1 Introduction and system model

In general, multi-level reconciliation using error correction codes can be described in two steps. The first step is digitization, which transforms the continuous Gaussian source X_B, into an m bit source $\mathcal{Q}(X_B)$, with its binary representation vectors $(X_B^i)_{i=0}^{m-1} = (X_B^{m-1}, \ldots, X_B^1, X_B^0)$. There is an inherent information loss due to the digitization process of the source. The second step can be modeled as source coding with side information on the MLC-MSD scheme. In RR Bob sends an encoding (compressed version) of $\mathcal{Q}(\vec{X}_B)$ to Alice, such that she can infer $\mathcal{Q}(\vec{X}_B)$ with high probability, using her own source $\vec{X}_A$ as side information. Let us define the efficiency

β as

$$\beta = \frac{H(\mathcal{Q}(X_B)) - R^{\text{Source}}}{I(X_B;X_A)} , \tag{3.2}$$

where $I(X_B;X_A)$ is the mutual information and $H(\mathcal{Q}(X_B))-R^{\text{Source}}$ is the net shared information between two parties, resp. per symbol, [48, 49] with $H(\cdot)$ being Shannon entropy and R^{Source} the source coding rate. Thus, the efficiency of reconciliation depends on the ability to design very good digitizers and highly efficient compression codes with minimum possible source coding rate (R^{Source}). Slepian and Wolf [50] have shown that $H(Y|Z)$ is the lower bound to the source coding rate when decoding Y given side information Z. Therefore, $R^{\text{s}} \geq H(\mathcal{Q}(X_B)|X_A)$.

A detailed schematic representation for the MLC-MSD scheme is presented in Figure 3.2, where Bob encodes his data onto m different individual levels. Let us denote by R_i^{Source} the corresponding source coding rate for each sub-level i in the MLC-MSD scheme. Then using the Slepian-Wolf theorem, all R_i^{Source} are lower bounded by the conditional entropy of the i^{th} bit of $\mathcal{Q}(X_B)$, given side information X_A and all the remaining *least significant bits* (LSBs) of $\mathcal{Q}(X_B)$:

$$R_i^{\text{Source}} \geq H(X_B^i|X_A, X_B^{i-1}, \dots, X_B^0) . \tag{3.3}$$

The total source coding rate is given by summing over the individual source code rates R_i^{Source}:

$$R^{\text{Source}} = \sum_{i=0}^{m-1} R_i^{\text{Source}},$$

which resembles the Slepian-Wolf theorem:

$$R^{\text{Source}} \geq \sum_{i=0}^{m-1} H(X_B^i|X_A, X_B^{i-1}, \dots, X_B^0) = H(\mathcal{Q}(X_B)|X_A) .$$

The detailed block diagram for the MLC-MSD scheme is depicted in Figure 3.2. We consider a digitization scheme with $M = 2^m$, $m > 1$, signal points in a D-dimensional real signal space, with signal points taken from the signal set $\mathbf{S} = \{a_0, a_1, \dots, a_{M-1}\}$, with probabilities $\Pr\{a_k\}$. Each signal point has its equivalent binary form defined by a bijective mapping $a = \mathcal{M}(\vec{x})$ of binary representation vectors $\vec{x} = (x_B^{m-1}, \dots, x_B^0)$ to signal points $a \in \mathbf{S}$. Two well defined mappings are binary and Gray mapping. As an example, for $m = 3$ levels, in one-dimensional signal space ($D = 1$), the $M = 2^3$ signal points are taken from $\mathbf{S} = \{-7, -5, -3, -1, +1, +3, +5, +7\}$. Fixing the values of co-ordinates i to 0, i.e. $x_B^i, \dots, x_B^0$, we obtain subsets of the signal set $\mathbf{S}$ by defining:

$$\begin{aligned}\mathbf{S}(x_B^i, \dots, x_B^0) = \{\vec{a} = \mathcal{M}(\vec{x}) \mid \vec{x} = (b^{m-1}, \dots, b^{i+1}, x_B^i, \dots, x_B^0), b^j \in \{0,1\},\\ j = i+1, \dots, m-1\} .\end{aligned} \tag{3.4}$$

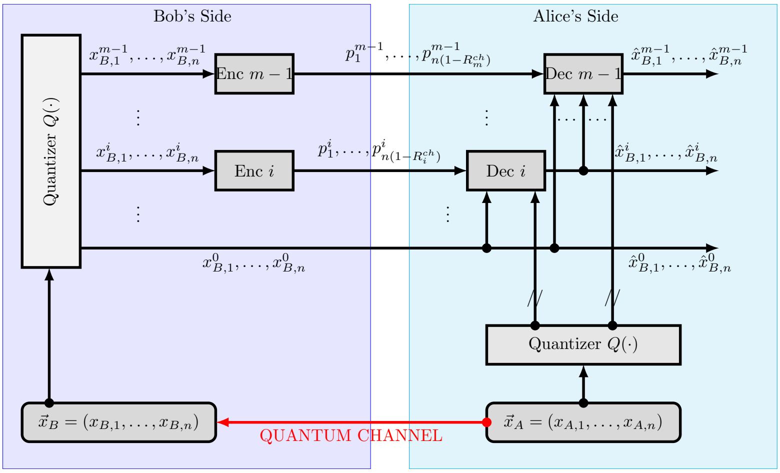

Bob's Side
Alice's Side
Quantizer $Q(\cdot)$
$x^{m-1}_{B,1},\dots,x^{m-1}_{B,n}$
Enc $m-1$
$p^{m-1}_1,\dots,p^{m-1}_{n(1-R^{ch}_m)}$
Dec $m-1$
$\hat{x}^{m-1}_{B,1},\dots,\hat{x}^{m-1}_{B,n}$
$x^i_{B,1},\dots,x^i_{B,n}$
Enc i
$p^i_1,\dots,p^i_{n(1-R^{ch}_i)}$
Dec i
$\hat{x}^i_{B,1},\dots,\hat{x}^i_{B,n}$
$x^0_{B,1},\dots,x^0_{B,n}$
$\hat{x}^0_{B,1},\dots,\hat{x}^0_{B,n}$
Quantizer $Q(\cdot)$
$\vec{x}_B=(x_{B,1},\dots,x_{B,n})$
QUANTUM CHANNEL
$\vec{x}_A=(x_{A,1},\dots,x_{A,n})$

Figure 3.2: The MLC-MSD scenario for RR. First the input is quantized into an m-bit source. Then each of the m sources is encoded and sent to Alice. The decoder has the side information from its own source and with the m encoded sources produces an estimate of the quantized source. Usually we transmit the least significant bits as plain-texts..

For more details about set partitioning and mapping see [7]. For the above mentioned constellation points, with $M = 8$ and binary partitioning:

$$\begin{aligned}
\mathbf{S}(x_B^0=0) &= \{\vec{a}=\mathcal{M}(\vec{x})|\vec{x}=\{000,010,100,110\}\} &&= \{-7,-3,+1,+5\}\ ,\\
\mathbf{S}(x_B^1x_B^0=10) &= \{\vec{a}=\mathcal{M}(\vec{x})|\vec{x}=\{010,110\}\} &&= \{-3,+5\}\ ,\\
\mathbf{S}(x_B^2x_B^1x_B^0=010) &= \{\vec{a}=\mathcal{M}(\vec{x})|\vec{x}=\{010\}\} &&= \{-3\}\ .
\end{aligned}$$

Figure 3.3 illustrates the schematic representation of the binary set partitioning for the above constellation points. In Section 3.2.2 we talk with more details about the digitization effect.

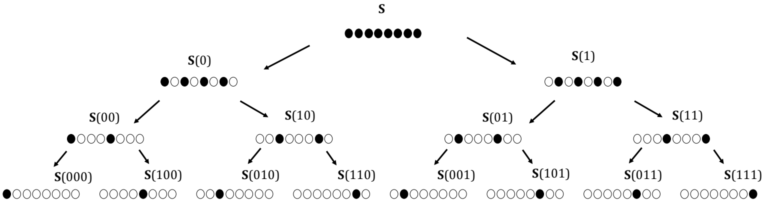


Figure 3.3: The binary set partitioning and the corresponding mapping.

In addition Figure 3.2 contains m individual levels. Some of the levels are transmitted as plain-text and some of the levels contain encoder and decoder blocks. In Section 3.2.5 we propose both analytical and numerical methods to calculate the code rates for individual levels. In addition, in Section 3.2.3 we explain with more details how encode the corresponding binary data in each level and how to use soft information for decoding purpose. Also we assume that LDPC codes are used for decoding.

3.2.2 Digitization effect

A digitizer converts continuous values to some discrete levels, characterized by a range R and number of output bits m. It is assumed that the digitizer uses the the following signal values: $\{-(M-1),\dots,-3,-1,1,3,\dots,(M-1)\}$, where $M=2^m$. The constellation symbols are normalized so that the average energy is equal to 1. Usually, $R=6\sigma$ is enough for the range of the digitizer. The digitizer provides a set of $M=2^m$ non-overlapping intervals with equal length $\delta=\dfrac{2R}{M-2}$ as follows:

$$I_j=\begin{cases}(-\infty,-R] & \text{if } j=0\,,\\ (-R+(j-1)\delta,-R+(j)\delta] & \text{if } 0<j<M-1\,,\\ [R,+\infty) & \text{if } j=M-1\,.\end{cases} \tag{3.5}$$

Let us denote by $\mathcal{Q}(x_B)$ the m-bit quantized version of x_B with its binary representation vector $(x_B^{m-1},\dots,x_B^1,x_B^0)$. Also, it is possible to assign different mappings to the output of the digitizer. For example, as depicted in Figure 3.4, an $m=4$ bit digitizer with range R is considered. Some possible mappings for the output are also presented. Considering a fixed step size δ for digitization, the entropy of the quantized source can be approximated by

$$H(Q(X_B))\approx h(X_B)-\log_2\delta\ ,$$

where $h(X_B)$ is the differential entropy defined for continuous variable X_B. A similar digitization can be applied on the Alice's side to get $Q(X_A)$. This also holds for the conditional entropy:

$$H(Q(X_B)|Q(X_A))\approx h(X_B|X_A)-\log_2\delta\ .$$

For the mutual information, if m is large enough:

$$I(Q(X_B);Q(X_A))\approx I(Q(X_B);X_A)\approx I(X_B;X_A)\ ,$$

where the equality holds when $\delta\to 0$.

Example 3.2.1. *The m-bit digitizer*

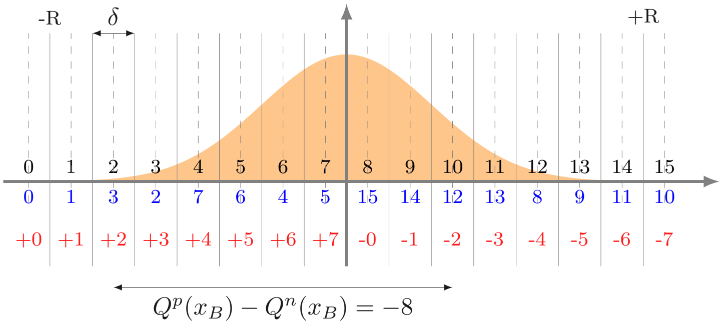


Figure 3.4: A digitizer with 4 bits. It provides 16 non-overlapping intervals. The binary, Gray and sign-magnitude mappings are also considered. It is clear that the distance between two points with equal least-significant bits (LSB)s is always fixed and it is equal to 8. Two quantized points $Q^p(x_B)$ and $Q^n(x_B)$ have equal LSBs, but their MSBs are assigned to 0 and 1, respectively.

As an example, consider a case when the covariance matrix of a bi-variate normal distribution is:

$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 4 \end{bmatrix} .$$

Then, the theoretical value for the differential entropy, can be calculated as:

$$h(X,Y) = 0.5 \ \log_2 \left((2\pi e)^2 \ \det(\Sigma)\right) \ ,$$

where $\det(\Sigma)$ denotes the determinant of the covariance matrix and the corresponding mutual information is equal to $I(X;Y) = h(X) + h(Y) - h(X,Y)$, as presented in Table 3.1. In addition, using a Monte-Carlo simulation, numerical estimations are calculated for $m \in \{3,4,5,6\}$, where m denotes the m-bit digitizer. Also, the corresponding histograms are plotted in Figure 3.5 for marginal and the joint distributions. It is clear that for the $m = 3$, the estimated values are not accurate, but for the $m \geq 4$, the numerical calculations are close to the theoretical values.

Theoretical values

	$I(X_A; X_B)$	$h(X_A)$	$h(X_B)$
–	0.1632	2.0471	3.0471

Numerical estimations

m	$I(Q(X_A); Q(X_B))$	$H(Q(X_A))$	$H(Q(X_B))$
3	0.136293	2.103829	3.103865
4	0.155513	3.061083	4.061103
5	0.160985	4.050079	5.049973
6	0.162606	5.047190	6.047170

Table 3.1: The theoretical values and the numerical estimations for m-bit digitizer with $\delta = 2^{3-m}$.

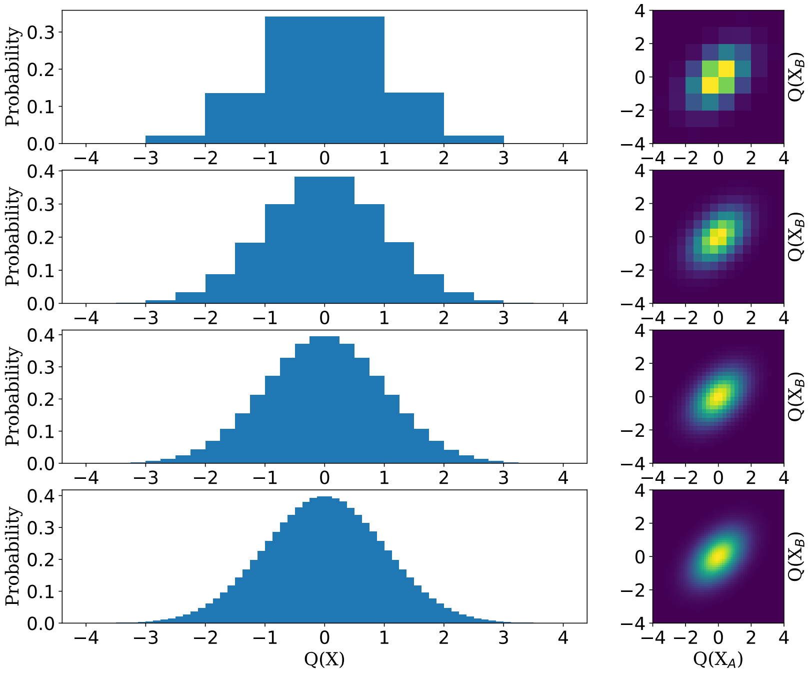


Figure 3.5: The digitization effect on the 1-D histogram (related to X_A) and corresponding 2-D histogram of the normalized quantized data.

3.2.3 MLC-MSD reconciliation with one-level coding

Now let us discuss the reconciliation process using MLC-MSD scheme. Also consider RR, where Alice reconciles her values to match Bob's. In this case, the reconciliation process can be fully described as a conventional communication theory problem. We start with the case when just the MSB of the Bob's data is used to generate the key (see Figure 3.6). Then we extend for two-levels and multi-levels. This problem was first addressed in [50] as source coding with side information. Assume that Alice and Bob have access to two correlated information sources X_A and X_B which follow a joint probability distribution $p_{X_A X_B}(x_A, x_B)$. The two parties wish to distill a common binary string from blocks of length n, $\vec{x}_A = (x_{A,i})_{i=1}^n$, $\vec{x}_B = (x_{B,i})_{i=1}^n$, by exchanging information as shown in Figure 3.6. In this configuration, Bob sends to Alice an encoded (compressed) version of his MSB $(x_{B,i}^{m-1})_{i=1}^n = \left(x_{B,1}^{m-1}, \cdots, x_{B,n}^{m-1}\right)$ and all the other LSBs are transmitted as plain-text. Then Alice uses her side information $\vec{x}_A = (x_{A,i})_{i=1}^n$ and Bob's LSBs to estimate $(x_{B,i}^{m-1})_{i=1}^n$ from $\vec{s}$.

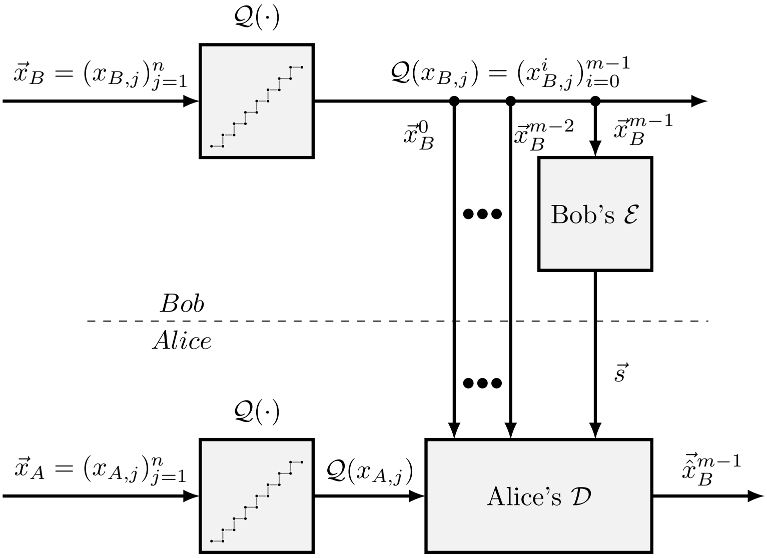


Figure 3.6: Reverse reconciliation scheme when just the most significant bit encoded. The remaining $m-1$ bits are transmitted as plain-text.

Here, the operation of each block is described with more details. Specifically, we describe the operation of Bob's encoder and the corresponding decoder on Alice's sides. In addition, we demonstrate how to calculate the soft information at the input of the Alice's decoder.

3.2.3.1 Bob's encoder

Let us denote by $\vec{s}$ the syndrome generated by Bob. For a given parity check matrix $H_{(n-k)\ n}$ and a bit string $\vec{x}_B^{m-1} = (x_{B,i}^{m-1})_{i=0}^{n-1}$ of length n, the syndrome $\vec{s}$ is calculated as:

$$\vec{s} = \vec{x}_B^{m-1} H^T \,, \tag{3.6}$$

where H^T denotes the transpose of the parity check matrix of the LDPC code (see Appendix A). Bob's encoder is actually a hash function, which compresses the data of length n to a syndrome of length $n-k$. Some other important facts, regarding the syndrome and Bob's encoder are the following:

- The syndrome is a vector of length $n-k$.
- The syndrome can be a non-zero vector depending on the Bob's received sequence.
- Each 0 value in $\vec{s}$ denotes that the corresponding parity equation is satisfied for $\vec{X}_B^{m-1}$.
- Each 1 value in $\vec{s}$ denotes that the corresponding parity equation is not satisfied for $\vec{X}_B^{m-1}$.

3.2.3.2 Calculation of the soft information

According to [44] the a-priori log-likelihood ratio (LLR) for a single bit is defined as:

$$\lambda_X = \ln\left(\frac{\Pr\{x=1\}}{\Pr\{x=0\}}\right) = \ln\left(\frac{p}{1-p}\right) \,, \tag{3.7}$$

where $p = \Pr\{x=1\}$. In a similar way, at the input of the decoder on Alice's side, the a-posteriori LLR for Bob's MSB, given Alice's quantized data and Bob's LSBs can be defined as:

$$\lambda_{X_B^{m-1}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-2}} = \ln\left(\frac{\Pr\{x_B^{m-1}=1|Q(x_A),\,\left(x_B^i\right)_{i=0}^{m-2}\}}{\Pr\{x_B^{m-1}=0|Q(x_A),\,\left(x_B^i\right)_{i=0}^{m-2}\}}\right) \,. \tag{3.8}$$

In addition, using the rules of the conditional probabilities

$$\Pr\{X,\,Y|Z\} = \Pr\{X|Z\}\Pr\{Y|X,\,Z\}, \tag{3.9}$$

one can write

$$\Pr\{X_B^{m-1}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-2}\} = \frac{\Pr\{X_B^{m-1},\,\left(X_B^i\right)_{i=0}^{m-2}|Q(X_A)\}}{\Pr\{\left(X_B^i\right)_{i=0}^{m-2}|Q(X_A)\}}, \tag{3.10}$$

where (3.10) can be used to simplify both numerator and denominator of (3.8) as follows:

$$\lambda_{X_B^{m-1}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-2}} = \ln\left(\frac{\Pr\{x_B^{m-1}=1,\,\left(x_B^i\right)_{i=0}^{m-2}|Q(x_A)\}}{\Pr\{x_B^{m-1}=0,\,\left(x_B^i\right)_{i=0}^{m-2}|Q(x_A)\}}\right). \tag{3.11}$$

Now, let us define two new random variables $Q^p(X_B)$ and $Q^n(X_B)$. $Q^p(X_B)$ takes the LSBs of Bob's data and assumes the MSB is equal to 0 and $Q^n(X_B)$ takes the LSBs of Bob's data and assumes the MSB is equal to 1. As depicted in Figure 3.4, $Q^p(X_B)$ belongs to the set of symbols on left hand side of the plane and $Q^n(X_B)$ belongs to the set of symbols on the right hand side.

Furthermore, it is clear that

$$\Pr\{Q(X_B)|Q(X_A)\} \approx \frac{\exp\left(\dfrac{-\left(Q(x_B)-Q(x_A)\right)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}}, \tag{3.12}$$

where σ^2 in (3.12) is the combination of the quantum channel noise and the digitizer noise. Using (3.12), equation (3.11) can be simplified to:

$$\begin{aligned}\lambda_{X_B^{m-1}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-2}} &= \ln\left(\frac{\exp\left(\dfrac{-\left(Q^n(x_B)-Q(x_A)\right)^2}{2\sigma^2}\right)}{\exp\left(\dfrac{-\left(Q^p(x_B)-Q(x_A)\right)^2}{2\sigma^2}\right)}\right)\\ &= \frac{\left(Q^p(x_B)-Q(x_A)\right)^2-\left(Q^n(x_B)-Q(x_A)\right)^2}{2\sigma^2}\\ &= \frac{\left(Q^p(x_B)-Q^n(x_B)\right)\left(Q^p(x_B)+Q^n(x_B)-2\,Q(x_A)\right)}{2\sigma^2}\\ &= \frac{-2^{m-1}\left(Q^p(x_B)+Q^n(x_B)-2\,Q(x_A)\right)}{2\sigma^2},\end{aligned} \tag{3.13}$$

where on the last equation, as depicted in Figure 3.4, for a digitizer with m-bits, $Q^p(x_B)-Q^n(x_B)=-2^{m-1}$. Thus

$$\lambda_{X_B^{m-1}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-2}} = \frac{-2^{m-1}\left(Q^{\text{avg}}(x_B)-Q(x_A)\right)}{\sigma^2}, \tag{3.14}$$

where, $Q^{\text{avg}}(x_B)=\frac{Q^p(x_B)+Q^n(x_B)}{2}$ denotes the average. This soft information can be used at the input of the LDPC decoder to estimate X_B^m.

3.2.3.3 Alice's decoder

Let us describe Alice's decoder. Here we show how Alice needs to modify her decoder to use the non-zero syndrome $\vec{s}$. The conventional LDPC decoder [51, 52] accepts soft information, and tries to recover the codeword from noisy received data by applying the message passing (MP) algorithm (See [44] and Appendix A). Let λ_v^0 denote Bob's MSB soft information as shown in (3.14) for variable-node (VN) v, on iteration 0. In addition assume that λ_{vc}^l (μ_{cv}^l) denotes the message from variable-node to check-node (from check-node to variable-node) after l iterations. The update equations for the messages under belief propagation are:

$$\lambda_{vc}^l = \begin{cases} \lambda_v^0 & \text{if } l = 0 \\ \lambda_v^0 + \sum\limits_{c' \in C_v \backslash c} \mu_{c'v}^l & \text{if } l \geq 1 \end{cases}$$

$$\mu_{cv}^l = 2 \tanh^{-1} \left(\prod_{v' \in V_c \backslash v} \tanh \left(\frac{\lambda_{v'c}^{l-1}}{2} \right) \right) ,$$

where $C_v \backslash c$ denotes all the check-nodes (CN) connected to the VN v, except the CN c and $V_c \backslash v$ is the set of VNs connected to CN c, except the VN v. The above formulation is valid when the syndrome $\vec{s}$ is an all zero vector, or all the parity equations are satisfied with zero parity.

In contrast to the conventional LDPC decoder, for the 1-level scheme the decoder on the Alice's side, as denoted in Figure 3.6, accepts one more additional input, denoted by syndrome $\vec{s}$, which is the syndrome of Bob's MSB which can be calculated from (3.6). The syndrome $\vec{s}$ is in general, a non-zero vector. Thus, we need to change the CN operation according to the received syndrome $\vec{s}$. The modified CN operation then can be represented as:

$$\mu_{cv}^l = (-1)^{S_c} \cdot 2 \tanh^{-1} \left(\prod_{v' \in V_c \backslash v} \tanh \left(\frac{\lambda_{v'c}^{l-1}}{2} \right) \right) , \qquad (3.15)$$

where $S_c \in \{0, 1\}$ represents the parity value at index c. It is clear that for $S_c = 0$, the CN operation remains the same as standard decoder, but for $S_c = 1$, the CN operation would flip the sign of the outgoing message.

3.2.3.4 Noise estimation

Accurate estimation of noise variance plays an important role for the calculation of the soft information. According to (3.14), incorrect estimation of the noise variance can highly affect the reliability of the soft information. Here, we present a method to obtain an accurate estimation of the noise variance. Under the assumption that the channel variance is not changing very fast, one can use some portion of the quantized data to estimate the noise variance.

Let us assume that we have sufficient samples of two jointly zero mean Gaussian random variables X_A and X_B, with standard deviations σ_A and σ_B, respectively. The bi-variate normal distribution can be described by:

$$f_{X_B,X_A}(x_B,x_A)=\frac{1}{2\pi\sqrt{|\Sigma|}}\exp\left(-\frac{1}{2}(x_A,x_B)\Sigma^{-1}(x_A,x_B)^T\right)\ , \tag{3.16}$$

where the covariance matrix is equal to:

$$\Sigma=\begin{bmatrix}\sigma_A^2 & \rho\,\sigma_A\,\sigma_B\\ \rho\,\sigma_A\,\sigma_B & \sigma_B^2\end{bmatrix}\ , \tag{3.17}$$

and

$$\rho=\frac{\mathbb{E}\{X_AX_B\}}{\sigma_A\sigma_B}\ , \tag{3.18}$$

is the correlation coefficient between X_A and X_B. By normalizing Alice's and Bob's data by their respective standard deviation, i.e.

$$x_A^j\to x_A^j/\sigma_A\ ,$$
$$x_B^j\to x_B^j/\sigma_B\ ,$$

the covariance matrix becomes

$$\Sigma\to\Sigma=\begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\ . \tag{3.19}$$

The conditional probability distribution describing Alice's outcome conditioned on Bob's is given by:

$$f_{X_A|X_B}(x_A|x_B)=\mathcal{N}(\rho x_B,(1-\rho^2))\ . \tag{3.20}$$

Using the given covariance matrix in (3.17), the mutual information of a bi-variate normal distribution can be calculated as:

$$I(X_A;X_B)=-\frac{1}{2}\log_2(1-\rho^2)\ . \tag{3.21}$$

For prepare and measure protocols, another interpretation of the mutual information is

$$I(X_A;X_B)=\frac{1}{2}\log_2(1+\mathrm{SNR})=\frac{1}{2}\log_2(1+\frac{\sigma_A^2}{\sigma_N^2})\ , \tag{3.22}$$

where σ_A^2 denotes the variance of the Alice's data and σ_N^2 denotes the variance of the additive white Gaussian noise channel. Then,

$$\mathrm{SNR}=\frac{\rho^2}{1-\rho^2}\ . \tag{3.23}$$

3.2.4 MLC-MSD reconciliation with two-level coding

Now let us extend the reconciliation scheme to a more general case when two levels are encoded and the rest of the LSBs are transmitted as plain-text (See Figure 3.7). In this case, the encoders and decoders work at two different code rates for each individual level. We defer the calculation of the individual rates to Section 3.2.5. As depicted in Figure 3.7, it is assumed that Bob sends $m-2$ LSBs directly to Alice and for the two MSBs, syndromes are sent. Then, Alice first tries to estimate the LLR of X_B^{m-2} and then estimates the X_B^{m-1}. The calculation of the soft information for the MSB was discussed in Section 3.2.3.2 . Here, we calculate the soft information on the Alice's side for the 2$^{\text{nd}}$ level (Second most significant bit).

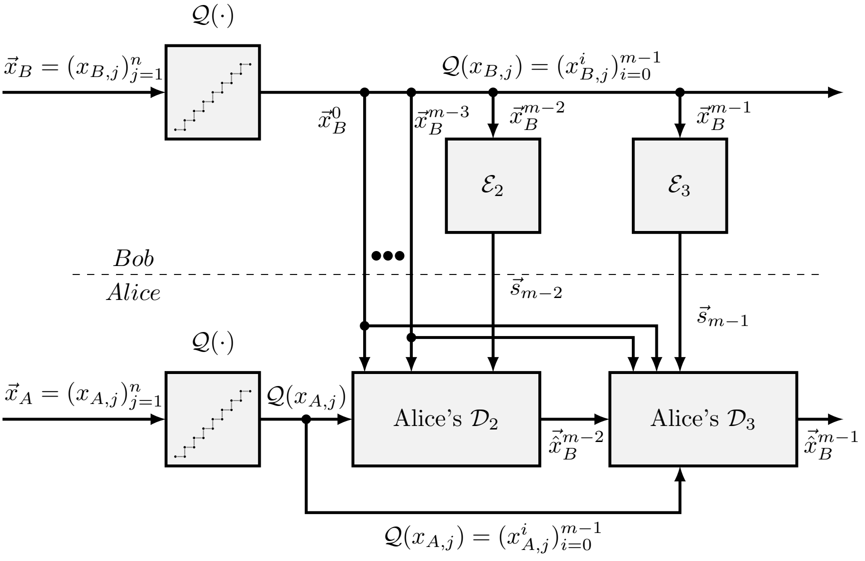


Figure 3.7: Reverse reconciliation when two most significant bits are encoded.

At the input of decoder $\mathcal{D}_2$ on Alice's side the LLR for Bob's second MSB, given Alice's quantized data and Bob's other remaining LSBs can be defined as:

$$\lambda_{X_B^{m-2}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-3}} = \ln\left(\frac{\Pr\{x_B^{m-2}=1|Q(x_A),\,\left(x_B^i\right)_{i=0}^{m-3}\}}{\Pr\{x_B^{m-2}=0|Q(x_A),\,\left(x_B^i\right)_{i=0}^{m-3}\}}\right). \tag{3.24}$$

Again, using (3.9), we can write

$$\Pr\{X_B^{m-2}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-3}\} = \frac{\Pr\{X_B^{m-2},\,\left(X_B^i\right)_{i=0}^{m-3}|Q(X_A)\}}{\Pr\{\left(X_B^i\right)_{i=0}^{m-3}|Q(X_A)\}}\;, \tag{3.25}$$

where (3.25) can be used to simplify both numerator and denominator of (3.24) as follows:

$$\lambda_{X_B^{m-2}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-3}} = \ln\left(\frac{\Pr\{x_B^{m-2}=1,\,\left(x_B^i\right)_{i=0}^{m-3}|Q(x_A)\}}{\Pr\{x_B^{m-2}=0,\,\left(x_B^i\right)_{i=0}^{m-3}|Q(x_A)\}}\right). \tag{3.26}$$

In addition, let us define four new random variables $Q^{pp}(X_B)$, $Q^{pn}(X_B)$, $Q^{np}(X_B)$ and $Q^{nn}(X_B)$. Where, $Q^{pp}(X_B)$ takes LSBs of Bob's data and assumes that the two MSBs are equal to 00 and so on. The numerator of (3.26) can be expanded as:

$$\begin{aligned}\Pr\{x_B^{m-2}=1,\,\left(x_B^i\right)_{i=0}^{m-3}|Q(x_A)\} &= \frac{1}{2}\Pr\{x_B^{m-1}=0,\,x_B^{m-2}=1,\,\left(x_B^i\right)_{i=0}^{m-3}|Q(x_A)\}\\ &+\frac{1}{2}\Pr\{x_B^{m-1}=1,\,x_B^{m-2}=1,\,\left(x_B^i\right)_{i=0}^{m-3}|Q(x_A)\}\\ &=\frac{\Pr\{Q^{pn}(x_B)|Q(x_A)\}+\Pr\{Q^{nn}(x_B)|Q(x_A)\}}{2},\end{aligned}$$

where in the last equation, it is assumed that $\Pr\{x_B^{m-1}=0|Q(x_A)\}=\Pr\{x_B^{m-1}=1|Q(x_A)\}=\frac{1}{2}$. This assumption is valid, because at this step, decoder $\mathcal{D}_2$ does not have any information about X_B^3. In other words, it can take zero or one with equal probability. Then, (3.26), can be simplified to:

$$\lambda_{X_B^{m-2}|Q(X_A),\,\left(X_B^i\right)_{i=0}^{m-3}} = \ln\left(\frac{\exp\left(\frac{-(Q^{nn}(x_B)-Q(x_A))^2}{2\sigma^2}\right)+\exp\left(\frac{-(Q^{pn}(x_B)-Q(x_A))^2}{2\sigma^2}\right)}{\exp\left(\frac{-(Q^{np}(x_B)-Q(x_A))^2}{2\sigma^2}\right)+\exp\left(\frac{-(Q^{pp}(x_B)-Q(x_A))^2}{2\sigma^2}\right)}\right), \tag{3.27}$$

which is the soft information at the input of the decoder $\mathcal{D}_2$.

The generalization to the higher levels is straight forward but in practice just two MSBs are encoded. In fact the channel coding rates for LSBs are very close to zero and no practical error correction exists for LSBs. In Section 3.2.5 we calculate the individual code rates for each level. Then it would be clear that just two MSBs are enough for the reconciliation tasks and all the remaining LSBs are then transmitted as plain-text.

3.2.5 Calculation of the individual source coding rates

3.2.5.1 Analytical method

For an m-bit digitizer, the individual conditional mutual information for each level is defined as:

$$\begin{aligned} I_i &= I(X_A; X_B^i | X_B^{i-1}, \ldots, X_B^0) \ , \\ &= H(X_B^i | X_B^{i-1}, \ldots, X_B^0) - H(X_B^i | X_A, X_B^{i-1}, \ldots, X_B^0) \ . \end{aligned} \tag{3.28}$$

Thus, using (3.3) it is clear that:

$$R_i^{\text{Source}} \geq H(X_B^i | X_B^{i-1}, \ldots, X_B^0) - I_i \ . \tag{3.29}$$

Equation (3.29) offers an analytical way to calculate the individual source rates for each level and, the only non-trivial quantity is the individual conditional mutual information I_i. To calculate I_i, the MLC-MSD approach is used [7]. Using the chain rule, we can always describe the total mutual information as a summation of conditional mutual information for individual levels

$$\begin{aligned} I(X_A; Q(X_B)) = \ & I(X_A; X_B^{m-1}, \ldots, X_B^0) = \ I(X_A; X_B^0) + I(X_A; X_B^1 | X_B^0) \\ & + \ldots + I(X_A; X_B^i | X_B^{i-1}, \ldots, X_B^0) + \ldots \\ & + I(X_A; X_B^{m-1} | X_B^{m-2}, \ldots, X_B^0) \ , \end{aligned} \tag{3.30}$$

which motivates our definition of the individual conditional mutual information in (3.28). According to [7], we can expand I_i as follows:

$$\begin{aligned} I_i =& I(X_A; X_B^i | X_B^{i-1}, \ldots, X_B^0) \qquad (3.31) \\ =& I(X_A; X_B^{m-1}, \ldots, X_B^i | X_B^{i-1}, \ldots, X_B^0) - I(X_A; X_B^{m-1}, \ldots, X_B^{i+1} | X_B^i, \ldots, X_B^0) \ , \end{aligned}$$

where each term on the right hand side can be calculated separately. More precisely,

$$\begin{aligned} & I(X_A; X_B^{m-1}, \ldots, X_B^i | X_B^{i-1}, \ldots, X_B^0) = \\ & \qquad \mathbb{E}_{x_B^{i-1}, \ldots, x_B^0 \, \in \, \{0,1\}^i} \left\{ \ I(X_A; X_B^{m-1}, \ldots, X_B^i | x_B^{i-1}, \ldots, x_B^0) \right\} \ , \end{aligned}$$

where $\mathbb{E}$ denotes the expectation value and can be calculated by averaging over all possible combinations of $x_B^{i-1}, \ldots, x_B^0$. Finally, according to [7], the full characterization of I_i requires a set of probability density functions (PDF)s $\boldsymbol{f}_{X_A|X_B^i}(x_A|x_B^i)$ which can be defined as:

$$\left\{ f_{X_A|X_B} \left(x_A | x_B^i, x_B^{i-1}, \ldots, x_B^0 \right) \mid \left(x_B^{i-1}, \ldots, x_B^0 \right) \in \{0,1\}^i \right\} \ ,$$

where,

$$f_{X_A|X_B}(x_A | x_B^i, x_B^{i-1}, \ldots, x_B^0) = \mathbb{E}_{b \, \in \, \mathbf{S}(x_B^i, \ldots, x_B^0)} \left\{ f_{X_A|X_B}(x_A | b) \right\} \ ,$$

where the signal point b is taken from the subset $\mathbf{S}(x_B^0 \ldots x_B^i)$ and $f_{X_A|X_B}(x_A|b)$ is the equivalent conditional quantized PDF of the continuous conditional PDF presented on (3.20).

Example 3.2.2. *Simulation results for a 4-bit digitizer*

As an example, we present simulation results for individual rates for 16 bins, when Bob's quantized data has a discrete Gaussian PDF. The results for individual channel coding rates are presented in Figure 3.8. The equivalent channel coding rate is equal to $R_i^{ch} = 1 - R_i^{\text{Source}}$. We assumed that Alice's data has a continuous Gaussian distribution with variance equal to 1 and both Alice and Bob use 4 level digitizers. Also, a fixed step size $\delta = 0.32$ is assumed for digitization of normalized continuous variables.

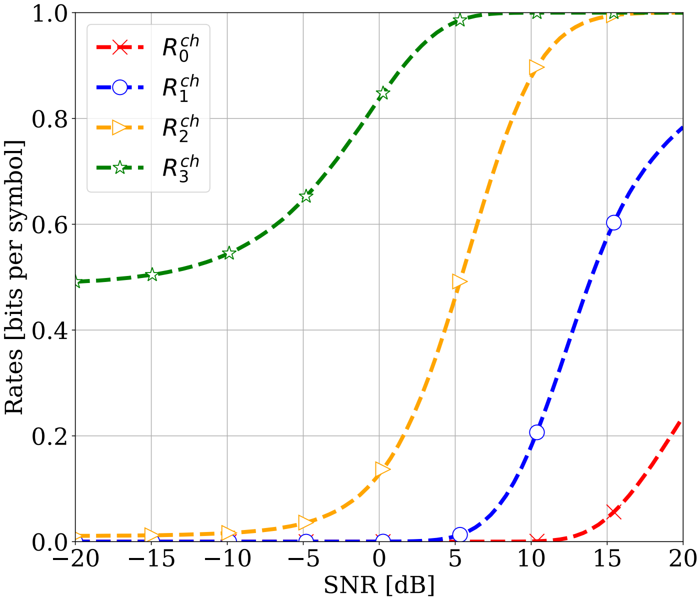


Figure 3.8: The equivalent channel coding rates ($R_i^{ch} = 1 - R_i^{\text{Source}}$) for 4-levels with binary partitioning versus SNR.

Figure 3.9 denotes the conditional mutual information for the i^{th} individual channel (I_i) as a function of the SNR. It is clear that the summation of the all individual conditional mutual information is very close to the Shannon capacity of the AWGN channel, and the small difference is a results of the digitization effect.

By summing up over all the individual mutual information taken from (3.28) it is clear that:

$$\sum_{i=0}^{m-1} I_i = I(\mathcal{Q}(X_B); X_A) \leq I(X_B; X_A) \ .$$

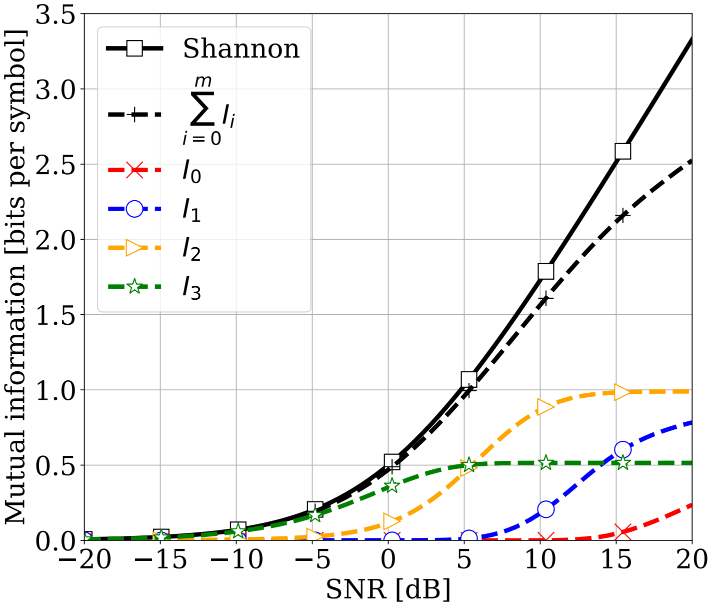


Figure 3.9: The individual conditional mutual information for the 4-levels with binary partitioning versus SNR.

3.2.5.2 Numerical calculation method

In this part, I propose a numerical method to calculate the individual source coding rates and individual mutual information between different levels. As long as we have access to some portion of Alice's and Bob's data we can find the joint probability mass function (PMF) and their marginal PMFs for Alice and Bob. For instance Figure 3.10 shows the numerical PMF extracted from the quantized data. To get the marginal PMFs it is enough to sum over the rows or columns of the 2-D histogram. The color map on 2-D histogram denotes the joint probability values. Since the normalized data are used for the calculation both set of data related to Alice and Bob have the same variance equal to one.

Let us remind the equation of individual mutual information for RR as presented in (3.28)

$$\begin{aligned} I_i &= I(Q(X_A); X_B^i | X_B^{i-1}, \ldots, X_B^0) \ , \\ &= H(X_B^i | X_B^{i-1}, \ldots, X_B^0) - H(X_B^i | Q(X_A), X_B^{i-1}, \ldots, X_B^0) \ . \end{aligned} \tag{3.32}$$

where we replaced X_A by $Q(X_A) = X_A^i X_A^{i-1} \cdots X_A^0$ and the assumption that the

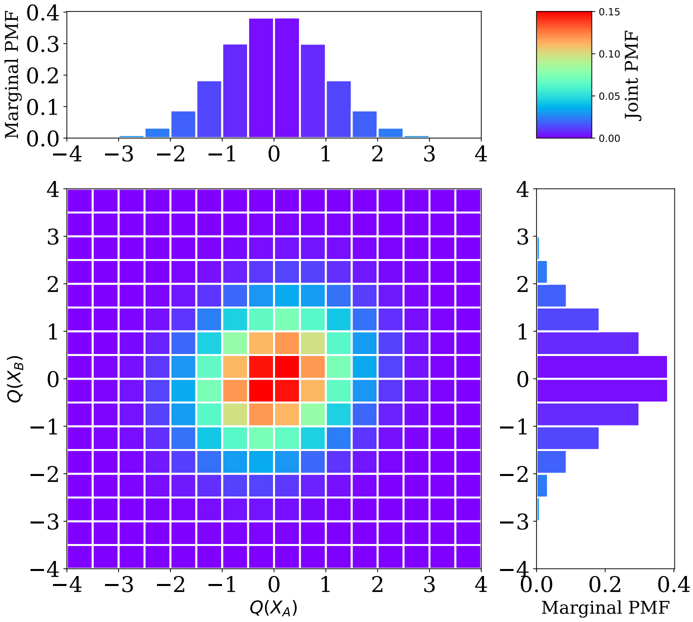


Figure 3.10: Numerical joint PMF for quantized bi-variate normally distributed random variables and their marginal PMFs for a 4-bit digitizer at SNR=−5 dB.

same digitizer is used on both sides. In the following we show how to calulate both parts of the right hand side of the above equation by using joint and the marginal PMFs.

For the first term, from the fact that $H(X|Y) = H(X,Y) - H(Y)$:

$$H(X_B^i|X_B^{i-1},\dots,X_B^0) = H(X_B^i,X_B^{i-1},\dots,X_B^0) - H(X_B^{i-1},\dots,X_B^0)\ .$$

Then, to calculate the term $H(X_B^i,X_B^{i-1},\dots,X_B^0)$, we can directly apply the definition of the entropy for discrete variables as follows:

$$H(X_B^i,\dots,X_B^0) = -\sum_{X_B^i,\dots,X_B^0} \Pr_{X_B^i,\dots,X_B^0}\{x_B^i,\dots,x_B^0\}\log_2\left(\Pr_{X_B^i,\dots,X_B^0}\{x_B^i,\dots,x_B^0\}\right)\ .$$

All the probabilities can be calculated from Bob's marginal PMF. For example $\Pr_{X_B^0}\{1\}$ denotes the probability that the LSB of Bob is equal to 1. In a similar way, $\Pr_{X_B^1 X_B^0}\{01\}$, denotes the probability that the two LSBs of Bob are 01.

For the second term in (3.32) it is possible to write

$$\begin{aligned} H(X_B^i|Q(X_A), X_B^{i-1}, \dots, X_B^0) = H(X_B^i, \dots, X_B^0|Q(X_A)) \\ - H(X_B^{i-1}, \dots, X_B^0|Q(X_A)) \ . \end{aligned}$$

And to calculate each term

$$H(X_B^i, \dots, X_B^0|Q(X_A)) = \sum_{Q(X_A)=a_0}^{a_{2^m-1}} \Pr_{Q(X_A)}\{a_i\} H(X_B^i, \dots, X_B^0|a_i) \ ,$$

where a_i is a realization of Alice's quantized data and $H(X_B^i, \dots, X_B^0|Q(X_A) = a_i)$ denotes the instant entropy value when $Q(X_A) = a_i$. The probability $\Pr_{Q(X_A)}\{a_i\}$ can be obtained by looking at Alice's marginal PMF and $H(X_B^i, \dots, X_B^0|Q(X_A) = a_i)$ can be calculated as:

$$-\sum_{X_B^i, \dots, X_B^0} \Pr_{X_B^i, \dots, X_B^0|Q(x_A)}\{x_B^i, \dots, x_B^0|a_i\} \log_2 \left(\Pr_{X_B^i, \dots, X_B^0|Q(x_A)}\{x_B^i, \dots, x_B^0|a_i\} \right) \ ,$$

where the probability values $\Pr_{X_B^i, \dots, X_B^0|Q(x_A)}\{x_B^i, \dots, x_B^0|a_i\}$ can be obtained from the joint PMF of Alice and Bob. Our numerical calculations confirm the same theoretical values for individual rates and individual mutual information (See Figure 3.9 and Figure 3.8).

3.3 Comparison between reverse and forward reconciliation

In Section 3.1 the difference between FR and RR was discussed in the context of multidimensional reconciliation. Here, we compare FR and RR for the MLC-MSD scheme. As illustrated in Figure 3.11, Alice and Bob are connected through two different channels (classical and quantum channels). First, Alice sends quantum states to Bob through a noisy quantum channel and later, depending on the reconciliation protocol, the two parties share some information on an authenticated classical channel. In the case of FR, Alice encodes her data and sends it to Bob who would correct his data according to syndromes generated by Alice, plus his own side information. On the other hand, in the case of RR, Bob sends the data, and Alice corrects her data according to his data (For detailed information see Figure 3.11).

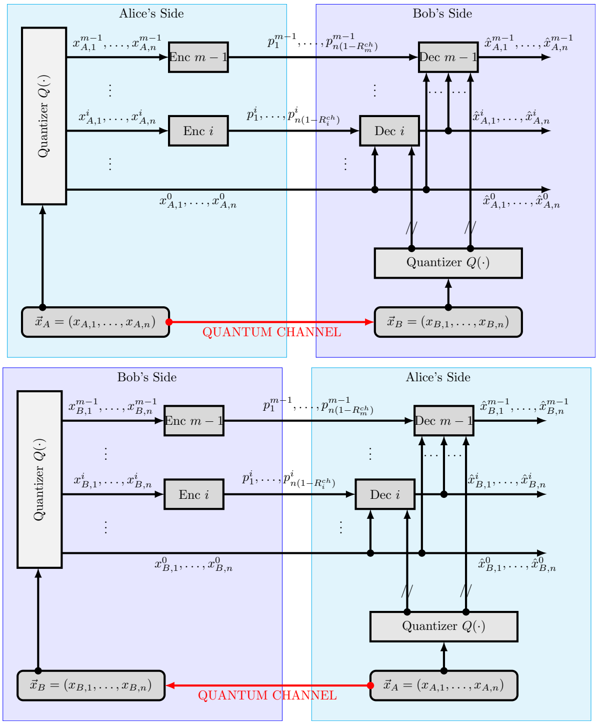

Alice's Side
Bob's Side
Quantizer $Q(\cdot)$
$x_{A,1}^{m-1},\ldots,x_{A,n}^{m-1}$
Enc $m-1$
$p_1^{m-1},\ldots,p_{n(1-R_m^{ch})}^{m-1}$
Dec $m-1$
$\hat{x}_{A,1}^{m-1},\ldots,\hat{x}_{A,n}^{m-1}$
$x_{A,1}^{i},\ldots,x_{A,n}^{i}$
Enc i
$p_1^{i},\ldots,p_{n(1-R_i^{ch})}^{i}$
Dec i
$\hat{x}_{A,1}^{i},\ldots,\hat{x}_{A,n}^{i}$
$x_{A,1}^{0},\ldots,x_{A,n}^{0}$
$\hat{x}_{A,1}^{0},\ldots,\hat{x}_{A,n}^{0}$
Quantizer $Q(\cdot)$
$\vec{x}_A=(x_{A,1},\ldots,x_{A,n})$
QUANTUM CHANNEL
$\vec{x}_B=(x_{B,1},\ldots,x_{B,n})$
Bob's Side
Alice's Side
Quantizer $Q(\cdot)$
$x_{B,1}^{m-1},\ldots,x_{B,n}^{m-1}$
Enc $m-1$
$p_1^{m-1},\ldots,p_{n(1-R_m^{ch})}^{m-1}$
Dec $m-1$
$\hat{x}_{B,1}^{m-1},\ldots,\hat{x}_{B,n}^{m-1}$
$x_{B,1}^{i},\ldots,x_{B,n}^{i}$
Enc i
$p_1^{i},\ldots,p_{n(1-R_i^{ch})}^{i}$
Dec i
$\hat{x}_{B,1}^{i},\ldots,\hat{x}_{B,n}^{i}$
$x_{B,1}^{0},\ldots,x_{B,n}^{0}$
$\hat{x}_{B,1}^{0},\ldots,\hat{x}_{B,n}^{0}$
Quantizer $Q(\cdot)$
$\vec{x}_B=(x_{B,1},\ldots,x_{B,n})$
QUANTUM CHANNEL
$\vec{x}_A=(x_{A,1},\ldots,x_{A,n})$

Figure 3.11: The comparison of forward (upper) and reverse (lower) reconciliation. Through the quantum channel Alice always sends quantum states of light to Bob. In the classical channel the direction of the messages are not the same and is determined by the protocol.

In both cases, the quantum channel can be described by $X_B = X_A + N$, where

N is the additive white Gaussian noise with zero mean and variance σ_N^2. Since, the two quantities X_A and N are independent, the following bounds are valid for the continuous data:

$$\begin{aligned} H(X_B) &\geq H(N) \ , \\ H(X_B) &\geq H(X_A) \ , \end{aligned}$$

For the reconciliation, it can be assumed that Alice and Bob have access to pairs of samples of a joint Gaussian distribution. Also, let us denote by $H(Q(X_A))$ and $H(Q(X_B))$ the entropy of the quantized version of Alice and Bob's data. According to Table 3.1 and Figure 3.5, for an efficient digitizer

$$\begin{aligned} &I(X_B; X_A) \approx I(Q(X_B); Q(X_A)) \ , \\ &H(Q(X_B)) - H(Q(X_B)|Q(X_A)) \approx H(Q(X_A)) - H(Q(X_A)|Q(X_B)) \ , \\ &H(Q(X_B)) + H(Q(X_A)|Q(X_B)) \approx H(Q(X_A)) + H(Q(X_B)|Q(X_A)) \ , \\ &H(Q(X_A), Q(X_B)) \approx H(Q(X_B), Q(X_A)) \ . \end{aligned}$$

According to the results presented in [50], and as depicted in Figure 3.12, in the case of FR (RR), it is enough for Alice and Bob to share $H(Q(X_A)|X_B)$ $(H(Q(X_B)|X_A))$ information (bit/symbol) through the classical channel to achieve perfect reconciliation. In this case the corresponding mutual information is

$$\begin{aligned} \text{FR:} \qquad & H(Q(X_A)) - H(Q(X_A)|X_B) \\ \text{RR:} \qquad & H(Q(X_B)) - H(Q(X_B)|X_A). \end{aligned}$$

The quantity $H(Q(X_A)|X_B)$ $(H(Q(X_B)|X_A))$ is a lower bound on source coding rate for FR (RR). For better understanding of the difference between these quantities as a function of SNR, a comparison has been done for the continuous variables on quantum channel using the differential entropy for Gaussian variables. The results are shown in Figure 3.13.

Now, let us compare with more details the forward and reverse reconciliation protocols. Also, let us denote by $I^s(X;Y) = I(X;Y)|_s$ and $H^s(X) = H(X)|_s$, the mutual information and the entropy at specific SNR s. The specific definition of these quantities would be clear from the context. In addition, remember that $R_i^{\text{Ch}} = 1 - R_i^{\text{Source}}$. For simplicity, we denote here the channel coding rate at SNR s by $R_{\text{RR}}^i|_s$ and $R_{\text{FR}}^i|_s$ for RR and FR, respectively.

3.3.1 Individual rates for reverse and forward reconciliations

It is known from (3.3) that for RR, the channel coding rate of the individual levels

$$R_{\text{RR}}^i|_s \leq 1 - H^s(X_B^i|X_A, X_B^{i-1}, \ldots, X_B^0) \ ,$$

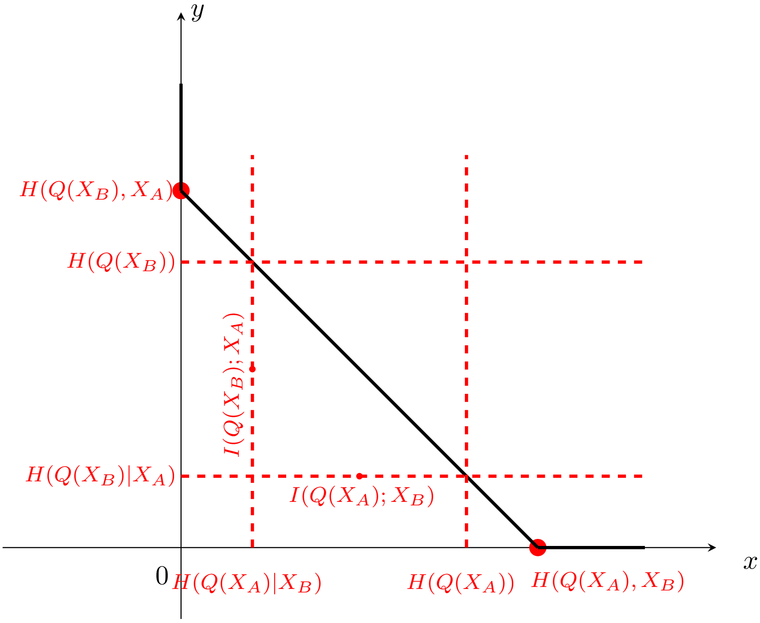


Figure 3.12: Admissible Rate Region according to Slepian-Wolf method.

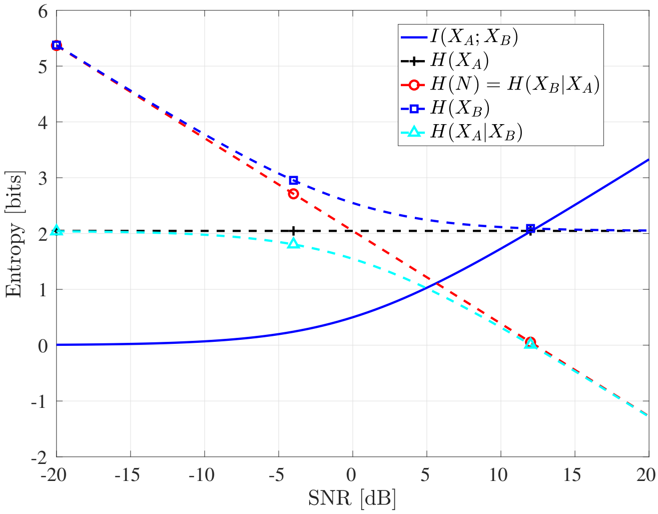


Figure 3.13: The theoretical values for differential entropies for continuous data on quantum channel.

thus,

$$
\begin{aligned}
R_{\mathrm{RR}}|_s &= \sum_i R^i_{\mathrm{RR}}|_s \leq m - H^s(Q(X_B)|X_A) \ , \\
I^i_{\mathrm{RR}}|_s &= I(X^i_B; X_A|X^{i-1}_B, \ldots, X^0_B) \ , \\
I^i_{\mathrm{RR}}|_s &= H^s(X^i_B|X^{i-1}_B, \ldots, X^0_B) - H^s(X^i_B|X_A, X^{i-1}_B, \ldots, X^0_B) \ .
\end{aligned}
$$

In a similar way, for FR:

$$
\begin{aligned}
R^i_{\mathrm{FR}}|_s &= 1 - H^s(X^i_A|X_B, X^{i-1}_A, \ldots, X^0_A) \ , \\
R_{\mathrm{FR}}|_s &= \sum_i R^i_{\mathrm{FR}}|_s \leq m - H^s(Q(X_A)|X_B) \ , \\
I^i_{\mathrm{FR}}|_s &= I(X^i_A; X_B|X^{i-1}_A, \ldots, X^0_A) \ , \\
I^i_{\mathrm{FR}}|_s &= H^s(X^i_A|X^{i-1}_A, \ldots, X^0_A) - H^s(X^i_A|X_B, X^{i-1}_A, \ldots, X^0_A) \ .
\end{aligned}
$$

Finally, for both schemes

$$
I_{\mathrm{RR}} = \sum_{i=1}^{m} I^i_{\mathrm{RR}} = I(X_A; Q(X_B)) \approx I(Q(X_A); X_B) = \sum_{i=1}^{m} I^i_{\mathrm{FR}} = I_{\mathrm{FR}} \ , \tag{3.33}
$$

where in (3.33), it is assumed that the digitization error is small and $I(X_A; Q(X_B)) \approx I(Q(X_A); X_B) \approx I(X_A; X_B)$.

3.3.2 Individual rates in terms of $I^i|_\infty$ and $I^i|_s$

The rate of the error correction codes can be represented in terms of the mutual information. In the case of FR:

$$
I^i_{\mathrm{FR}}|_s = H^s(X^i_A|X^{i-1}_A, \ldots, X^0_A) - H^s(X^i_A|X_B, X^{i-1}_A, \ldots, X^0_A) \ , \tag{3.34}
$$

and at $s = \infty$

$$
I^i_{\mathrm{FR}}|_\infty = H^\infty(X^i_A|X^{i-1}_A, \ldots, X^0_A) - H^\infty(X^i_A|X_B, X^{i-1}_A, \ldots, X^0_A) \ .
$$

since, $s \to \infty$ then, $X_B = X_A$ and $H^\infty(X^i_A|X_B, X^{i-1}_A, \ldots, X^0_A) = 0$, thus:

$$
I^i_{\mathrm{FR}}|_\infty = H^\infty(X^i_A|X^{i-1}_A, \ldots, X^0_A) = H^s(X^i_A|X^{i-1}_A, \ldots, X^0_A) \ . \tag{3.35}
$$

The last equality is valid, because the Alice's data is independent of the SNR. Then, for the case of FR

$$
R^i_{\mathrm{FR}}|_s = 1 - [I^i_{\mathrm{FR}}|_\infty - I^i_{\mathrm{FR}}|_s]. \tag{3.36}
$$

This can be checked by substituting (3.34) and (3.35) in (3.36), which is equivalent to the results presented in [9].

In a similar way for RR,

$$I^i_{\text{RR}}|_s = H^s(X^i_B|X^{i-1}_B, \ldots, X^0_B) - H^s(X^i_B|X_A, X^{i-1}_B, \ldots, X^0_B) \ , \tag{3.37}$$

when $s \to \infty$, again $X_A = X_B$ and $X^i_B = X^i_A$ for $i = 1, \ldots, m$, thus:

$$I^i_{\text{RR}}|_\infty = H^\infty(X^i_B|X^{i-1}_B, \ldots, X^0_B) - H^\infty(X^i_B|X_A, X^{i-1}_B, \ldots, X^0_B) \ ,$$

which can be simplified to

$$\begin{aligned} I^i_{\text{RR}}|_\infty &= H^\infty(X^i_B|X^{i-1}_B, \ldots, X^0_B) \\ &= H^\infty(X^i_A|X^{i-1}_A, \ldots, X^0_A) = H^s(X^i_A|X^{i-1}_A, \ldots, X^0_A) \ . \end{aligned} \tag{3.38}$$

Finally, subtracting (3.37) from (3.38):

$$\begin{aligned} I^i_{\text{RR}}|_\infty - I^i_{\text{RR}}|_s = {} & H^s(X^i_B|X_A, X^{i-1}_B, \ldots, X^0_B) \\ & + \underbrace{H^s(X^i_A|X^{i-1}_A, \ldots, X^0_A) - H^s(X^i_B|X^{i-1}_B, \ldots, X^0_B)}_{-\Delta^s_i} \ , \end{aligned}$$

or equivalently

$$H^s(X^i_B|X_A, X^{i-1}_B, \ldots, X^0_B) = I^i_{\text{RR}}|_\infty - I^i_{\text{RR}}|_s + \Delta^s_i \ , \tag{3.39}$$

where $\Delta^s_i = H^s(X^i_B|X^{i-1}_B, \ldots, X^0_B) - H^s(X^i_A|X^{i-1}_A, \ldots, X^0_A)$. Thus, for RR, the individual channel coding rate is:

$$R^i_{\text{RR}}|_s = 1 - H^s(X^i_B|X_A, X^{i-1}_B, \ldots, X^0_B) = 1 - [I^i_{\text{RR}}|_\infty - I^i_{\text{RR}}|_s] - \Delta^s_i \ , \tag{3.40}$$

which is not completely similar to (3.36) for FR, and has a correction term Δ^s_i.

Finally, it is clear that at $s \to \infty$, the two quantities $I^i_{\text{RR}}|_\infty$ and $I^i_{\text{FR}}|_\infty$ are equal (See (3.35) and (3.38)). By assuming that individual mutual information for FR and RR are almost equal to each other, for all the SNR points[1], i.e, $I^i_{\text{FR}}|_s \approx I^i_{\text{RR}}|_s$ then:

$$R^i_{\text{RR}}|_s = 1 - [I^i_{\text{RR}}|_\infty - I^i_{\text{RR}}|_s] - \Delta^s_i \approx 1 - [I^i_{\text{FR}}|_\infty - I^i_{\text{FR}}|_s] - \Delta^s_i \approx R^i_{\text{FR}}|_s - \Delta^s_i \ . \tag{3.41}$$

Finally, if we sum up over all individual channels

$$R_{\text{RR}}|_s \approx R_{\text{FR}}|_s - \sum_{i=1}^{m} \Delta^s_i = R_{\text{FR}}|_s + H(Q(X_A)) - H(Q(X_B)) \ . \tag{3.42}$$

[1] This assumption could be valid with a good approximation. At least, it can be assumed that the code rates are designed in such a way that the mutual information between the two scenarios (FR and RR) be equal and the summation of the mutual information of the individual levels are maximized on both cases.

3.4 Randomized Reconciliation

The concept of randomized decoding was first introduced in [53]. The other related algorithms in this family are known as list decoding and Chase algorithm [54, 55], where they use the soft measurement information at the input of a hard decoder. Specifically, the Chase algorithm uses a simple hard decoder and generates a list of candidate test-codewords by flipping the least reliable positions in the received sequence.

Before describing the algorithm for the randomized reconciliation, let us briefly describe the concept of reconciliation using list decoding. The concept behind this soft decoder is depicted in Figure 3.14. Here we assume that the core of the decoder is a simple hard decoder with the capability to find a unique codeword, when the noisy received sequence is surrounded by a sphere of radius $\lfloor \frac{d-1}{2} \rfloor$, where d is the minimum distance of the error correction code. Also, we assume four possible codewords $\vec{C}^1, \vec{C}^2, \vec{C}^3, \vec{C}^4$, and the received sequence $\vec{y}_H$. Thus, in this case, a hard decision decoder provides a unique codeword $\vec{C}^1$, with its unique error pattern $\vec{y}_H \oplus \vec{C}^1$. The goal of the list decoding is to generate different error patterns, by changing the least reliable positions in the received sequence $\vec{y}_H$. For the new test-vector $\vec{y}_T$, it is possible for our hard decoder to find a new codeword depending on whether or not $\vec{y}_T$ falls into the sphere of a new codeword.

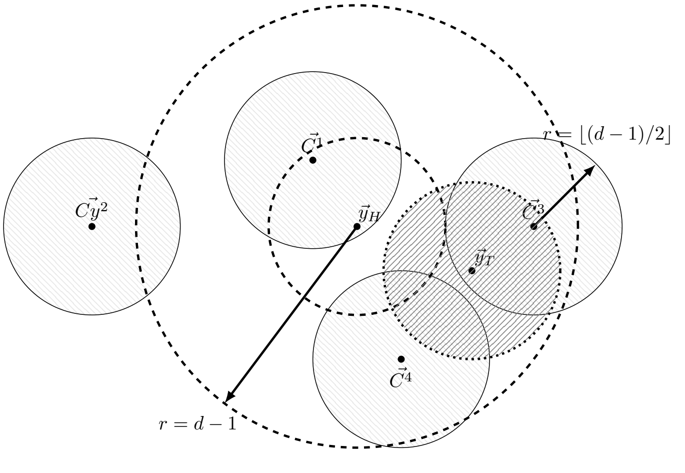


Figure 3.14: Decoding using measured soft information and hard decoder. Taken by some modification from [56] .

More specifically, in this section the same procedure is used for a reconciliation scheme. For the reconciliation, a channel code induces a partitioning of the source data space into cosets that can be indexed by their respective syndromes. Let C be a (n, k)-linear block code. The problem of decoding reconciliation using the corresponding channel code is to find the transmitted codeword, given the measured side information vectors $\vec{X_A}$ and $\vec{S_B}$ at the decoder. The encoder computes the syndrome of the sequence $\vec{y}_H = (y_{H,i})_{i=0}^{n-1}$ by computing $\vec{s} = \vec{y}_H \times \ H^T$, where H is the parity check matrix of the code. Then the decoding operation is equivalent to finding a codeword with the same syndrome. By assuming that each coset contains only one valid codeword this problem is equivalent to our channel coding problem.

Here, we propose a randomized reconciliation for CV-QKD using a modified version of the Chase algorithm called *stochastic Chase algorithm* (SCA) [57, 56]. In the SCA, the test-codewords are generated according to their reliability. Let us denote by $\vec{\lambda} = (\lambda_i)_{i=0}^{n-1}$ the log likelihood of Bob's soft information on Alice's side which is a vector of length n. It was shown in Section 3.2.3.2, how to calculate the soft measurement information (See (3.14)). Equivalently, its hard decision value $\vec{y}_H = (y_{H,i})_{i=0}^{n-1}$ can be calculated as:

$$y_{H,i} = \begin{cases} 1 & \text{if } \lambda_i \geq 0 \\ 0 & \text{if } \lambda_i < 0 \end{cases} .$$

Also, in the probability domain, the vector $\vec{p} = (p_i)_{i=0}^{n-1}$ denotes the probability that the corresponding bit at position $i \in \{0, \ldots, n-1\}$ is equal to 1, where

$$p_i = \frac{e^{\lambda_i}}{1 + e^{\lambda_i}} .$$

It is clear that $\lambda_i \in (-\infty, \infty)$, and $p_i \in (0, 1)$. When $\lambda_i \to \infty$ the value of $p_i \to 1$, and in similar way, when $r_i \to -\infty$ the value of $p_i \to 0$, and for $r_i \to 0$ the value of $p_i \to \frac{1}{2}$. This shows that the least reliable positions in a received codeword have LLR values close to 0, which are most likely the error positions.

The algorithm for the randomized reconciliation can be found in Algorithm 1. In the initialization, the algorithm extracts two kinds of information from $\vec{r}$. First, it finds the first τ least reliable positions in the received codeword and finds its equivalent hard decision $\vec{y}_H$. Second, it finds the probability of being 1 for each unreliable index. Then, in the main loop, the algorithm generates L different test-vectors. This can be easily done by flipping the value of the index i in $\vec{y}_H$ based on its flipping probability p_i. Then for each test-vector $\vec{y}_T$, the hard LDPC decoder finds the syndrome $\vec{s}_T$ and compares it with the Bob's syndrome $\vec{s}$. The algorithm ends when it finds a test-codeword which has zero Hamming distance with Bob's syndrome. Otherwise, the algorithm looks among all the L test-codewords and finds the one with minimum Hamming distance.

3.4.1 Complexity of the algorithm

Let us consider a reconciliation process based on MLC-MSD. The complexity of randomized reconciliation using SCA is compared with the complexity of a soft LDPC decoder. For the soft LDPC decoder a min-sum (MS) algorithm is used. The core of the SCA is a hard decoder using a bit-flipping (BF) algorithm. Let us denote the average number of VNs (CNs) by $\bar{d}_v$ $\left(\bar{d}_c\right)$. For each iteration of the MS algorithm,

Algorithm 1 Randomized Reconciliation

```
———————— Initialize
Step 1)  Find $\tau$ least reliable bits in $\vec{r}$
Step 2)  $\vec{j} = (j_i)_1^\tau$                 ▷ index of least reliable bits
Step 3)  $\vec{p} = (p_i)_1^\tau$                 ▷ probability of being one
Step 4)  $\vec{y}_H$                               ▷ equivalent hard decision vector
———————— Main loop
$l \Leftarrow 0$                                   ▷ counter for test-codewords
$L$                                                ▷ maximum number of test-codewords
while $l \le L$ do
    $\vec{y}_T = \vec{y}_H$
    for $i \in \vec{j}$ do
        $y_{T,i} = \mathrm{BSC}_{p_i}(0)$
    end for
    Decode $\vec{y}_T$, to get $\vec{x}_T$ and the syndrome $\vec{s}_T$
    Calculate the distance: $D_{\vec{s}_T,\vec{s}} = \sum_{i=1}^{m} S_i \bigoplus S_{T,i}$
    $l \Leftarrow l + 1$
end while
———————— Best codeword
The output codeword is the one with minimum $D_{\vec{s}_T,\vec{s}}$.
```

the VN v sums over all the incoming messages. Thus in average $n \cdot \bar{d}_v$ summation are required. On the other hand, for each CN c, assuming an average CN degree $\bar{d}_c$, the output message is calculated by finding the minimum of all incoming messages. Thus in general $m \cdot \bar{d}_c$ comparisons are necessary. Also, the same number of xor ($\oplus$) operations are required in order to find the sign of the output message.

In contrast, the BF algorithm just acts on the binary data and all the operations are $\oplus$ operations. The algorithm at each iteration finds the syndrome, which requires $m \cdot \bar{d}_c$ xor operations and then flips the position of some VNs when the specific bit has a set of non-zero syndromes larger than specific threshold T. The summary of our discussion can be found in Table 3.2.

Considering the fact that the complexity of bit-wise operations are negligible in comparison with operations acting on real numbers, it is clear that the MS algorithm has much higher complexity than BF algorithm.

Now, if we assume that the SCA uses L different codewords to find the best one, we can compare the complexity of MS and SCA. The SCA can apply L completely independent test-codewords and selects the best codeword by measuring the distance of the final syndrome to Bob's syndrome. Thus, the SCA requires L comparisons to find the best codeword. It is clear that to have the same complexity on MS and SCA L should be approximately very close to n^2. For example, for a code of length 10^3 bits, the complexity of the SCA with 10^6 test-vectors would be the same as the

Table 3.2: The complexity per iteration.

	# of additions	# of comparison	# of $\oplus$
MS algorithm	$n \cdot \bar{d}_v$	$m \cdot \bar{d}_c$	$m \cdot \bar{d}_c$
BF algorithm	0	0	$m \cdot \bar{d}_c$

complexity of the MS algorithm.

3.5 Reconciliation for a wide range of SNRs

So far, we have described how to design different reconciliation schemes for CV-QKD for a specific signal to noise ratio. The efficiency of these reconciliation schemes can be defined as the ratio of the net shared information revealed for the reconciliation and the mutual information. The precise definition depends on the reconciliation protocol, for example in RR, in the case of multidimensional reconciliation, the efficiency can be defined as:

$$\begin{aligned}\beta_{\text{multi-dimensional}} &= \frac{R^{\text{Ch}}}{I_{\text{AWGN}}(X_B; X_A)} \\ &= \frac{R^{\text{Ch}}}{I_{\text{BI-AWGN}}(X_B; X_A)} \cdot \frac{I_{\text{BI-AWGN}}(X_B; X_A)}{I_{\text{AWGN}}(X_B; X_A)} \\ &= \beta_{\text{Code}} \cdot \beta_{\text{Mapping}} \ ,\end{aligned}$$

where β_{Code} denotes the efficiency of the error correction code on BI-AWGN channel, and β_{Mapping} denotes the mapping efficiency between the AWGN channel and virtual BI-AWGN channel. On the other hand, the efficiency of the MLC-MSD scheme, in the case of RR, is defined as:

$$\begin{aligned}\beta_{\text{MLC-MSD}} &= \frac{H(\mathcal{Q}(X_B)) - R^{\text{Source}}}{I(X_B; X_A)} \\ &= \frac{H(\mathcal{Q}(X_B)) - R^{\text{Source}}}{I(\mathcal{Q}(X_B); X_A)} \cdot \frac{I(\mathcal{Q}(X_B); X_A)}{I(X_B; X_A)} \\ &= \frac{H(\mathcal{Q}(X_B)) - m + R^{\text{Ch}}}{I(\mathcal{Q}(X_B); X_A)} \cdot \frac{I(\mathcal{Q}(X_B); X_A)}{I(X_B; X_A)} \\ &= \beta_{\text{Code}} \cdot \beta_{\text{Disc}} \ ,\end{aligned}$$

where β_{Code} denotes the code efficiency, and β_{Disc} denotes the digitization efficiency.

The common issue in all the above reconciliation schemes is the fixed code rate. It means that the code rate is designed for a certain SNR value for reliable reconciliation.

Then, by moving forward from this value using the same code, the efficiency moves down from the optimal value. On the other hand, by moving backward from this point, higher efficiency can be obtained, with lower reliability or higher frame error rate. One approach to solve this issue is to use multiple LDPC codes with different code rates to cover a wide SNR range [58].

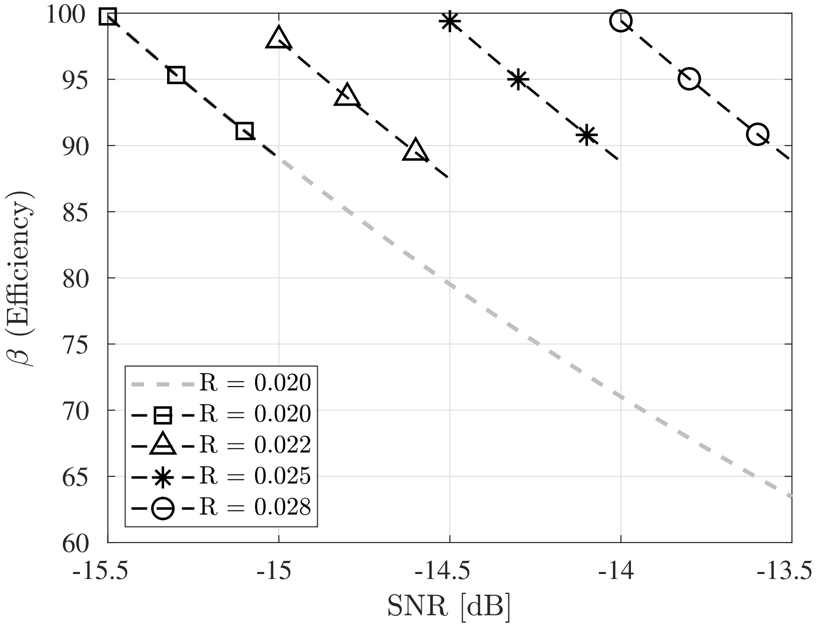


Figure 3.15: Reconciliation efficiencies for different SNRs. The gray line denotes the efficiency when a fixed code of rate 0.02 is used. The other curves denote the cases when other LDPC codes are used for different SNR ranges.

Designing multiple LDPC codes for each SNR is not an efficient solution since a huge parity check matrix is required for each code rate. Two other techniques, widely used to create rate adaptive codes, are *puncturing* and *shortening*. By using these two techniques one can create new code rates from a code with a fixed code rate. For instance, assuming that the original code rate is $R = \frac{k}{n}$, then the rate of the punctured code is

$$R_{\text{Punc}} = \frac{k}{n-p} \,, \tag{3.43}$$

where p denotes the length of the puncturing. Puncturing enables us to increase the code rate. During the puncturing process, p symbols are eliminated from the codeword. The punctured bits are not transmitted, and at the decoder they are replaced by least reliable bits. For example, in the case of a binary erasure channel (BEC) they are considered as erasure and in the case of an AWGN channel they are replaced with zero LLR values.

The second technique is known as shortening and the rate of the shortened code is defined as

$$R_{\text{Short}} = \frac{k-s}{n-s} \ , \tag{3.44}$$

where s denotes the length of the shortened symbols. Figure 3.16 demonstrates the principle of shortening for standard error correction codes.

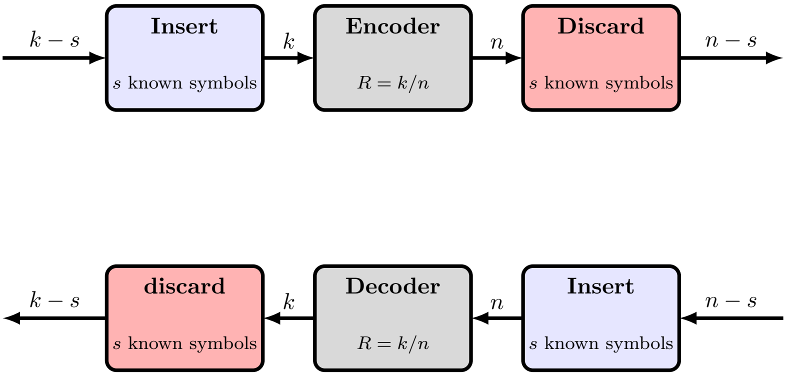


Figure 3.16: The block diagram of the shortening process. The base code has rate $R = \dfrac{k}{n}$ and the shortened code has the rate $R_{\text{Short}} = \dfrac{k-s}{n-s}$.

Rate adaptive techniques were developed for both DV-QKD and CV-QKD, where they combine shortening and puncturing methods to design rate adaptive reconciliation [58, 36]. Recently, a rate adaptive reconciliation technique was developed by using Raptor codes [59], where authors combined a multi-dimensional reconciliation scheme with Raptor codes to develop a rate-less reconciliation protocol. The main issue of this scheme is that the two parties need extra communication to send an additional check code to finish the reconciliation process [60].

CHAPTER 4

On the Design of Highly Efficient MET-LDPC codes

This chapter investigates how to design highly efficient multi-edge-type low-density parity-check (MET-LDPC) codes for CV-QKD. The readers are invited to see our arXived article [18]. It is assumed that readers are familiar with Irregular LDPC codes. For the readers, who are not familiar with this class of error correction codes, more details are provided in Appendix A.

Section 4.1 introduces the MET-LDPC codes and provides all the requirements for understanding the code structure. In Section 4.2, we describe the density evolution (DE) for the MET-LDPC codes and some of its approximations are presented. In Section 4.3, we propose the concept of G-EXIT charts for MET-LDPC codes. In Section 4.4, the code design optimization problem is defined and we talk more about the structure of these codes. Our new proposed optimization algorithm is presented in Section 4.5 and some examples codes designed with the optimization algorithm are presented in Section 4.6. Finally, Section 4.7 provides some simulation results to show the finite size performance of our designed codes.

4.1 Multi-Edge-Type LDPC codes

MET-LDPC codes are a generalization of the concept of irregular LDPC codes [44, 61]. These codes provide improvements in performance and complexity by giving more flexibility over different edge types. In this structure each node is characterized by the number of connections (sockets) to edges of each edge-type. An irregular LDPC code can be considered as a single-edge-type LDPC (SET-LDPC) code in this context. Using MET-LDPC codes we are able to design capacity achieving codes without using VNs with very high degree which provides a less complex implementation. Also, it exploits the advantage of using degree one VNs, which are very useful for designing LDPC codes at low rate and low SNR[61]. It is important to recall that in the case of irregular LDPC codes the minimum usable VN degree is 2.

It is very convenient to begin with a comprehensible *table format representation* of MET-LDPC codes as presented in Table 4.1. The left side of the table corresponds to the VNs and the right side describes the CN structures.

Table 4.1: Table presentation of a MET-LDPC code with n_e edge types.

variable-node							check-node				
$\nu_{\mathbf{bd}}$	$\mathbf{b}$		$\mathbf{d^v}$				$\mu_{\mathbf{d}}$	$\mathbf{d^c}$			
ν_1	$b_{0,1}$	$b_{1,1}$	$d^v_{1,1}$	$d^v_{2,1}$	$\cdots$	$d^v_{n_e,1}$	μ_1	$d^c_{1,1}$	$d^c_{2,1}$	$\cdots$	$d^c_{n_e,1}$
ν_2	$b_{0,2}$	$b_{1,2}$	$d^v_{1,2}$	$d^v_{2,2}$	$\cdots$	$d^v_{n_e,2}$	μ_2	$d^c_{1,2}$	$d^c_{2,2}$	$\cdots$	$d^c_{n_e,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
ν_i	$b_{0,i}$	$b_{1,i}$	$d^v_{1,i}$	$d^v_{2,i}$	$\cdots$	$d^v_{n_e,i}$	μ_i	$d^c_{1,i}$	$d^c_{2,i}$	$\cdots$	$d^c_{n_e,i}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
ν_Q	$b_{0,Q}$	$b_{1,Q}$	$d^v_{1,Q}$	$d^v_{2,Q}$	$\cdots$	$d^v_{n_e,Q}$	μ_P	$d^c_{1,P}$	$d^c_{2,P}$	$\cdots$	$d^c_{n_e,P}$

For instance, the i^{th} row at the CN side corresponds to a CN of type $\mathbf{d_{c,i}} = (d^c_{1,i}, d^c_{2,i}, \cdots, d^c_{n_e,i})$ where each element $d^c_{j,i}$ in $\mathbf{d_{c,i}}$ describes the degree of the CN i along j^{th} edge type. n_e denotes the total number of edge-types. Then, the matrix $\mathbf{d_c} = \begin{pmatrix} \mathbf{d_{c,1}} \\ \vdots \\ \mathbf{d_{c,P}} \end{pmatrix}$ describes the *edge distribution* of the CN side. In addition, the vector $\mu_{\mathbf{d}} = (\mu_1, \ldots, \mu_i, \ldots, \mu_P)$ describes the *node distribution* of the CN side, where P is the number of different CN types. Similarly, $\mathbf{d_v} = \begin{pmatrix} \mathbf{d_{v,1}} \\ \vdots \\ \mathbf{d_{v,Q}} \end{pmatrix}$ describes the edge distribution of the VNs, where each row vector $\mathbf{d_{v,i}} = (d^v_{1,i}, d^v_{2,i}, \cdots, d^v_{n_e,i})$ describes a specific VN type. The node distribution on the VN side is denoted by $\nu_{\mathbf{bd}} = (\nu_1, \ldots, \nu_i, \ldots, \nu_Q)$, where Q is the number of VN types. Finally, the vectors $\mathbf{b_i} = (b_{0,i}, b_{1,i})$ describes the channel to be punctured or not. Here, in this article, we denote a non-punctured node by $\mathbf{b_i} = (0, 1)$. In addition, it is noteworthy to mention that the elements of the vectors $\nu_{\mathbf{bd}}$ and $\mu_{\mathbf{d}}$ are non-negative fractional numbers and the elements of the matrices $\mathbf{d_v}$ and $\mathbf{d_c}$ are non-negative integer numbers. For example, let N be the length of the code-word, then for each CN of type i the quantity $\mu_i N$ is the number of constraint-nodes of type i in the graph. Similarly, the quantity $\nu_j N$ is the number of VNs of type j in the graph.

Furthermore, the code ensemble for a MET-LDPC code can be specified by two multi-variable-polynomials, one associated to VNs and the other associated to CNs (constraint-nodes). The *node-perspective* representation of these multi-variable-polynomials are:

$$\nu(\mathbf{r}, \mathbf{x}) = \sum_{i=1}^{Q} \nu_i \mathbf{r}^{\mathbf{b_i}} \mathbf{x}^{\mathbf{d_{v,i}}}, \quad \mu(\mathbf{x}) = \sum_{i=1}^{P} \mu_i \mathbf{x}^{\mathbf{d_{c,i}}} \ , \tag{4.1}$$

respectively. In (4.1), $\mathbf{x} := (x_1, \cdots, x_{n_e})$ and $\mathbf{r} := (r_0, \cdots, r_{n_r})$, where n_r is the number of different channels. In this thesis it is assumed that, $n_r = 2$. r_0 stands for punctured channel and r_1 is the channel parameter for te code. Finally, $\mathbf{x}^{\mathbf{d_{c,i}}} := \prod_{j=1}^{n_e} x_j^{d^c_{j,i}}$, $\mathbf{x}^{\mathbf{d_{v,i}}} := \prod_{j=1}^{n_e} x_j^{d^v_{j,i}}$ and $\mathbf{r}^{\mathbf{b_i}} := \prod_{j=1}^{n_r} r_j^{b_{j,i}}$.

In addition, the *edge perspective* degree distribution can be described as a vector of multi-variable polynomials, for VNs and CNs, respectively,

$$\begin{aligned}\lambda(\mathbf{r},\mathbf{x}) &= \left(\frac{\nu_{x_1}(\mathbf{r},\mathbf{x})}{\nu_{x_1}(\mathbb{1},\mathbb{1})}, \frac{\nu_{x_2}(\mathbf{r},\mathbf{x})}{\nu_{x_2}(\mathbb{1},\mathbb{1})}, \dots, \frac{\nu_{x_{n_e}}(\mathbf{r},\mathbf{x})}{\nu_{x_{n_e}}(\mathbb{1},\mathbb{1})}\right) \ , \\ \rho(\mathbf{x}) &= \left(\frac{\mu_{x_1}(\mathbf{x})}{\mu_{x_1}(\mathbb{1})}, \frac{\mu_{x_2}(\mathbf{x})}{\mu_{x_2}(\mathbb{1})}, \dots, \frac{\mu_{x_{n_e}}(\mathbf{x})}{\mu_{x_{n_e}}(\mathbb{1})}\right) \ , \end{aligned} \tag{4.2}$$

where,

$$\nu_{x_i}(\mathbf{r},\mathbf{x}) = \frac{\partial}{\partial x_i}\nu(\mathbf{r},\mathbf{x}) \ , \qquad \mu_{x_i}(\mathbf{x}) = \frac{\partial}{\partial x_i}\mu(\mathbf{x}) \ ,$$

and $\mathbb{1}$ denotes a vector of all 1's where the length being determined by context. The coefficients of ν and μ are constrained to ensure that the number of sockets of each type is the same on both sides (variable and check) of the graph. This gives rise to n_e linear conditions on the coefficients of ν and μ as follows:

$$\nu_{x_i}(\mathbb{1},\mathbb{1}) = \mu_{x_i}(\mathbb{1}), \ i = 1, \cdots, n_e \ .$$

Finally, the nominal code rate for non-punctured codes is given by

$$R = \nu(\mathbb{1},\mathbb{1}) - \mu(\mathbb{1}) \ .$$

Example 4.1.1. *A MET-LDPC code with rate* 0.02

Consider a MET-LDPC code ensemble with rate 0.02, with the following structure as presented in Table 4.2:

Table 4.2: Rate 0.02 MET-LDPC code with 3 edge types.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$	$\mathbf{d}$	$\mu_{\mathbf{d}}$	$\mathbf{d}$
0.02	[0 1]	2 51 0	0.016	4 0 0
0.02		3 60 0	0.004	9 0 0
0.96		0 0 1	0.30	0 3 1
			0.66	0 2 1

BI-AWGN: $\sigma^*_{\text{DE}} = 5.94$

The corresponding polynomial representation for this code is:

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.02\, r_1 x_1^2 x_2^{51} + 0.02\, r_1 x_1^3 x_2^{60} + 0.96\, r_1 x_3 \ , \\ \mu(\mathbf{x}) &= 0.016\, x_1^4 + \ 0.004\, x_1^9 + \ 0.30\, x_2^3 x_3^1 + \ 0.66\, x_2^2 x_3^1 \ .\end{aligned}$$

The code has $n_e = 3$ edge types, $Q = 3$ types of VNs and $P = 4$ types of CNs. The corresponding Tanner graph, for this code is shown in Figure 4.1.

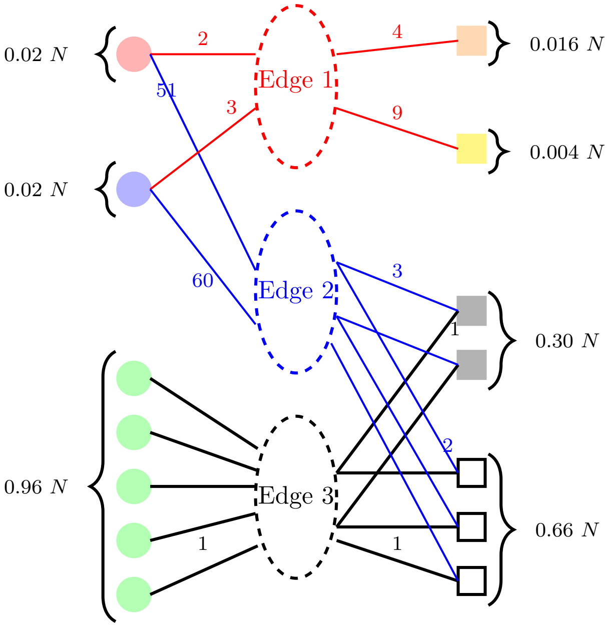


Figure 4.1: Graphical representation of a three-edge type-LDPC code presented in Table 4.2, where ○ represents the VNs and □ represents the CNs. In addition different node types have different colors. The percentage of node types are shown as fractions of the code length N, where N is the number of transmitted code-word bits. This code consists of 3 different types of VNs and 4 types of CNs. It is composed of 3 different edge-types.

4.1.1 MET-LDPC codes with the cascade structure

In this subsection we exploit the advantages of the *cascade structure* [61] for MET-LDPC codes. First, we introduce the structure and then an optimization method to design highly efficient cascade structures. The main advantage of the cascade structure is its simple edge distribution, which means that many of the elements of the matrices $\mathbf{d^v}$ and $\mathbf{d^c}$ are zero. This useful feature provides considerable reduction in search space when designing degree distributions. An example of schematic representation of the cascade structure for the MET-LDPC codes is depicted in Figure 4.1. It is clear that the structure of the whole graph is a combination of three connected sub-graphs. The first part belongs to an irregular *base code* (red color). The second part belongs to VNs and CNs of degree one (black color). Finally, the last part, which we call *connector part*, connects these two sub-graphs to each other (blue color). The corresponding parity check matrix for this code has the following structure

$$\mathbf{H} = \begin{pmatrix} \mathbf{I} & \mathbf{C} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} . \tag{4.3}$$

These three disjoint sub-matrices should be optimized to generate the overall code with rate R. The sub-matrix $\mathbf{B}$, is related to an irregular LDPC code with rate r_b (base code). The sub-matrix $\mathbf{I}$ corresponds to degree one VNs and CNs and the sub-matrix $\mathbf{C}$ corresponds to the connection part which connects the base code to nodes with degree one. The connector part could contain just one edge type (as presented on Figure 4.1) or it can contains more than one edge type (Fig. 5.a [14]). Thus, this class of codes can contain different edge numbers.

Now, consider a cascade structure with $n_e = 3$ edge types. The overall degree distribution and its table format representation are as follows:

$$\nu(r, \mathbf{x}) = \sum_{i=1}^{Q-1} \nu_i \; x_1^{d^v_{1,i}} x_2^{d^v_{2,i}} + \nu_3 \; x_3 \; ,$$

$$\mu(\mathbf{x}) = \sum_{i=1}^{P-2} \mu_i \; x_1^{d^c_{1,i}} + \mu_{P-1} \; x_2^{d^c_{2,P-1}} x_3 + \mu_P x_2^{d^c_{2,P}} x_3 \; .$$

Table 4.3: MET-LDPC code with cascade structure and 3 edge types.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$		$\mathbf{d^v}$			$\mu_{\mathbf{d}}$	$\mathbf{d^c}$		
ν_1	0	1	$d^v_{1,1}$	$d^v_{2,1}$	0	μ_1	$d^c_{1,1}$	0	0
ν_2	0	1	$d^v_{1,2}$	$d^v_{2,2}$	0	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	μ_{P-2}	$d^c_{1,P-2}$	0	0
ν_{Q-1}	0	1	$d^v_{1,Q-1}$	$d^v_{2,Q-1}$	0	μ_{P-1}	0	$d^c_{2,P-1}$	1
ν_Q	0	1	0	0	1	μ_P	0	$d^c_{2,P}$	1

Each column in the $\mathbf{d^v}$ and $\mathbf{d^c}$ matrices corresponds to one of the 3 different edge types highlighted by different colors. For instance, the edge-type one corresponds to the sub-matrix $\mathbf{B}$ and is presented in column one (red color), the edge-type two corresponds to sub-matrix $\mathbf{C}$ and is presented in column two (blue color) and the sub-matrix $\mathbf{I}$ corresponds to the edge-type three and degree one nodes (black color). The schematic of the Tanner graph of the cascade code is similar to Figure 4.1. In Example 4.1.1, it is assumed that the VN and CN degree distributions have three and four terms, but in general they could have Q and P terms.

4.2 Density evolution and other asymptotic analysis tools

Now, let us describe how the belief-propagation (BP) algorithm works in the MET framework, and how the intermediate densities are defined. Assume that $m^l_{vc(e)}$ (respectively, $m^l_{cv(e)}$) denotes the LLR message from VN to CN (respectively, CN to VN) along edge-type e at iteration l. Similarly, assume that, $f(m^l_{vc(e)})$ (respectively, $f(m^l_{cv(e)})$) denotes the probability density function (PDF) of the messages from VN to CN along edge-type e, (respectively, CN to VN), and $f(m^0)$ denotes the PDF of channel LLR. Then the PDF at the output of a VN of type $\mathbf{d_{v,i}}$ along edge-type e is:

$$f(m^l_{vc(e)}) = f(m^0) \otimes \left[f(m^{l-1}_{cv(e)})\right]^{\otimes(d^v_{e,i}-1)} \bigotimes_{j=1,j\neq e}^{n_e} \left[f(m^{l-1}_{cv(j)})\right]^{\otimes d^v_{j,i}} , \tag{4.4}$$

where just messages which belong to specific edge-type e are assumed to be independent and identically distributed. Similarly, the PDF of the message at the output of a CN of type $\mathbf{d_{c,i}}$ along edge-type e is equal to

$$f(m^l_{cv(e)}) = \left[f(m^l_{vc(e)})\right]^{\boxtimes(d^c_{e,i}-1)} \mathop{\boxtimes}_{j=1,j\neq e}^{n_e} \left[f(m^l_{vc(j)})\right]^{\boxtimes d^c_{j,i}} . \tag{4.5}$$

Also, we use $\otimes$ and $\boxtimes$ to denote the convolution for VNs and CNs respectively. More details about the BP algorithm for MET-LDPC codes can be found in [44, 61, 13].

In addition, assume that, $\vec{\mathsf{a}}^l_v = (\mathsf{a}^l_{v,1}, \cdots, \mathsf{a}^l_{v,n_e})$[1] and $\vec{\mathsf{a}}^l_c = (\mathsf{a}^l_{c,1}, \cdots, \mathsf{a}^l_{c,n_e})$ denote the vector of L-densities at the output of VNs and CNs after $l \geq 1$ iterations. Then, the density evolution (DE) for the L-densities can be written as:

$$\vec{\mathsf{a}}^{l+1}_v = \lambda(\vec{\mathsf{a}}_{\mathrm{BIOSMC}}, \vec{\mathsf{a}}^{l+1}_c) , \tag{4.6}$$
$$\vec{\mathsf{a}}^{l+1}_c = \rho(\vec{\mathsf{a}}^l_v) , \tag{4.7}$$

where, $\rho(\mathbf{x})$ and $\lambda(\mathbf{r},\mathbf{x})$ are presented in (4.2) and $\vec{\mathsf{a}}_{\mathrm{BIOSMC}}$ is the L-density of the binary-input output symmetric memory-less (BIOSM) channel. The initial L-density vector is $\vec{\mathsf{a}}^0_v = \vec{\mathbf{\Delta}}_0$. $\vec{\mathbf{\Delta}}_0$ is a vector of densities, where each density is δ_0, which is equivalent to erasure with probability 1. In the following some approximation methods for density evolution for MET-LDPC codes are discussed, including:

1. Gaussian approximation,
2. Hybrid-density evolution,
3. Semi-Gaussian approximation,
4. Generalized extrinsic-information transfer (G-EXIT) chart.

[1] $\vec{\mathsf{a}}^l_v$ denotes a vector of L-densities and $\mathsf{a}^l_{v,i}$ corresponds to the L-density along the i^{th} edge. The deffinition of the L-densities can be found in Appendix A.

4.2.1 The Gaussian approximation

The Gaussian assumption is the easiest approximation (in terms of computational complexity) for the density evolution (DE). Here we discuss about the accuracy of this approximation of DE. In [44], it has been shown that for irregular-LDPC codes for a BI-AWGN channel the channel densities CN to VN and VN to CN can be estimated by symmetric Gaussian distribution. This approximation is valid because the intermediate densities during the belief propagation algorithm remain symmetric [44]. As long as the CN degrees are small, the VN degrees are large and the code rate is in typical range, the Gaussian approximation for irregular-LDPC codes can be accurate [13]. Unfortunately most of above mentioned assumptions are not satisfied in the case of MET-LDPC codes, where degree one VNs exist and the codes are designed for applications with low code rate and very low SNR. To see if Gaussian assumption remains accurate during the BP, we have to consider three types of densities including the channel density, the density of messages at the output of VNs and at the output of CNs. In the following these three types of densities are discussed separately.

4.2.1.1 Accuracy of Gaussian assumption for channel densities

First, let us discuss the channel densities. Assume that a binary codeword $C = (c_1, \cdots, c_n)$ is transmitted on a BI-AWGN channel with Binary Phase Shift Keying (BPSK) transmission ($0 \rightarrow +1,$ and $1 \rightarrow -1$). The received symbol can be modeled as $y_i = x_i + n_i$, where $x_i \in \{\pm 1\}$, and n_i is white Gaussian noise with zero mean and variance σ_n^2. In the LLR domain we can write for the a-posteriori probability:

$$m_0 = \ln\left(\frac{\Pr(x_i = +1|y_i)}{\Pr(x_i = -1|y_i)}\right) = \ln\left(\frac{\Pr(x_i = +1)}{\Pr(x_i = -1)}\right) + \ln\left(\frac{\Pr(y_i|x_i = +1)}{\Pr(y_i|x_i = -1)}\right) ,$$

where for equally likely inputs it can be simplified to:

$$m_0 = \ln\left(\frac{\Pr(x_i = +1|y_i)}{\Pr(x_i = -1|y_i)}\right) = \ln\left(\frac{\Pr(y_i|x_i = +1)}{\Pr(y_i|x_i = -1)}\right) = \frac{2}{\sigma_n^2} y_i .$$

Assuming that the all-zero codeword is sent, the channel LLR for MET-LDPC codes is a Gaussian random variable with mean $2/\sigma_n^2$ and variance $4/\sigma_n^2$. Thus, the symmetric Gaussian assumption is valid for channel densities in the LLR domain.

4.2.1.2 Accuracy of Gaussian assumption for variable-node densities

Now, let us discuss the validity of the Gaussian approximation at the output of the VNs. According to (4.4), and the VN operation during the BP algorithm, the PDF of the message from a VN to CN along edge e is equal to the convolution of PDFs for m_0 and independent messages from its neighbor CNs. Since the channel density is Gaussian the output density is Gaussian if and only if all the other independent incoming messages from the other CNs are Gaussian. On the other hand, if the incoming messages from CNs are not Gaussian it can be observed that as long as the number of terms are large enough (for VNs with high degree) the output density can be approximated by a Gaussian distribution, which is a consequence of the central limit theorem.

4.2.1.3 Accuracy of Gaussian assumption for check-node densities

Finally, consider the Gaussian approximation for the densities at the output of CNs. It has been shown in [62] that by treating the signs and magnitudes of the LLRs separately, the CN update rule can be expressed as:

$$m^l_{cv(e)} = \prod_{j \neq e} \operatorname{sign}\left(m^l_{vc(j)}\right) \ \psi\left(\sum_{j \neq e} \psi\left(\left|m^l_{vc(j)}\right|\right)\right) \ , \tag{4.8}$$

where $|\cdot|$ denotes the magnitude of the LLRs which determines the certainty of the message. The function $\psi(\cdot)$ is a decreasing function for positive real numbers with self inverse as follows:

$$\psi(x) = \psi^{-1}(x) = \ln\left(\frac{e^x + 1}{e^x - 1}\right) \ .$$

Now assume that in (4.8) just one of the inputs has LLR close to zero, then ψ would be close to ∞ and that specific term would be dominant in the summation. It follows that the final value of (4.8) would be 0. On the other hand, assuming a high value for input LLR, ψ would be close to zero and thus it does not have significant effect on the summation of the LLRs. Thus one can interpret the CN operation as a *soft-min* operation. It means that there is no guaranty that the output of the CN would be Gaussian.

Example 4.2.1. *Inaccuracy at CNs*

As an example, in Figure 4.2 we show the Kullback-Leibler (KL) divergence [63] between CN to VN messages and their corresponding symmetric Gaussian PDFs at $\frac{E_b}{N_0} = -1.0$ dB for rate 0.1 MET-LDPC code with the following degree distribution,

$$\begin{aligned} \nu(\mathbf{r}, \mathbf{x}) &= 0.1 \ r_1 x_1^3 x_2^{20} + 0.0025 \ r_1 x_1^3 x_2^{25} + 0.875 \ r_1 x_3 \ , \\ \mu(\mathbf{x}) &= 0.025 \ x_1^{15} + 0.875 \ x_2^3 x_3^1 \ . \end{aligned}$$

Small value for the KL divergence means that the two PDFs are similar to each other, and large values means that the divergence is high. As depicted in Figure 4.2 the Gaussian approximation is only accurate after many decoding iterations. Specifically, for edge-type one, which has a CN of degree 15, the Gaussian approximation does not follow a predictable pattern.

It is also possible to see the effect of SNR and number of iteration on the Gaussian approximation. Specifically, as depicted in Figure 4.3 and Figure 4.4, the KL

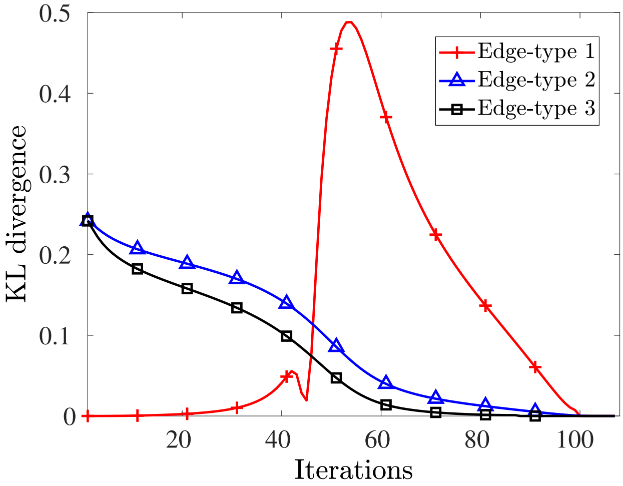


Figure 4.2: The KL divergence between messages from CN to VN and their equivalent symmetric Gaussian PDFs for all edge-types for rate 0.1 MET-LDPC code at $\frac{E_b}{N_0} = -1.00$ dB. The iteration number increased gradually until we achieve error probability less than 10^{-11}.

divergence has smaller values only at high SNRs and after some decoding iterations, which shows that the validity of the Gaussian approximation is not assured at low SNRs. Also it is clear that at very low SNRs the KL divergence can not reach zero, even after many iterations. Besides, even at high SNRs, at the very beginnings the Gaussian approximation is not accurate.

4.2.2 Hybrid density evolution

As discussed in Section 4.2.1, the Gaussian approximation is not accurate because it cannot be assured that the PDF at the output of CNs can be estimated by a symmetric Gaussian distribution. According to our the simulation results it is clear that the Gaussian approximation is not accurate at very low SNRs and for MET-LDPC codes with large CN degrees. Even if we use a Gaussian approximation and wish to obtain an accurate Gaussian estimation at later iteration, it is not an accurate estimation for the true distribution of the messages. The reason is that the Gaussian assumption at very early iterations will cause an unexpected error between the Gaussian estimation and the expected true distribution. Thus using the Gaussian approximation for designing MET-LDPC codes would not provide highly efficient codes.

In [13] the authors introduced a new approximation method called hybrid density evolution (hybrid-DE). In their proposed algorithm they used a combination of the

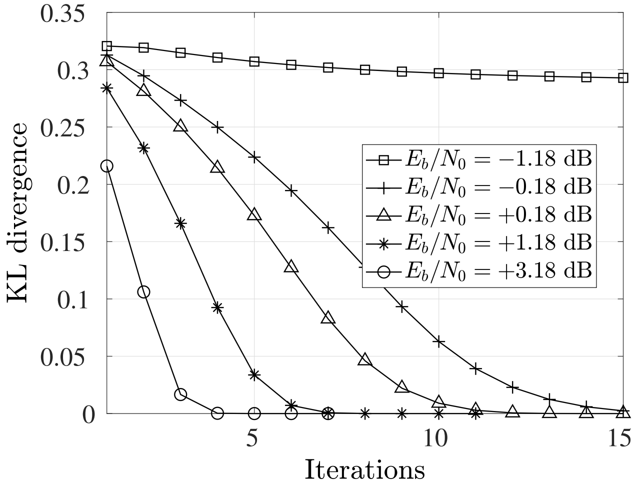


Figure 4.3: The KL divergence between output PDF of the CN to VN for edge-type two of rate 0.1 MET-LDPC code and the symmetric Gaussian PDF of the same mean. The iteration number increased gradually until we achieve error probability less than 10^{-11}.

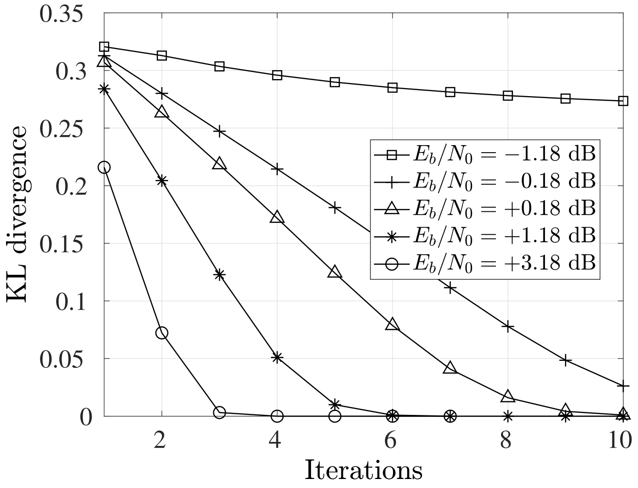


Figure 4.4: The KL divergence between output PDF of the CN to VN for edge-type three of rate 0.1 MET-LDPC code and the symmetric Gaussian PDF of the same mean. The iteration number increased gradually until we achieve error probability less than 10^{-11}.

full density evolution and a Gaussian approximation. The reason is that making the assumption of a symmetric Gaussian distribution at later decoding iterations is reasonable (See Figure 4.2). The hybrid-DE starts with the full density evolution and then switches to the Gaussian approximation. In addition, they introduced two hard and soft switching approaches. In the hard approach a fixed value was assumed for the maximum number of iterations, and in soft method the KL divergence was used to decide when the output PDFs can be estimated by a symmetric Gaussian distribution. If $\alpha \in [0, 1]$ denotes the portion of algorithm when full DE is used, then $(1 - \alpha)$ of the algorithm would be based on Gaussian approximation. A large value of α means that the complexity of the hybrid-DE would be similar to full-DE and a small value of α means that the complexity of algorithm should be similar to the Gaussian approximation method. When working at very low SNRs and for codes with high CN degrees which is the case of interest for quantum key distribution, the value of α should be high in order to get an accurate threshold estimation.

4.2.3 Semi-Gaussian approximation

The idea of Semi-Gaussian approximation was first introduced in [64] for irregular LDPC codes. Here we present a modified version of this algorithm for MET-LDPC codes on BI-AWGN channel. This analysis tool is significantly more accurate than the conventional Gaussian approximation. For simplicity assume that there are no punctured VN in the MET-LDPC code. Thus, as discussed in Section 4.2.1.2 the assumption of a symmetric Gaussian approximation for the PDFs at the output of VNs is always accurate and valid. This was also shown in [13] and was confirmed by our simulation results. For the CNs the true CN operation is used to find the output densities and we assume that the input densities are Gaussian. Since the Gaussian assumption is just used on the VN side, this algorithm is called the *Semi-Gaussian* approximation. The key idea in Semi-Gaussian approximation is that a Gaussian approximation is not used for the densities at the output of CNs but only for messages from VNs to CNs.

Table 4.4: Average number of operations per iteration. The average degree of VN is denoted by $\bar{d}_v$ and the average degree of CN is $\bar{d}_c$.

	Full-DE		**Gaussian**		**Hybrid-DE**		**Semi-Gaussian**	
	VN	**CN**	**VN**	**CN**	**VN**	**CN**	**VN**	**CN**
Sums	-	-	$\bar{d}_v$	$\bar{d}_c$	$(1-\alpha)\ \bar{d}_v$	$(1-\alpha)\ \bar{d}_c$	$\bar{d}_v$	-
Lookup-tables	-	-	-	$\bar{d}_c$	-	$(1-\alpha)\ \bar{d}_c$	-	-
Exponentials	-	-	-	$\bar{d}_c$-1	-	$(1-\alpha)\ (\bar{d}_c-1)$	-	-
Convolution	$\bar{d}_v$	$\bar{d}_c - 1$	-	-	$\alpha\ \bar{d}_v$	$\alpha\ (\bar{d}_c-1)$	-	$\bar{d}_c - 1$

The semi-Gaussian approach starts with a symmetric Gaussian distribution at the input of CNs and calculates the output densities using a single step density evolution. Then for the next iteration the output density is estimated by a symmetric Gaussian PDF. Thus, there is no need to convolve the input messages to calculate the

PDF at the output of VN. Table 4.4 compares the number of average operations per decoding iteration for density evolution and some approximation algorithms including Gaussian approximation, Hybrid-DE and our semi-Gaussian algorithm. The number of operations are compared for one iteration. The full-DE requires convolution at both VNs and CNs. The results for hybrid-DE with parameter α is taken from [13].

The overall complexity of the semi-Gaussian algorithm is significantly less than Hybrid-DE and density evolution. In density evolution and hybrid-DE we need to wait for the PDF of former iterations in order to analyze the next iteration. In contrast, for the semi-Gaussian approximation it is possible to start from any arbitrary point. This amazing fact gives us the possibility to use the semi-Gaussian approach for calculation of EXIT or G-EXIT chart as introduced later in this chapter. One can calculate some points of the G-EXIT chart and interpolate the remaining points. It is also possible to focus on points very close to the threshold to increase the number of analysis points. Simulation results show that our proposed method provides an accurate estimate of convergence behavior and threshold of the MET-LDPC codes.

4.2.4 Generalized Extrinsic-Information Transfer (G-EXIT) Charts

The EXIT charts [65] and G-EXIT charts [66, 67] play a remarkable role in analysis and design of LDPC codes [67]. For the sake of simplicity detailed definitions of the EXIT and G-EXIT functions are provided in Appendix A. In addition, for a detailed review on this analysis tool for irregular LDPC codes see [44, 66, 68, 69].

4.3 G-EXIT charts for MET-LDPC codes

The EXIT and G-EXIT charts are widely used to design irregular LDPC codes [65, 67]. Despite their powerful advantages this tool has not been used to design MET-LDPC codes and the characteristics of these tools for designing MET-LDPC codes are not well understood. In this section, we first review the concept of G-EXIT charts for a simple irregular LDPC code by an example. Then, we describe our novel method to introduce and plot the G-EXIT charts for the MET-LDPC codes. For simplicity the BE channel is considered here, because the EXIT charts and G-EXIT charts are equivalent in the BE channel [66, 67]. However all the results can be extended to other BIOSM channels (For more details about different BIOSM channels see Appendix A).

Consider an irregular LDPC code (the definition of the irregular LDPC codes and their degree distribution can be found in Appendix A) with edge perspective degree distribution $\lambda(x)$ and $\rho(x)$. On a binary erasure channel with parameter q (BEC(q)), the density evolution can be described as:

$$\epsilon_v^{l+1} = q \ \lambda \left(1 - \rho\left(1 - \epsilon_v^l\right)\right) \ , \tag{4.9}$$

where ϵ_v^l denotes the erasure probability after l iterations at the output of VN. The iterative density evolution in (4.9) means that the erasure probability at the output of

VN is considered as the erasure probability at the input of CN for the next iteration. This iterative equation can be represented as:

$$\epsilon_v^l = q\ \lambda(\epsilon_c^{l-1})\ , \qquad \epsilon_c^l = 1 - \rho(1 - \epsilon_v^l)\ ,$$

where $\epsilon_c^0 = 1$.

To plot the EXIT (G-EXIT) curves one can use the parametric form of the EXIT curves for VNs and CNs as presented in Table A.2. In the case of BEC this can be simplified because $\mathtt{h} = H(\epsilon) = \epsilon$. Thus, the parametric form of the CN EXIT curve is $\{\epsilon_v^l, \epsilon_c^l = \mathcal{F}(\epsilon_v^l)\}$ and the parametric form for the inverse of the VN EXIT curve is $\{\epsilon_v^l = \mathcal{G}(\epsilon_c^{l-1}), \epsilon_c^{l-1}\}$. The EXIT (G-EXIT) function for the CN is denoted by $\mathcal{F}(\cdot)$, and $\mathcal{G}(\cdot)$ denotes the EXIT (G-EXIT) function for the VN. It can be shown that for the BEC(q), the EXIT functions for the VN and CN are

$$\mathcal{G}(x) = q\lambda(x)\ , \tag{4.10}$$

$$\mathcal{F}(x) = 1 - \rho(1 - x)\ . \tag{4.11}$$

In addition, the convergence behavior of this code can be simply explained by the G-EXIT chart on a BE channel. One can plot two curves as presented in Table A.2. In Example 4.3.1 we plotted the G-EXIT chart for an irregular-LDPC code on a BE channel.

Example 4.3.1. *G-EXIT chart for irregular-LDPC codes on BEC*

Consider a simple irregular LDPC code with node perspective degree distributions

$$\nu(q, x) = \ 0.5\ q\ x^2 + \ 0.5\ q\ x^3\ , \qquad \mu(x) = 0.4\ x^4 + \ 0.1\ x^9\ .$$

The corresponding edge perspective degree distribution is

$$\lambda(q, x) = \ 0.4\ q\ x + \ 0.6\ q\ x^2\ , \qquad \rho(x) = 0.64\ x^3 + \ 0.36\ x^8\ .$$

The G-EXIT chart for this code on BEC(q) is shown in Figure 4.5, where q takes three different values $q \in \{0.2, 0.4, 0.6\}$. The threshold of this code on BEC is $q^* = 0.4$. For channel parameters $q \leq q^*$ the two curves do not cross, which determines that the density evolution converges with error probability goes to zero, but for $q > q^*$ the two curves cross each other which means that the density evolution stops at a non-zero error probability.

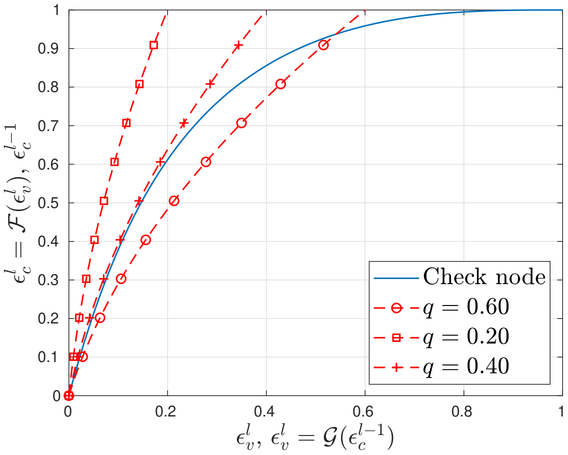


Figure 4.5: The dashed lines are for VN with different channel parameters and the solid line is for the CN. The curves belong to a code with rate 0.5 with maximum VN degree 3 and maximum CN degree 9.

4.3.1 MET-LDPC codes on BEC

Consider a MET-LDPC code with cascade structure and $n_e = 3$ edge types. The degree distribution of this code on a binary erasure channel with parameter q is:

$$\nu(q, \mathbf{x}) = q \sum_{i=1}^{Q-1} \nu_i \; x_1^{d_{1,i}^v} x_2^{d_{2,i}^v} + q \; \nu_3 \; x_3 \; ,$$

$$\mu(\mathbf{x}) = \sum_{i=1}^{P-2} \mu_i \; x_1^{d_{1,i}^c} + \mu_{P-1} \; x_2^{d_{2,P-1}^c} x_3 + \mu_P \; x_2^{d_{2,P}^c} x_3 \; .$$

To analyze the density evolution for MET-LDPC codes the edge perspective degree distribution is required. For MET-LDPC codes the edge perspective distribution can be obtained from (4.2). For the VNs

$$\lambda(\mathbf{q}, \mathbf{x}) = \left(\lambda^1(q, \mathbf{x}), \lambda^2(q, \mathbf{x}), \lambda^3(q, \mathbf{x})\right) \; ,$$

where

$$\lambda^1(q,\mathbf{x}) = q \; \frac{\sum\limits_{i=1}^{Q-1} \nu_i \; d^v_{1,i} \; x_1^{d^v_{1,i}-1} x_2^{d^v_{2,i}}}{\sum\limits_{i=1}^{Q-1} \nu_i \; d^v_{1,i}} = \lambda^1(q,x_1,x_2) \ ,$$

$$\lambda^2(q,\mathbf{x}) = q \; \frac{\sum\limits_{i=1}^{Q-1} \nu_i \; d^v_{2,i} \; x_1^{d^v_{1,i}} x_2^{d^v_{2,i}-1}}{\sum\limits_{i=1}^{Q-1} \nu_i \; d^v_{2,i}} = \lambda^2(q,x_1,x_2) \ ,$$

$$\lambda^3(q,\mathbf{x}) = q = \lambda^3(q) \ .$$

For the CNs

$$\rho(\mathbf{x}) = \left(\rho^1(\mathbf{x}), \rho^3(\mathbf{x}), \rho^3(\mathbf{x})\right) \ ,$$

where

$$\rho^1(\mathbf{x}) = \frac{\sum\limits_{i=1}^{P-2} \mu_i \; d^c_{1,i} \; x_1^{d^c_{1,i}-1}}{\sum\limits_{i=1}^{P-2} \mu_i \; d^c_{1,i}} = \rho^1(x_1) \ ,$$

$$\begin{aligned}\rho^2(\mathbf{x}) &= \frac{\mu_{P-1} \; d^c_{2,P-1} \; x_2^{d^c_{2,P-1}-1} x_3 + \mu_P \; d^c_{2,P} \; x_2^{d^c_{2,P}-1} x_3}{\mu_{P-1} \; d^c_{2,P-1} + \mu_P \; d^c_{2,P}} \\ &= \rho^2(x_2,x_3) \ , \\ \rho^3(\mathbf{x}) &= \frac{\mu_{P-1} \; x_2^{d^c_{2,P-1}} + \mu_P \; x_2^{d^c_{2,P}}}{\mu_{P-1} + \mu_P} = \rho^3(x_2) \ .\end{aligned}$$

To investigate the convergence behavior of this code the vectors of erasure probabilities at the output of VNs and CNs along different edge types has to be considered, which is denoted by $\vec{\epsilon^l_v} = [\epsilon^l_{v,1}, \epsilon^l_{v,2}, \epsilon^l_{v,3}]$ and $\vec{\epsilon^l_c} = [\epsilon^l_{c,1}, \epsilon^l_{c,2}, \epsilon^l_{c,3}]$. Then, at iteration l, the update rule for the VNs are:

$$\begin{aligned}\epsilon^l_{v,1} &= \mathcal{G}^1(q, \epsilon^{l-1}_{c,1}, \epsilon^{l-1}_{c,2}) = \lambda^1(q, \epsilon^{l-1}_{c,1}, \epsilon^{l-1}_{c,2}) \ , \\ \epsilon^l_{v,2} &= \mathcal{G}^2(q, \epsilon^{l-1}_{c,1}, \epsilon^{l-1}_{c,2}) = \lambda^2(q, \epsilon^{l-1}_{c,1}, \epsilon^{l-1}_{c,2}) \ , \\ \epsilon^l_{v,3} &= \mathcal{G}^3(q) = q \ .\end{aligned}$$

For the CN the update rules are:

$$\begin{aligned}\epsilon^l_{c,1} &= \mathcal{F}^1(\epsilon^l_{v,1}) = 1 - \rho^1(1-\epsilon^l_{v,1}) \ , \\ \epsilon^l_{c,2} &= \mathcal{F}^2(\epsilon^l_{v,2}, \epsilon^l_{v,3}) = 1 - \rho^2(1-\epsilon^l_{v,2}, 1-\epsilon^l_{v,3}) \ , \\ \epsilon^l_{c,3} &= \mathcal{F}^3(\epsilon^l_{v,2}) = 1 - \rho^3(1-\epsilon^l_{v,2}) \ .\end{aligned}$$

The initial erasure values at the starting point are $\vec{\epsilon_c^0} = [1,\ 1,\ 1]$. Table 4.5 shows the parametric form of the G-EXIT curves for individual edges. For edge type 1 the CN curve ($\mathcal{F}^1$) is a function of erasure probability at the output of VN on edge 1 ($\epsilon_{v,1}^l$). The CN curve along edge type 2, ($\mathcal{F}^2$), is a function of the erasure probabilities at the output of VNs along edge types 2 and 3 ($\epsilon_{v,2}^l$ and $\epsilon_{v,3}^l$). We can be simplify this because $\epsilon_{v,3}^l = q$ and thus, $\mathcal{F}^2$ is a function of $\epsilon_{v,2}^l$, and q. Finally, for edge type 3 the G-EXIT curve is a vertical line at point $\epsilon_{v,3}^l = q$.

On the other hand, the VN curve along edge type 1 ($\mathcal{G}^1$) and edge type 2 ($\mathcal{G}^2$) are both functions of erasure probabilities at the output of CNs along edges 1 and 2 ($\epsilon_{c,1}^l$ and $\epsilon_{c,2}^l$). For edge type 3 just after one iteration all the erasure probabilities reach a fixed value q and the two curves for VNs and CNs are two completely matching vertical lines.

Table 4.5: The parametric representation of EXIT (G-EXIT) charts for MET-LDPC codes on BEC(q).

	EXIT curve of CN	Inverse of EXIT curve of VN
Edge 1	$\{\epsilon_{v,1}^l,\ \mathcal{F}^1(\epsilon_{v,1}^l)\}$	$\{\mathcal{G}^1(q, \epsilon_{c,1}^{l-1}, \epsilon_{c,2}^{l-1}),\ \epsilon_{c,1}^{l-1}\}$
Edge 2	$\{\epsilon_{v,2}^l,\ \mathcal{F}^2(\epsilon_{v,2}^l, q)\}$	$\{\mathcal{G}^2(q, \epsilon_{c,1}^{l-1}, \epsilon_{c,2}^{l-1}),\ \epsilon_{c,2}^{l-1}\}$
Edge 3	$\{\epsilon_{v,3}^l,\ \mathcal{F}^3(\epsilon_{v,2}^l)\}$	$\{\mathcal{G}^3(q),\ \epsilon_{c,3}^{l-1}\}$

As an example, consider the MET-LDPC code with cascade structure represented in Table 4.2. The corresponding G-EXIT charts on a BE channel are presented in Figure 4.6.

In addition, the overall convergence of the code depends on the combination of the densities along different edge types. For example, the convergence behavior of this code is depicted in Figure 4.7 for different channel parameters. It is clear that the threshold of this code on BEC is $q^* = 0.97505$. For the channel parameters $q > q^*$, the code does not converge and for the $q < q^*$ the code converges to zero error probability after 200 iterations.

4.3.2 MET-LDPC codes on a BIOSM channel

As discussed above, in the case of a BEC we are able to plot the G-EXIT charts along each edge type by knowing the L-densities. Here, we introduce G-EXIT charts for the BIOSM channel. Assume that $\vec{\mathsf{a}}_v^l = (\mathsf{a}_{v,1}^l, \cdots, \mathsf{a}_{v,n_e}^l)$ and $\vec{\mathsf{a}}_c^l = (\mathsf{a}_{c,1}^l, \cdots, \mathsf{a}_{c,n_e}^l)$ denote the vector of L-densities at the output of VNs and CNs after $l \geq 1$ iterations. Then the density evolution for the L-densities as presented in (4.6) and (4.7) can be written as:

$$\begin{aligned}
\vec{\mathsf{a}}_v^{l+1} &= \lambda(\vec{\mathsf{a}}_{\mathrm{BIOSMC}}, \vec{\mathsf{a}}_c^{l+1}) \ , \\
\vec{\mathsf{a}}_c^{l+1} &= \rho(\vec{\mathsf{a}}_v^{l}) \ ,
\end{aligned}$$

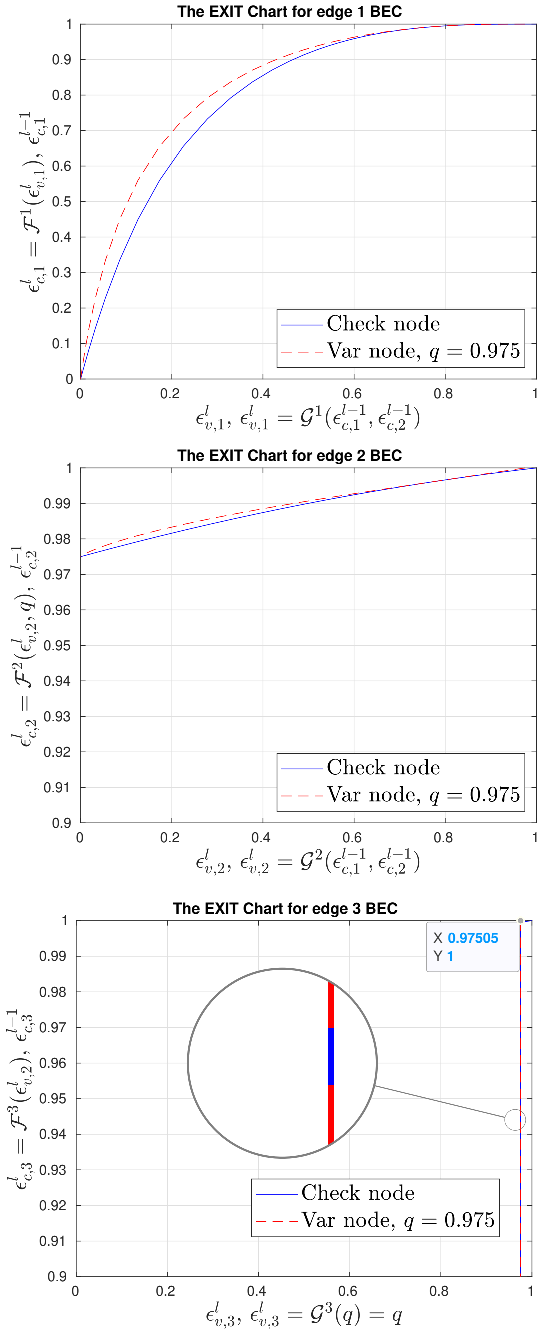


Figure 4.6: The G-EXIT charts for separate edges for the rate 0.02 MET-LDPC code detailed in Table 4.2. The threshold of this code on a BEC is 0.97505, the Shannon limit on binary erasure channel for rate 0.02 is 0.98.

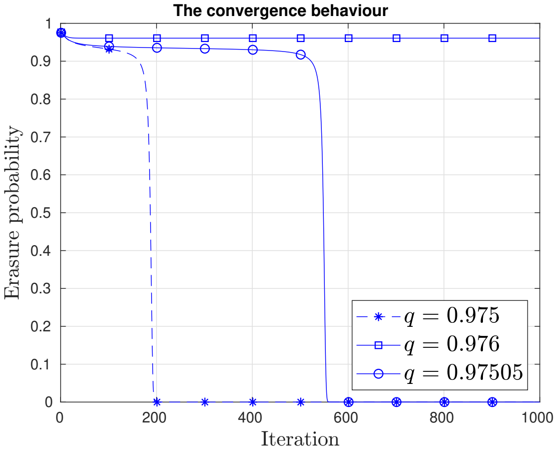


Figure 4.7: The convergence behavior of a MET-LDPC code on a BEC channel.

where $\rho(\mathbf{x})$ and $\lambda(\mathbf{r}, \mathbf{x})$ are presented in (4.2) and $\vec{\mathsf{a}}_{\text{BIOSMC}}$ is the L-density of the BIOSM channel. The initial L-density vector is $\vec{\mathsf{a}}_v^0 = \vec{\mathbf{\Delta}}_0$. By $\vec{\mathbf{\Delta}}_0$ we mean a vector of densities, where each density is δ_0, which is equivalent to erasure with probability 1. Similar to the case of BEC we consider the MET-LDPC codes with cascade structure and three edge types. Then the density evolution can be written as:

$$\begin{aligned}
\mathsf{a}_{v,1}^l &= \lambda^1(\mathsf{a}_{\text{BIOSMC}}, \mathsf{a}_{c,1}^{l-1}, \mathsf{a}_{c,2}^{l-1}) \ , \\
\mathsf{a}_{v,2}^l &= \lambda^2(\mathsf{a}_{\text{BIOSMC}}, \mathsf{a}_{c,1}^{l-1}, \mathsf{a}_{c,2}^{l-1}) \ , \\
\mathsf{a}_{v,3}^l &= \mathsf{a}_{\text{BIOSMC}} \ .
\end{aligned}$$

For the CN the update rules are:

$$\begin{aligned}
\mathsf{a}_{c,1}^l &= \rho^1(\mathsf{a}_{v,1}^l) \ , \\
\mathsf{a}_{c,2}^l &= \rho^2(\mathsf{a}_{v,2}^l, \mathsf{a}_{v,3}^l) \ , \\
\mathsf{a}_{c,3}^l &= \rho^3(\mathsf{a}_{v,2}^l) \ .
\end{aligned}$$

In contrast to the case of a BEC the *intermediate* L-densities $\vec{\mathsf{a}}^l$ for general BIOSM channel do no have a simple description. However we can estimate them with some equivalent density families and then apply the EXIT functional (G-EXIT functional) to obtain the EXIT (G-EXIT) charts. As presented in [44] the most *faithful* equivalence rule is to choose the element of the channel family which has *equal entropy*. In Appendix A the entropy functional is defined for the L-densities. In addition, it is shown how to apply the EXIT and G-EXIT functionals to the L-densities.

Let us assume that for a pair of $(\lambda^i,\ \rho^i)$ we are able to guess the true intermediate L-densities. We denote by $\mathsf{a}_{v,i}^l$ $(\mathsf{a}_{c,i}^l)$ the densities emitted at the VNs (CNs) at iteration l along edge type i. The parametric forms for the EXIT curves can then be displayed in Table 4.6.

Table 4.6: The parametric representation of the EXIT chart for MET-LDPC codes on a BIOSM channel.

	EXIT curve of CN	Inverse of EXIT curve of VN
Edge i	$\{\mathtt{h}^l_{v,i},\ \mathtt{h}^l_{c,i}\}$	$\{\mathtt{h}^l_{v,i},\ \mathtt{h}^{l-1}_{c,i}\}$

In Table 4.6 the entropy functional[2] for the L-densities are defined as

$$\mathtt{h}^l_{c,i} = H(\rho(\mathtt{a}^{l-1}_{v,i}))\ , \qquad \mathtt{h}^l_{v,i} = H(\mathtt{a}_{\text{BIOSMC}}\ \circledast\ \lambda(\mathtt{a}^l_{c,i}))\ .$$

and for the G-EXIT curves the parametric forms are displayed in Table 4.7.

Table 4.7: The parametric representation of G-EXIT chart for MET-LDPC codes on BIOSM channel.

		Inverse of
	G-EXIT curve of CN	Dual G-EXIT curve of VN
Edge i	$\{\mathtt{h}^l_{v,i},\ G(\mathtt{a}^l_{v,i}, \mathtt{a}^l_{c,i})\}$	$\{\mathtt{h}^l_{v,i},\ G(\mathtt{a}^l_{v,i}, \mathtt{a}^{l-1}_{c,i})\}$

In the following example we show how to plot the G-EXIT curve for a MET-LDPC code on a BI-AWGN channel.

Example 4.3.2. *G-EXIT chart for MET-LDPC codes on a BI-AWGN*

In this example we show how the G-EXIT charts can be used for MET-LDPC codes on a BI-AWGN channel. Consider the MET-LDPC code with rate 0.02 in Table 4.2. The Shannon limit for rate 0.02 is equal to $E_b/N_0 = -1.53$ dB ($\sigma^*_{\text{Sh}} = 5.96$) and our proposed code has a threshold equal to -1.5 dB ($\sigma^*_{\text{DE}} = 5.94$) which is just 0.03 dB away from capacity. With E_b being the energy per bit and N_0 being the energy of the noise, the relation between E_b/N_0, the SNR and σ for an AWGN channel with binary transmission is given by (linear scale)

$$\begin{aligned} \text{SNR} &= 2R^{\text{ch}}\frac{E_b}{N_0}\ , \\ \sigma &= \frac{1}{\sqrt{\text{SNR}}}\ . \end{aligned}$$

Using (4.2) the edge perspective degree distribution of this code can be written as

$$\begin{aligned} \lambda(\mathbf{r},\mathbf{x}) &= \left[0.6\, r_1 x_1^2 x_2^{60} + 0.4\, r_1 x_1 x_2^{51},\ 0.54505\, r_1 x_1^3 x_2^{59} + 0.4595\, r_1 x_1^2 x_2^{50},\ r_1\right]\ , \\ \rho(\mathbf{x}) &= \left[0.64\, x_1^3 + 0.36\, x_1^8,\ 0.4054\, x_2^2 x_3 + 0.5946\, x_2 x_3,\ 0.3125\, x_2^3 + 0.6875\, x_2^2\right]\ , \end{aligned}$$

[2] The definition of the entropy functional, EXIT and G-EXIT functionals are presented in Appendix A.

where for the VNs

$$\begin{aligned}\lambda^1(r_1,x_1,x_2) &= 0.6\,r_1x_1^2x_2^{60} + 0.4\,r_1x_1x_2^{51}\ ,\\ \lambda^2(r_1,x_1,x_2) &= 0.54505\,r_1x_1^3x_2^{59} + 0.4595\,r_1x_1^2x_2^{50}\ ,\\ \lambda^3(r_1) &= r_1\ ,\end{aligned}$$

and for the CNs

$$\begin{aligned}\rho^1(x_1) &= 0.64\,x_1^3 + 0.36\,x_1^8\ ,\\ \rho^2(x_2,x_3) &= 0.4054\,x_2^2x_3 + 0.5946\,x_2x_3,\\ \rho^3(x_2) &= 0.3125\,x_2^3 + 0.6875\,x_2^2\ .\end{aligned}$$

Then, for the density evolution the intermediate densities on the VN side are

$$\begin{aligned}\mathsf{a}_{v,1}^l &= \mathsf{a}_{\text{BIOSMC}} \circledast \left[0.6\,\left(\mathsf{a}_{c,1}^{l-1}\right)^{\circledast 2} \circledast \left(\mathsf{a}_{c,2}^{l-1}\right)^{\circledast 60} + 0.4\,\mathsf{a}_{c,1}^{l-1} \circledast \left(\mathsf{a}_{c,2}^{l-1}\right)^{\circledast 51}\ \right],\\ \mathsf{a}_{v,2}^l &= \mathsf{a}_{\text{BIOSMC}} \circledast \left[0.54505\,\left(\mathsf{a}_{c,1}^{l-1}\right)^{\circledast 3} \circledast \left(\mathsf{a}_{c,2}^{l-1}\right)^{\circledast 59} + 0.4595\,\left(\mathsf{a}_{c,1}^{l-1}\right)^{\circledast 2} \circledast \left(\mathsf{a}_{c,2}^{l-1}\right)^{\circledast 50}\right]\ ,\\ \mathsf{a}_{v,3}^l &= \mathsf{a}_{\text{BIOSMC}}\ ,\end{aligned}$$

where $\circledast$ denotes the convolution of VNs, and for the CNs

$$\begin{aligned}\mathsf{a}_{c,1}^l &= 0.64\,\left(\mathsf{a}_{v,1}^l\right)^{\boxtimes 3} + 0.36\,\left(\mathsf{a}_{v,1}^l\right)^{\boxtimes 8}\ ,\\ \mathsf{a}_{c,2}^l &= \mathsf{a}_{\text{BIOSMC}} \boxtimes \left[0.4054\,\left(\mathsf{a}_{v,2}^l\right)^{\boxtimes 2} + 0.5946\,\mathsf{a}_{v,2}^l\right],\\ \mathsf{a}_{c,3}^l &= 0.3125\,\left(\mathsf{a}_{v,2}^l\right)^{\boxtimes 3} + 0.6875\,\left(\mathsf{a}_{v,2}^l\right)^{\boxtimes 2}\ ,\end{aligned}$$

where $\boxtimes$ denotes the convolution of CNs. Figure 4.8 shows the convergence behavior of each edge for the above mentioned code.

4.4 Code design optimization problem

Let us denote the decoding threshold of a MET-LDPC code by θ. The parameter θ describes the maximum noise level in which the decoder can provide a reliable transmission. For a BI-AWGN channel the parameter $\theta = \sigma$, which is standard deviation of the noise. For the BE channel $\theta = q$, which is the erasure probability. The parameter θ is related to the code structure and can be written as $\theta\left(\nu(\mathbf{r},\mathbf{x}),\mu(\mathbf{x})\right)$. Thus, for a given rate, the code design problem can be written as an optimization problem with cost function θ as follows:

$$\theta^* = \underset{\nu(\mathbf{r},\mathbf{x}),\ \mu(\mathbf{x})}{\text{argmax.}}\ \theta \qquad (4.12)$$

$$\text{s.t.}\ \begin{cases}\nu_{x_i}(\mathbb{1},\mathbb{1}) = \mu_{x_i}(\mathbb{1}), & i = 1,\cdots,n_e\ .\\ R = \nu(\mathbb{1},\mathbb{1}) - \mu(\mathbb{1})\ .\end{cases}$$

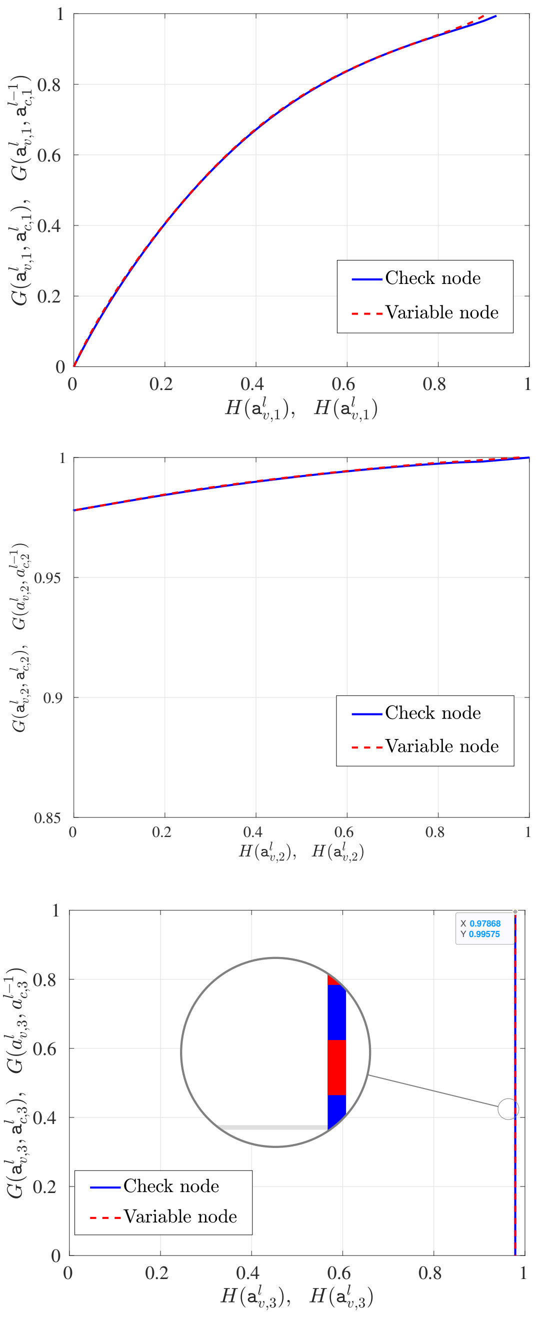


Figure 4.8: The G-EXIT charts for the three edge types for the rate 0.02 MET-LDPC code for a BI-AWGN channel. The detailed code structure can be found in Table 4.2.

This means that the optimization problem looks among all valid degree distributions and finds the code with maximum threshold. The precise calculation of the threshold can be done using density evolution. Using the density evolution this problem is an optimization problem with non-linear cost function. Though the constraints of this optimization problem can be simplified to a convex set, solving (4.12) is complicated due to the non-linearity of the objective function even in the case of irregular LDPC codes.

4.4.1 Constraint sets

Let us consider the constraint sets of the original optimization problem in (4.12) for a MET-LDPC code with degree distribution as presented in Table 4.1. The n_e linear constraints for the socket count equalities (SCEs) imply that the weighted sum of the VNs and CNs for each edge type should be equal. For the j^{th} constraint it can be written as

$$\sum_{i=1}^{Q} \nu_i d^v_{j,i} = \sum_{i=1}^{P} \mu_i d^c_{j,i} \ ,$$

where $1 \leq j \leq n_e$. Its matrix representation is

$$\begin{pmatrix} d^v_{1,1} & d^v_{1,2} & \cdots & d^v_{1,Q} \\ \vdots & \vdots & \ddots & \vdots \\ d^v_{j,1} & d^v_{j,2} & \cdots & d^v_{j,Q} \\ \vdots & \vdots & \ddots & \vdots \\ d^v_{n_e,1} & d^v_{n_e,2} & \cdots & d^v_{n_e,Q} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_Q \end{pmatrix} = \begin{pmatrix} d^c_{1,1} & d^c_{1,2} & \cdots & d^c_{1,P} \\ \vdots & \vdots & \ddots & \vdots \\ d^c_{j,1} & d^c_{j,2} & \cdots & d^c_{j,P} \\ \vdots & \vdots & \ddots & \vdots \\ d^c_{n_e,1} & d^c_{n_e,2} & \cdots & d^c_{n_e,P} \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_P \end{pmatrix} . \tag{4.13}$$

By calling the above matrices $\mathbf{D^v} = \mathbf{d_v^T}$, $\vec{\nu}$, $\mathbf{D^c} = \mathbf{d_c^T}$ and $\vec{\mu}$ respectively, we have

$$\mathbf{D^v}\vec{\nu} = \mathbf{D^c}\vec{\mu} \ . \tag{4.14}$$

If a nonsingular matrix $\mathbf{D^v}$ exists the vector $\vec{\nu}$ can be calculated by solving

$$\vec{\nu} = (\mathbf{D^v})^{-1}\mathbf{D^c}\vec{\mu} \ . \tag{4.15}$$

Thus, by considering $\mathbf{D^v}$, $\mathbf{D^c}$ and $\vec{\mu}$ as independent variables the vector $\vec{\nu}$ is a the dependent variable and (4.15) satisfies the validity. This means that to find the best degree distribution it is possible to reduce the dimension of the search space. Furthermore, there is no need for joint optimization of CN and VN coefficients because of their linear dependency. Hence (4.15) introduce a linear dependency for node distribution of the VNs.

In addition, the second constraint is related to the code rate R. Since we assume that there are no puncture nodes, $\sum_1^Q \nu_j = 1$ and $\sum_1^P \mu_i = 1 - R$, where R is the rate of the code. In total the optimization algorithm therefore needs to find $P - 1$ fractional parameters and $n_e(P + Q)$ integer parameters.

Example 4.4.1. *Linear dependency*

As an example, Table 4.8 describes a rate 0.02 MET-LDPC ensemble presented in [10]. Using (4.14) the matrix form representation for the socket count equality is given by

Table 4.8: Table presentation of Rate 0.02 degree structure for a MET-LDPC code with 3 edge-types.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$	$\mathbf{d^v}$	$\mu_{\mathbf{d}}$	$\mathbf{d^c}$
0.0225	[0 1]	2 57 0	0.010625	3 0 0
0.0175		3 57 0	0.009375	7 0 0
0.96		0 0 1	0.6	0 2 1
			0.36	0 3 1

BIAWGN: $\sigma^*_{\mathrm{DE}} = 5.93$

$$\begin{pmatrix} 2 & 3 & 0 \\ 57 & 57 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = \begin{pmatrix} 3 & 7 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0.0106 \\ 0.0094 \\ 0.6 \\ 0.36 \end{pmatrix}.$$

Solving this equation for ν by using (4.15) we obtain

$$\begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = \begin{pmatrix} 0.0225 \\ 0.0175 \\ 0.9600 \end{pmatrix},$$

which are exactly the same result as obtained by an optimization method in [10] (c.f. Table 4.8).

4.4.2 Cost function for irregular LDPC codes

On a binary erasure channel and for irregular-LDPC codes, i.e. $n_e = 1$, the density evolution is equivalent to monitoring scalar value erasure probability. Thus, the EXIT chart can be used to convert this optimization problem to a linear curve fitting optimization problem on a convex set [65, 67, 69, 70]. Remember that the EXIT chart provides two curves related to the CNs and the inverse of VNs. For any converged code these two curves do not cross each other and the VN curve is always on top of the CN curve. Then, using the Area theorem [68, 66], the best code can be obtained by matching these two curves. If $\mathcal{A}$ denotes the area between these two curves the

(4.12) can be simplified to

$$\underset{\nu(q,x),\ \mu(x)}{\text{argmin.}}\ \mathcal{A} \tag{4.16}$$
$$\text{s.t.}\ \begin{cases}\nu_x(1,1) = \mu_x(1), \\ R = \nu(1,1) - \mu(1)\ .\end{cases}$$

This simplified optimization problem presented in (4.16) looks for an ensemble of irregular-LDPC codes where the area between the two curves related to VNs and CNs is minimal. This can be done by starting with a simple CN polynomial and then use curve fitting using Taylor series [65].

4.4.3 Cost function for MET-LDPC codes

In the case of a MET-LDPC code the optimization problem is not always straight forward. Here we propose a design approach by using the concept of EXIT charts along different edge types. Let us denote by $\mathcal{A}^i$ the area between the curves of EXIT charts along i^{th} edge. Then, for the MET-LDPC codes the optimization problem (4.12) can be simplified to

$$\underset{\nu(q,\mathbf{x}),\ \mu(\mathbf{x})}{\text{argmin.}} \sum_{i}^{n_e} \mathcal{A}^i \tag{4.17}$$
$$\text{s.t.}\ \begin{cases}\nu_{x_i}(\mathbb{1},\mathbb{1}) = \mu_{x_i}(\mathbb{1}), & i = 1, \cdots, n_e\ . \\ R = \nu(\mathbb{1},\mathbb{1}) - \mu(\mathbb{1})\ .\end{cases}$$

This problem is a generalization of the curve fitting problem for irregular-LDPC codes. Here we jointly optimize the EXIT curves for all edges. Luckily, in the cascade structure the EXIT curves for the thirs edge are vertical lines related to the degree one nodes. Tthe area between the two curves is allways zero (See Figure 4.6 and Figure 4.8). Thus the optimization problem simplified to jointly optimize the EXIT curves for edges 1 and 2. Since the EXIT curves for CN sides along edge 1 and edge 2 are independent, we can start by designing two EXIT curves for CNs and then fit two VN curves *jointly* to these CN curves.

4.5 Optimization for cascade structure

After having defined the optimization problem in the las section, we now focus on solving it. The design of MET-LDPC codes is still a challenging problem. Just very recently some optimization algorithms were developed for designing MET-LDPC codes [14, 13, 12]. In general all the design approaches solve the non-linear optimization problem as presented in (4.12). Usually a brute-force search is running on a set of valid degree distributions to find the best ensemble. The density evolution (DE) or one of its approximations [13, 14] are used to check the performance of the codes. For example, the author in [13] used a hybrid-Gaussian approximation to reduce the complexity of DE for MET-LDPC codes. Recently a joint optimization was developed by [12], where two complicated inner and outer optimizations were conducted to find the node and edge distributions. The major disadvantage of [13, 12] is that they are limited to code rates higher than 0.1. Our proposed algorithm can however design highly efficient codes at low rate. In Section 4.2, we proposed a new semi-Gaussian approximation as to reduce the complexity of the DE for MET-LDPC codes. In addition, we denoted in Section 4.4 that a linear dependency exists between the CNs and VNs in terms of code rate, which greatly reduces the search space for valid structures. In this section, for the case of MET-LDPC codes with cascade structure, we propose a novel optimization algorithm to design highly efficient codes. We use the concepts of EXIT chart and semi-Gaussian approximation to design the codes.

Remember from Section 4.1.1 that the MET-LDPC codes with cascade structure can be described by three sub-graphs as presented in Figure 4.1. In cascade structure the red part of the graph can be considered as an irregular LDPC code with rate r_b which we call the base code (See Figure 4.1, and delete all the other blue and black edges and all other connected nodes). Later in this section we will see how to design an appropriate base code using the EXIT charts for the cascade structures. It is important to mention that designing good base codes is not equivalent to designing good irregular LDPC codes for BIOSM channels. For the moment, assume that a base code with rate r_b exists. In the following theorem we will prove that the portion of the degree one nodes in the overall cascade structure is determined b knowing the degree distribution of the base code (red part).

Theorem 4.5.1. *Consider a MET-LDPC code with rate R in cascade structure. The portion of degree one VNs is equal to $1 - \frac{R}{r_b}$, where r_b is the rate of the base code.*

Proof. Let us assume that the base code has the following degree distribution :

$$\nu(x) = \sum_{i=1}^{Q-1} \nu'_i \; x^{d^v_{1,i}} \;, \tag{4.18}$$

$$\mu(x) = \sum_{i=1}^{P-2} \mu'_i \; x^{d^c_{1,i}} \;. \tag{4.19}$$

where $\sum\limits_{i=1}^{Q-1} \nu'_i = 1$ and $\sum\limits_{i=1}^{P-2} \mu'_i = 1 - r_b$. The socket count equality for the base code is

$$\sum_{i=1}^{Q-1} \nu'_i \; d^v_{1,i} = \sum_{i=1}^{P-2} \mu'_i \; x^{d^c_{1,i}} \;. \tag{4.20}$$

In addition, assume that the degree distribution of the overall cascade structure is according to Table 4.3 and the socket count equality for edge-type 1 of the overall cascade structure is given by

$$\sum_{i=1}^{Q-1} \nu_i \; d^v_{1,i} = \sum_{i=1}^{P-2} \mu_i \; x^{d^c_{1,i}} \; . \tag{4.21}$$

It is clear that (4.20) and (4.21) are scaled versions of each other. Let us denote the scale factor by s. This yields

$$\sum_{i=1}^{Q-1} \nu_i = \frac{\sum\limits_{i=1}^{Q-1} \nu'_i}{s} = \frac{1}{s} \; , \tag{4.22}$$

$$\sum_{i=1}^{P-2} \mu_i = \frac{\sum\limits_{i=1}^{P-2} \mu'_i}{s} = \frac{1-r_b}{s} \; . \tag{4.23}$$

$$\tag{4.24}$$

In addition, for the overall cascade structure

$$\nu(\mathbf{1},\mathbf{1}) = \sum_{i=1}^{Q-1} \nu_i + \nu_Q = 1 \; , \tag{4.25}$$

$$\mu(\mathbf{1}) = \sum_{i=1}^{P-2} \mu_i + \mu_{P-1} + \mu_P = 1 - R \; , \tag{4.26}$$

$$\nu_Q = \mu_{P-1} + \mu_P \; . \tag{4.27}$$

Thus, (4.25) and (4.26) can be simplified to

$$\nu(\mathbf{1},\mathbf{1}) = \frac{1}{s} + \nu_Q = 1 \; ,$$

$$\mu(\mathbf{1}) = \frac{1-r_b}{s} + \nu_Q = 1 - R \; .$$

Finally, solving this equation for s yields

$$s = \frac{r_b}{R} \; , \tag{4.28}$$

which implies that $\nu_Q = 1 - \frac{1}{s} = 1 - \frac{R}{r_b}$.

□

The main practical result of Theorem 4.5.1 is that if we use an irregular base code with rate r_b to design a cascade structure of rate R, the scale factor $s = \dfrac{r_b}{R}$ and the scaled-down version of the designed base code with rate r_b can be considered as red part in overall cascade structure.

4.5.1 Design of good base codes using EXIT charts

To design highly ??? MET-LDPC codes in cascade structure a good irregular LDPC base code is required. However the procedure of designing an optimal base code differs from the procedure of designing conventional irregular-LDPC codes (See Section 4.4.2) because the noise model is different. As discussed in Section 4.4.2, designing an irregular LDPC code for a given channel parameter is a curve fitting problem and we have to minimize the area between the EXIT curves as the gap between the two EXIT curves determines the distance to capacity. The design procedure for the base code, however, has to take into account the additional noise comming from edge type 2. For example, consider the Irregular code presented in Figure 4.5. The modified version of this code showing the effect of edge type 2 has the following degree distribution

$$\nu(q, x_1, x_2) = q\ (0.5\ x_1^2 x_2^{d^v_{2,1}} +\ 0.5\ x_1^3 x_2^{d^v_{2,2}})\ , \tag{4.29}$$

$$\mu(x) = 0.4\ x_1^4 +\ 0.1\ x_1^9\ . \tag{4.30}$$

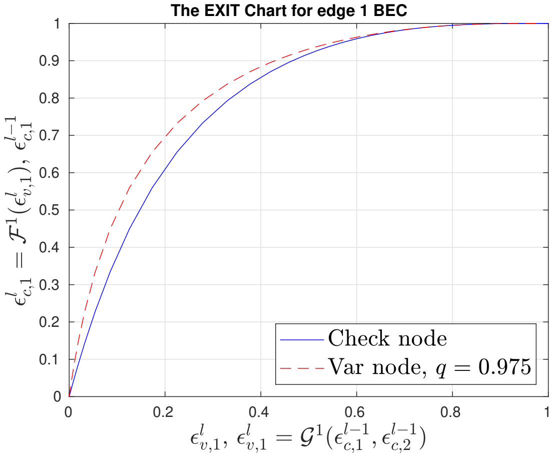


Figure 4.9: The EXIT chart for base code by considering the effect of additional noise coming from edge type 2. The channel parameter is $q = 0.975$. The code rate is 0.5 and the maximum degree of VN along edge type one is 3 and the maximum VN degree along edge type two is 60.

The corresponding EXIT chart which considers the effect of the additional noise coming from the output of CNs along edge type 2 is presented in Figure 4.9. The corresponding curves are plotted for $d^v_{2,1} = 50$ and $d^v_{2,2} = 60$. Comparing Figure 4.5 and Figure 4.9 demonstrates that the base code designed for MET-LDPC codess is not always the optimal irregular LDPC code for the BE channel. More precisely, if the additional noise along edge type 2 is just considered as pure noise (equivalently,

$\epsilon^l_{c,2} = 1$) the EXIT chart in Figure 4.9 would be the same as the EXIT chart in Figure 4.5.

4.5.2 Optimization for the connector part

After designing the base code with rate r_b and determining the portion of the degree one nodes according to Theorem 4.5.1 the final step is to find the best connector part to build an overall MET-LDPC code with rate R. Thus, as presented in Table 4.3, just the parameters related to edge type 2 remain unknown (blue part). The SCEs for edge type 2 and edge type 3 read

$$\begin{cases} d^c_{2,P-1}\ \mu_{P-1} & + \ d^c_{2,P}\ \mu_P = \sum_{i=1}^{Q-1} d^v_{2,i}\ \nu_i \ , \\ \qquad \mu_{P-1} & + \qquad \mu_P = \nu_Q = 1 - \dfrac{1}{s} \ . \end{cases} \tag{4.31}$$

In (4.31) the parameters ν_i for $i \in \{1, \cdots, Q\}$ and s are known. The unknown parameters are μ_{P-1}, μ_P, $d^c_{2,P-1}$, $d^c_{2,P}$ and $d^v_{2,i}$ for $i \in \{1, \cdots, Q-1\}$. We note that $d^v_{2,i}$ are integers. Solving this set of equations for μ_{P-1} and μ_P gives us the parametric solution for connector parts satisfying the SCE. The parametric forms for the CN coefficients μ_{P-1} and μ_P are

$$\mu_{P-1} = \frac{d^c_{2,P}\ \nu_Q - \sum_{i=1}^{Q-1} d^v_{2,i}\ \nu_i}{d^c_{2,P} - d^c_{2,P-1}} \ , \tag{4.32}$$

$$\mu_P = \nu_Q - \mu_{P-1} \ , \tag{4.33}$$

where μ_{P-1} and μ_P should be between 0 and ν_Q.

In addition, the schematic representation for the solutions of this set of equations for μ_{P-1} and μ_P are presented in Figure 4.10 which is the cross point of the following lines

$$\begin{cases} y = & -m\ x + b \ , \\ y = & -x + \nu_Q \ , \end{cases}$$

where

$$m = \frac{d^c_{2,P-1}}{d^c_{2,P}} \ , \qquad b = \frac{\sum_{i=1}^{Q-1} d^v_{2,i}\ \nu_i}{d^c_{2,P}} \ .$$

It is clear from Figure 4.10 that by changing the unknown integer parameters ($d^v_{2,i}$), we can change the slope (m) and y-intercept (b) to find different pairs of (μ_{P-1}, μ_P). It it important to mention that each point determines a new connector part for the overall MET-LDPC code. Finally to find the best connector part for the overall code we can use the semi-Gaussian approximation (or any other approximation of DE) to find the best threshold for the overall code. This can be done by testing different pairs of (μ_{P-1}, μ_P) according to (4.32) and (4.33).

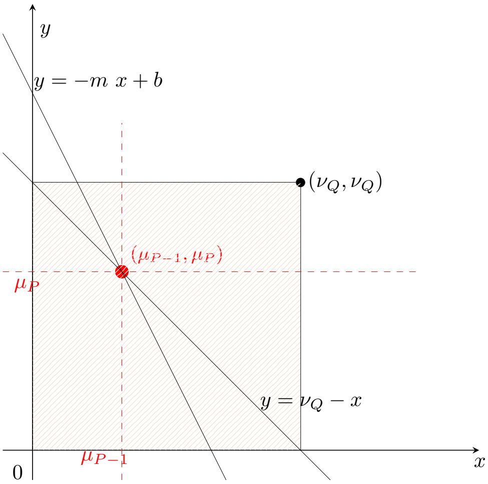


Figure 4.10: The inner optimization problem finds sets of points (μ_{P-1}, μ_P) satisfying $\mu_P + \mu_{P-1} = \nu_Q$.

The flowchart for the code design procedure is presented in Figure 4.11. In the first step a base code with rate r_b is designed according to Section 4.5.1. Then, in the second step the scaling parameter s is found from Theorem 4.5.1. Putting the scaled-down version of the base code in the cascade structure determines the fraction of degree one VNs and CNs. In the next step a new connector part is tested to build the overall MET-LDPC code with rate R. This procedure continues until we find a set of highly efficient codes or until all the unique connector parts have been tested. If for all the created connector parts the efficiency of the overall code was not good enough a new base code has to be considered with different rate r_b.

In Section 4.6 some examples of highly efficient MET-LDPC codes with cascade structure are represented, where we used our new optimization algorithm for the design.

4.6 Highly efficient codes

In this section we propose some of our highly efficient MET-LDPC codes. To compare the performance of these codes we define the asymptotic efficiency of each code as $\beta_{\text{DE}} = \frac{R}{C(\mathtt{p}_{DE})}$, where R is the code rate and $C(\mathtt{p}_{DE})$ is the capacity of the binary-input output symmetric memory-less (BIOSM) channel with parameter $\mathtt{p} = \mathtt{p}_{DE}$. $\mathtt{p}_{DE}$ is the threshold of the code obtained by running full density evolution and $\mathtt{p}_{Sh}$ is the Shannon threshold. To see how we calculate the capacity of BIOSM channels see Appendix A.

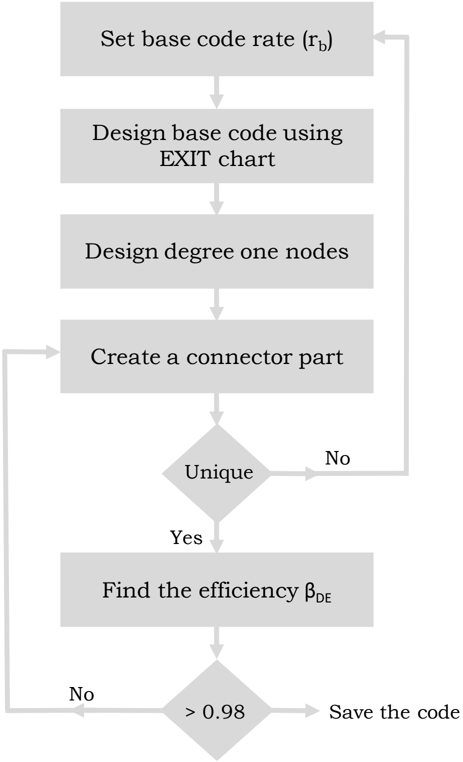

Set base code rate (r_b)
Design base code using EXIT chart
Design degree one nodes
Create a connector part
Unique
No
Yes
Find the efficiency β_{DE}
No
> 0.98
Save the code

Figure 4.11: The block diagram for the optimization algorithm of a rate R MET-LDPC code with cascade structure using a base code of rate r_b.

Example 4.6.1. *MET-LDPC code with rate* 0.01.

The degree distribution of the code is

Table 4.9: Rate 0.01 MET-LDPC code.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$	$\mathbf{d^v}$			$\mu_{\mathbf{d}}$	$\mathbf{d^c}$		
0.01	[0 1]	2	103	0	0.008	4	0	0
0.01		3	125	0	0.002	9	0	0
0.98		0	0	1	0.32	0	3	1
					0.66	0	2	1
$\sigma^*_{\text{Sh}} = 8.46$			$\sigma^*_{\text{DE}} = 8.41$			$\beta_{\text{DE}} = 98.7\%$		

Based on our knowledge, there is no MET-LDPC code in other literature specifically designed for rate 0.01. A rate 0.01 code can be obtained by augmenting a rate

0.02 MET-LDPC code with a length 2 repetition code.

Example 4.6.2. *MET-LDPC code with rate* 0.02.

We propose two MET-LDPC codes with rate 0.02. The first code was represented before in Table 4.2. Here we present the second code with rate 0.02 with the following degree distribution in table format

Table 4.10: Rate 0.02 MET-LDPC code.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$	$\mathbf{d^v}$			$\mu_{\mathbf{d}}$	$\mathbf{d^c}$		
0.024	[0 1]	2	43	0	0.002	3	0	0
0.016		3	61	0	0.018	5	0	0
0.96		0	0	1	0.088	0	3	1
					0.872	0	2	1

$\sigma^*_{\text{Sh}} = 5.96$	$\sigma^*_{\text{DE}} = 5.94$	$\beta_{\text{DE}} = 99.2\%$

The only existing MET-LDPC code with rate 0.02 was designed by [10] with the following degree distribution

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.0225\ r_1 x_1^2 x_2^{57} + 0.0175\ r_1 x_1^3 x_2^{57} + 0.96\ r_1 x_3^1\ ,\\ \mu(\mathbf{x}) &= 0.010625\ x_1^3 + 0.009375\ x_1^7 + 0.6\ x_2^2 x_3^1 + 0.36\ x_2^3 x_3^1\ .\end{aligned}$$

The threshold of this code in BI-AWGN channel is $\sigma^*_{\text{DE}} = 5.91$ with asymptotic efficiency of $\beta_{\text{DE}} = 98.22\%$.

Example 4.6.3. *MET-LDPC code with rate* 0.05.

Table 4.11 shows the degree structure of a rate 0.05 MET-LDPC code. The threshold of this code using DE on BI-AWGN channel is equal to $\sigma^*_{\text{DE}} = 3.68$ ($\frac{E_b}{N_0} = -1.32$ dB). The theoretical Shannon limit for a code of rate 0.05 is equal to $\sigma^*_{\text{Sh}} = 3.73$ (-1.44 dB) and this code is just 0.12 dB away from capacity. Another code designed for rate 0.05 was presented in [45] and has the following degree distribution

Table 4.11: Rate 0.05 MET-LDPC code.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$	$\mathbf{d^v}$			$\mu_{\mathbf{d}}$	$\mathbf{d^c}$		
0.054	[0 1]	2	22	0	0.026	3	0	0
0.046		3	22	0	0.024	7	0	0
0.90		0	0	1	0.40	0	3	1
					0.50	0	2	1

$\sigma^*_{\text{Sh}} = 3.73$	$\sigma^*_{\text{DE}} = 3.68$	$\beta_{\text{DE}} = 97.36\%$

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.04\ r_1 x_1^2 x_2^{34} + 0.03\ r_1 x_1^3 x_2^{34} + 0.93\ r_1 x_3^1\ ,\\ \mu(\mathbf{x}) &= 0.01\ x_1^8 + 0.01\ x_1^9 + 0.41\ x_2^2 x_3^1 + 0.52\ x_2^3 x_3^1\ .\end{aligned}$$

The threshold of this code in BI-AWGN channel is $\sigma^*_{\text{DE}} = 3.67$ with asymptotic efficiency of $\beta_{\text{DE}} = 96.78\%$.

Example 4.6.4. *MET-LDPC code with rate* 0.10.

Table 4.12 shows a code with $n_e = 3$ edge type and the threshold of this code in a BI-AWGN channel using DE is equal to $\sigma^*_{\text{DE}} = 2.57$ ($\frac{E_b}{N_0} = -1.22$ dB). The theoretical Shannon limit is equal to $\sigma^*_{\text{Sh}} = 2.6$ (-1.28 dB) and this code is just 0.06 dB away from capacity.

Table 4.12: Rate 0.10 MET-LDPC code.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$	$\mathbf{d^v}$	$\mu_{\mathbf{d}}$	$\mathbf{d^c}$
0.075	[0 1]	2 23 0	0.025	12 0 0
0.050		3 18 0	0.875	0 3 1
0.875		0 0 1		

$\sigma^*_{\text{Sh}} = 2.6$	$\sigma^*_{\text{DE}} = 2.57$	$\beta_{\text{DE}} = 98.33\%$

For comparison another code with rate 0.1 designed by [14] has the degree distribution

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.0775\ r_1x_1^1x_2^1x_3^{21} + 0.0477\ r_1x_1^2x_2^1x_3^{20} + 0.8747\ r_1x_4^1\ ,\\ \mu(\mathbf{x}) &= 0.0011\ x_1^6x_2^4 + 0.0028\ x_1^6x_2^5 + 0.0214\ x_1^7x_2^5 + \ 0.0412\ x_3^2x_4 + 0.8335\ x_3^3x_4\ .\end{aligned}$$

The threshold of this code in BI-AWGN channel is $\sigma^*_{\text{DE}} = 2.54$ with asymptotic efficiency of $\beta_{\text{DE}} = 96.37\%$.

Example 4.6.5. *MET-LDPC code with rate* 0.25.

The corresponding polynomial representation for this code is:

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.1875\ r_1x_1^2x_2^{12} + 0.125\ r_1x_1^3x_2^4 + 0.6875\ r_1x_3^1\ ,\\ \mu(\mathbf{x}) &= 0.0625\ x_1^{12} + 0.6875\ x_2^4x_3^1\ .\end{aligned}$$

The corresponding table representation is also presented in Table 4.13.

Table 4.13: Rate 0.25 MET-LDPC code.

$\nu_{\mathbf{bd}}$	$\mathbf{b}$	$\mathbf{d^v}$	$\mu_{\mathbf{d}}$	$\mathbf{d^c}$
0.1875	[0 1]	2 12 0	0.0625	12 0 0
0.125		3 4 0	0.6875	0 4 1
0.6875		0 0 1		

$\sigma^*_{\text{Sh}} = 1.55$	$\sigma^*_{\text{DE}} = 1.50$	$\beta_{\text{DE}} = 95.07\%$

Now let us compare the threshold of our designed code with some other codes with rate 0.25 designed in [14] and [12]. For example in [14] the authors designed another non-punctured code with rate 0.25 with the following degree distribution

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.1512\ r_1x_1^3 + 0.006\ r_1x_1^4 + 0.0928\ r_1x_1^{12} +\ 0.75\ r_1x_2^2\ ,\\ \mu(\mathbf{x}) &= 0.6588\ x_1^2x_2^2 + 0.0912\ x_1^3x_2^2\ .\end{aligned}$$

The threshold of this code in BI-AWGN channel is $\sigma^*_{\text{DE}} = 1.48$ with asymptotic efficiency of $\beta_{\text{DE}} = 93.08\%$.

The second code is a punctured code with rate 0.25 designed in [12] with the following degree distribution

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.1563\ r_0x_1^4x_3^4x_4^1 + 0.0938\ r_1x_3^2x_4^1 + 0.9062\ r_1x_2^1x_4^1\ ,\\ \mu(\mathbf{x}) &= 0.3125\ x_1^2x_2^1 + 0.1875\ x_2^1x_4^4 + 0.4063\ x_2^1x_3^2x_4^1\ .\end{aligned}$$

The threshold of this code in BI-AWGN channel is $\sigma^*_{\text{DE}} = 1.49$ with asymptotic efficiency of $\beta_{\text{DE}} = 93.65\%$.

4.7 Simulation Results

Finally, to evaluate the performance of our codes, Figure 4.12 shows the simulated frame error rate (FER) of our rate 0.02 MET-LDPC code, represented in Table 4.2, with finite block length. We compare our results with the simulated FER of a rate 0.02 code designed by [10] and used in Ref. [71] for multi-dimensional reconciliation for CV-QKD.

As can be seen in Figure 4.12 our proposed code outperforms the code published in [10] even with a shorter block length of $n = 1.5 \times 10^6$ bits. For both codes the progressive edge growth (PEG) algorithm [72] was used to construct quasi-cyclic MET-LDPC codes with $Z \times Z$ circulant permutation matrices. Based on our simulation results it is clear that by increasing the length of the code to $n = 1.5 \times 10^6$ bits, 0.1 dB additional gain can be obtained at FER = 0.1. In addition, in Figure 4.12 the vertical dashed lines respectively show the Shannon asymptotic threshold for rate 0.02 on a BIAWGN channel and the asymptotic threshold obtained by density evolution for our code and the code of [10].

The efficiencies of the two codes versus the FER are compared in Fig. 4.13. It can be observed that for all FERs the efficiency of our code is higher than the code of [10]. For instance, for an efficiency of 95% the code in [71, 10] has a FER of 0.6 but our code is able to provide the same efficiency with a FER of as low as 0.3.

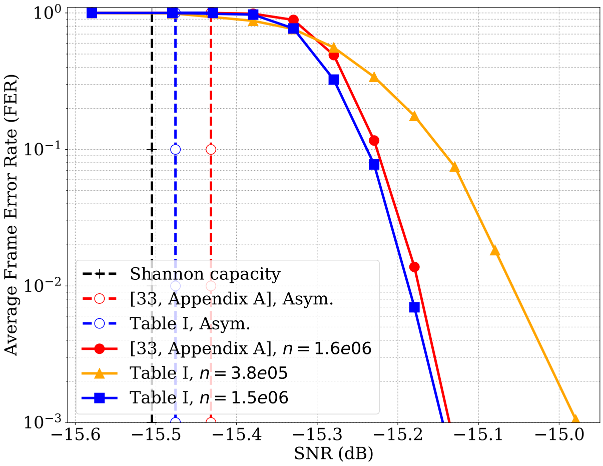


Figure 4.12: The frame error rate vs SNR for rate 0.02 MET-LDPC code. To plot the FER curves we set the maximum number of iterations to 500 and for each point 100 frames of errors were counted. We used $Z = 256$ for the solid orange curve and $Z = 1024$ for the solid red and blue curves.

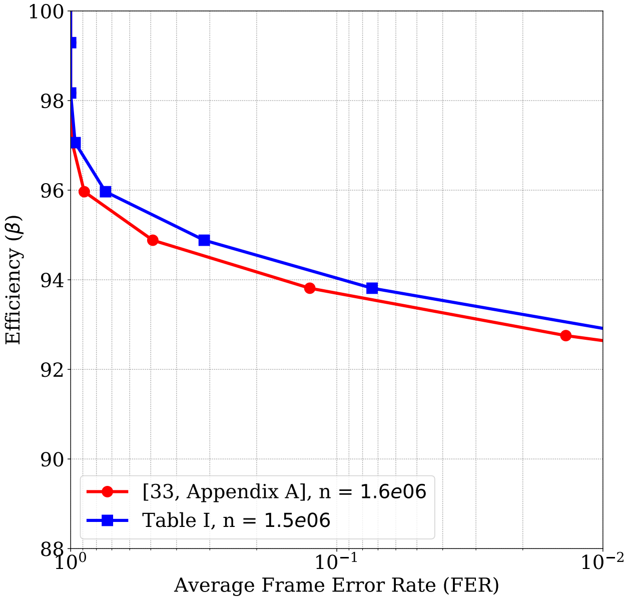


Figure 4.13: The efficiency vs frame error rate for rate 0.02 MET-LDPC code. To plot the FER curves we set the maximum number of iterations to 500 and for each point 100 frames of errors are counted..

CHAPTER 5

Conclusion and Future Directions

This thesis addressed some of the important topics in the post-processing of the CV-QKD to provide long distance secure key rate. More specifically, for the reconciliation schemes in CV-QKD some of the ubiquitous reconciliation schemes were discussed in details. Some of the major contributions in Chapter 3 are

- Numerical and analytical methods to precisely calculate the individual channel coding rates for MLC-MSD reconciliation.
- Calculation of soft information at the input of the soft decoder for one-level decoding and two-level decoding for MLC-MSD reconciliation.
- Detailed comparison between forward and reverse reconciliation.
- Randomized reconciliation scheme was introduced to increase the throughput of the system. In randomized reconciliation a high throughput hard decoder were used instead of complicated soft decoder.

Another important topic which was addressed in this thesis is the major need to design highly efficient codes for low rate regime in applications like QKD. In Chapter 4, we focused on MET-LDPC codes. A new precise approximation method for DE, called semi-Gaussian approximation was proposed for the MET-LDPC codes. In addition, we developed the concept of EXIT charts for the MET-LDPC codes. The EXIT chart is one of the powerful tools for designing the codes. A new algorithmic approach was proposed to design highly efficient MET-LDPC codes. Then some new highly efficient MET-LDPC codes were designed using the proposed algorithm. The application of these MET-LDPC codes are not limited to the CV-QKD and many of the published codes can be used also in other applications including satellite communication, wireless communication and optical communication. More precisely some of the major contributions of Chapter 4 can be summarized as follows:

- A new approximation for DE of the MET-LDPC codes called semi-Gaussian approximation. In this method the Gaussian approximation is applied just to VN operations and the CN operations are calculated precisely. This method can help us to plot the EXIT charts without running the full DE. More details can be found in Section 4.2.3.

- Developed the concept of EXIT and generalized EXIT (G-EXIT) charts for MET-LDPC codes. EXIT and G-EXIT charts are very useful tools to analysis and design of irregular LDPC codes. These concepts where never developed for the case of MET-LDPC codes. In Section 4.3 we denoted how it is possible to define and plot the EXIT and GEXIT charts for MET-LDPC codes. We started with BE channel and then the general formulation for BIOSM channel was presented.

- Developed a new algorithmic approach to design highly efficient MET-LDPC codes. The optimization problem for MET-LDPC codes is presented in Section 4.4. Then an algorithmic approach is introduced to design highly efficient MET-LDPC codes. For more details about our design procedure see Section 4.5.

- Published a set of new high efficient MET-LDPC codes for reconciliation process. These codes were optimized to have maximum threshold. A comparison of the asymptotic efficiency of our designed codes and some the best codes available in literature are presented in Table 5.1.

Table 5.1: Degree distribution for some MET-LDPC code ensembles designed by our proposed algorithm and comparison with other existing codes.

<table>
<tr><td colspan="5">BIAWGN channel, $R = 0.01$, $\sigma^*_{Sh} = 8.46$</td></tr>
<tr><td>Ref</td><td>$\nu(\mathbf{r}, \mathbf{x})$</td><td>$\mu(\mathbf{x})$</td><td>$\sigma^*_{DE}$</td><td>Efficiency</td></tr>
<tr><td></td><td>$0.01\ r_1x_1^2x_2^{103} + 0.01\ r_1x_1^3x_2^{125} + 0.98\ r_1x_3^1$</td><td>$0.008\ x_1^4 + 0.002\ x_1^9 + 0.32\ x_2^3x_3^1 +$ $0.66\ x_2^2x_3^1$</td><td>8.41</td><td>98.70%</td></tr>
<tr><td colspan="5">BIAWGN channel, $R = 0.02$, $\sigma^*_{Sh} = 5.96$</td></tr>
<tr><td>Ref</td><td>$\nu(\mathbf{r}, \mathbf{x})$</td><td>$\mu(\mathbf{x})$</td><td>$\sigma^*_{DE}$</td><td>Efficiency</td></tr>
<tr><td></td><td>$0.02\ r_1x_1^2x_2^{51} + 0.02\ r_1x_1^3x_2^{60} + 0.96\ r_1x_3^1$</td><td>$0.016\ x_1^4 + 0.004\ x_1^9 + 0.3\ x_2^3x_3^1 +$ $0.66\ x_2^2x_3^1$</td><td>5.94</td><td>99.20%</td></tr>
<tr><td></td><td>$0.024\ r_1x_1^2x_2^{43} + 0.016\ r_1x_1^3x_2^{61} + 0.96\ r_1x_3^1$</td><td>$0.002\ x_1^3 + 0.018\ x_1^5 + 0.088\ x_2^3x_3^1 +$ $0.872\ x_2^2x_3^1$</td><td>5.82</td><td>95.52%</td></tr>
<tr><td>[10]</td><td>$0.0225\ r_1x_1^2x_2^{57} + 0.0175\ r_1x_1^3x_2^{57} + 0.96\ r_1x_3^1$</td><td>$0.010625\ x_1^3 + 0.009375\ x_1^7 + 0.6\ x_2^2x_3^1 +$ $0.36\ x_2^3x_3^1$</td><td>5.91</td><td>98.22%</td></tr>
<tr><td colspan="5">BIAWGN channel, $R = 0.05$, $\sigma^*_{Sh} = 3.73$</td></tr>
<tr><td></td><td>$0.054\ r_1x_1^2x_2^{22} + 0.046\ r_1x_1^3x_2^{22} + 0.90\ r_1x_3^1$</td><td>$0.026\ x_1^3 + 0.024\ x_1^7 + 0.40\ x_2^3x_3^1 +$ $0.50\ x_2^2x_3^1$</td><td>3.68</td><td>97.36%</td></tr>
<tr><td>[45]</td><td>$0.04\ r_1x_1^2x_2^{34} + 0.03\ r_1x_1^3x_2^{34} + 0.93\ r_1x_3^1$</td><td>$0.01\ x_1^8 + 0.01\ x_1^9 + 0.41\ x_2^2x_3^1 +$ $0.52\ x_2^3x_3^1$</td><td>3.67</td><td>96.78%</td></tr>
<tr><td colspan="5">BIAWGN channel, $R = 0.1$, $\sigma^*_{Sh} = 2.6$</td></tr>
<tr><td></td><td>$0.075\ r_1x_1^2x_2^{23} + 0.05\ r_1x_1^3x_2^{18} + 0.875\ r_1x_3^1$</td><td>$0.025\ x_1^{12} + 0.875\ x_2^3x_3^1$</td><td>2.57</td><td>98.37%</td></tr>
<tr><td>[12]</td><td>$0.1063\ r_1x_1^3x_2^{23} + 0.0216\ r_1x_1^2x_2^8 + 0.8722\ r_1x_3^1$</td><td>$0.0278\ x_1^{13} + 0.8722\ x_2^3x_3^1$</td><td>2.55</td><td>96.95%</td></tr>
<tr><td>[13]</td><td>$0.0775\ r_1x_1^1x_2^1x_3^{21} + 0.0477\ r_1x_1^2x_2^1x_3^{20} + 0.8747\ r_1x_4^1$</td><td>$0.0011\ x_1^6x_2^4 + 0.0028\ x_1^6x_2^5 + 0.0214\ x_1^7x_2^5 +$ $0.0412\ x_3^2x_4 + 0.8335\ x_3^3x_4$</td><td>2.5424</td><td>96.41%</td></tr>
<tr><td colspan="5">BIAWGN channel, $R = 0.25$, $\sigma^*_{Sh} = 1.55$</td></tr>
<tr><td></td><td>$0.1875\ r_1x_1^2x_2^{12} + 0.125\ r_1x_1^3x_2^4 + 0.6875\ r_1x_3^1$</td><td>$0.0625\ x_1^{12} + 0.6875\ x_2^4x_3^1$</td><td>1.5031</td><td>95.07%</td></tr>
<tr><td>[14]</td><td>$0.1512\ r_1x_1^3 + 0.006\ r_1x_1^4 + 0.0928\ r_1x_1^{12} +$ $0.75\ r_1x_2^2$</td><td>$0.6588\ x_1^2x_2^2 + 0.0912\ x_1^3x_2^2$</td><td>1.4839</td><td>93.08%</td></tr>
<tr><td>[12]</td><td>$0.1563\ r_0x_1^4x_3^4x_4^1 + 0.0938\ r_1x_3^2x_4^1 + 0.9062\ r_1x_2^1x_4^1$</td><td>$0.3125\ x_1^2x_2^1 + 0.1875\ x_2^1x_4^4 + 0.4063\ x_2^1x_3^2x_4^1$</td><td>1.4894</td><td>93.65%</td></tr>
</table>

- Developed a set of c++ tools for analyzing and designing MET-LDPC codes. The details of these tools can be found in Appendix B and Appendix C.

- Implemented the reconciliation scheme for CV-QKD based on MLC-MSD method, in C++. More details can be found in Appendix B.

Findings of this research have considerable applications for long distance QKD. To further our research we are planning to investigate the potential advantages of MLC-MSD reconciliation and resolve some of the existing limitations in both multi-dimensional reconciliation and MLC-MSD scheme. In addition, further studies which take high-throughput reconciliation into account will need to be considered. We are currently in the process of investigating high throughput reconciliation scheme, with two possible options. First, we are trying to implement an efficient decoder which is specifically designed for the MET-LDPC codes. Since, a big portion of the MET-LDPC codes are degree one nodes, it can be very useful to tailor a decoding algorithm specifically for MET-LDPC codes instead of using min-sum algorithm or other variants of decoding algorithm which is typically used for the irregular LDPC codes. In addition, graphics processing unit (GPU) or Field Programmable Gate Arrays (FPGA) implementations should be considered in future. Second, we are investigating to design new reconciliation methods with higher number of bits per symbol. For example, the multi-dimensional reconciliation scheme extracts one bit per symbol, but in contrast the MLC-MSD scheme in principle is able to extract more than one bit per symbol. Also, other variants of the reconciliation algorithms are considered. For example, our new proposed stochastic reconciliation scheme can be used to increase the throughput while decreasing the complexity of the decoding.

Besides, the code design algorithm can be developed to optimize different cost functions. Our current algorithm optimizes the codes to have the best asymptotic thresholds. In practice the best MET-LDPC codes with finite block length can be considered to have the best decoding performance with minimum decoding iteration.

APPENDIX A

Introduction to LDPC codes

A.1 Preliminaries

This section is technical and some basic definitions and examples are used to describe the basic concepts. We advise to quickly skim it to see what materials are discussed but a detailed study is not recommended. At any later point when needed, you can return and find the specific materials. For a more detailed introduction into these concepts we refer reader to [44].

A.1.1 Channel model

Let us denote by $\mathcal{X}$ the channel input alphabet and $\mathcal{Y}$ the channel output alphabet. Then the conditional density $f_{Y|X}(y|x)$ completely describes the channel model with random variables X and Y. For discrete channel models we simply replace the conditional density by the conditional discrete probability $\Pr_{Y|X}(y|x)$. In addition let us clarify that the lower case letter $x \in \mathcal{X}$ denotes a specific outcome with probability $f_X(x)$ of random variable X. If $|\mathcal{X}| = 2$ the channel is known as binary input channel. Conventionally the binary input alphabets are $\mathcal{X} = \{-1, +1\}$ or $\mathcal{X} = \{1, 0\}$.

Binary erasure channel (BEC) The binary erasure channel with parameter ϵ is denoted as BEC(ϵ). The random variable X at the input can take values $x \in \mathcal{X} = \{-1, \ +1\}$ and the output random variable can take values from the output alphabet $\mathcal{Y} = \{-1, \ ?, \ +1\}$. The transition probability is discrete and equal to

$$\Pr_{Y|X}(y|x) = \begin{cases} 1-\epsilon \ , & y = x \ , \\ \epsilon \ , & y = ? \ , \\ 0 \ , & \text{otherwise} \ . \end{cases}$$

Binary symmetric channel (BSC) The binary symmetric channel with parameter δ is denoted as BSC(δ). The random variable X at input can take values $x \in \mathcal{X} = \{-1, \ +1\}$ and the output random variable can take values from the same alphabet $\mathcal{Y} = \{-1,\ +1\}$. The transition probability is discrete and equal to

$$\Pr_{Y|X}(y|x) = \begin{cases} 1-\delta\ , & y = x\ , \\ \delta\ , & \text{otherwise}\ . \end{cases}$$

Binary input additive white Gaussian noise channel (BI-AWGN) The binary input additive white Gaussian noise channel with parameter σ is denoted as BI-AWGN(σ). The channel noise model is described by Gaussian noise with zero mean and standard deviation σ. The random variable X at input can take values $x \in \mathcal{X} = \{-1,\ +1\}$ and the output random variable can take values from the real-valued numbers $\mathcal{Y} = \mathbb{R}$. The transition probability density function is equal to

$$f_{Y|X}(y|x) = \frac{e^{-\dfrac{(y-x)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\ .$$

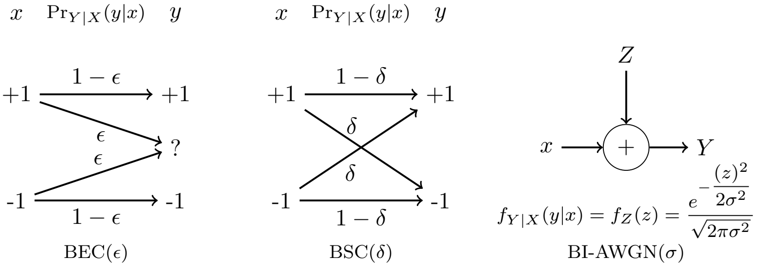


Figure A.1: The schematic representation of standard BIOSM channels.

Binary-input output-symmetric memory-less channels A channel is said to be output-symmetric if and only if $f_{Y|X}(y|x) = f_{Y|X}(-y|-x)$. It can be easily shown that all the channels introduced above are output-symmetric. It is convenient to call this family of channels the *binary-input output-symmetric memory-less* (BIOSM) channels.

In addition, the channel parameter p can be defined for all of the above mentioned BIOSM($\mathtt{p}$) channels. For example the channels parameters for BEC, BSC and BI-AWGN channels are $\epsilon \in [0,\ 1]$, $\delta \in [0,\ 0.5]$ and $\sigma \in [0,\ \infty]$. Later we demonstrate how the channel parameters can be used to calculate the channel entropy $H(X|Y)$.

Log-Likelihood Ratio (LLR) Consider a BIOSM channel with transition probability $\Pr_{Y|X}(y|x)$. The log-likelihood ratio (LLR) function $l(y)$ is defined as

$$l(y) = \ln \frac{\Pr_{Y|X}(y|1)}{\Pr_{Y|X}(y|-1)}\ .$$

The log-likelihood ratio associated to the random variable Y is defined as $L = l(Y)$. $l(y)$ is said to be the output of the channel in the L-domain. It is important to notice that L is a random variable itself.

In [44] the authors demonstrated that L is a *sufficient statistic* for decoding. It means that an optimal decoder can be based on the LLR $l(y)$ instead of y itself.

Definition of the L-Density $\mathtt{a}(y)$ The distribution of the log-likelihood ratios $L = l(Y)$ of the BIOSM channels conditioned on $X = 1$ is defined as the *L-density*. Here we represent the L-densities for all the BIOSM channels

Table A.1: The distribution of the L conditioned on the $X = 1$.

BIOSM($\mathtt{p}$)	**L-Density** $\mathtt{a}(y)$	**Description**
BEC(ϵ)	$\mathtt{a}(y) = \epsilon\ \Delta_0(y) + \bar{\epsilon}\ \Delta_\infty(y)$	$\bar{\epsilon} = 1 - \epsilon$
BSC(δ)	$\mathtt{a}(y) = \delta\ \Delta_{\frac{\delta}{\bar{\delta}}}(y) + \bar{\delta}\ \Delta_{\frac{\bar{\delta}}{\delta}}(y)$	$\bar{\delta} = 1 - \delta$
BI-AWGNC(σ)	$\mathtt{a}(y) = \sqrt{\frac{\sigma^2}{8\pi}}\ e^{-\frac{(y - \frac{2}{\sigma^2})^2\sigma^2}{8}}$	

The symmetry of the probability density function Let $f(x)$ be a probability density function over $\mathbb{R}$. The density is said to be symmetric if $f(-x) = e^{-x}f(x)$. All the L-densities in Table A.1 are symmetric.

A.2 Channel entropy and channel capacity

Consider a BIOSM channel with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$ and transition probability $\Pr_{Y|X}(y|x)$. Also assume that $\mathtt{a}(y)$ denotes the L-density of this channel. Then according to [44]-Lemma 4.36 the capacity of this channel is a linear functional of its L-Density and can be represented as

$$C(\mathtt{a}) = \int \mathtt{a}(y)(1 - \log_2(1 + e^{-y}))\ dy\ , \tag{A.1}$$

where $C(\mathtt{a})$ is the *capacity functional* of the BIOSM channels in bits. Using the fact that for symmetric channels the optimal input distribution is uniform and the input alphabet has equal prior probability $\Pr_X(x) = \frac{1}{2}$, for $x \in \{-1, +1\}$ then the conditional entropy $H(X|Y)$ can be defined as

$$H(X|Y) = H(X) - I(X;Y) = H(X) - C(\mathsf{a}) = \int \mathsf{a}(y) \log_2(1 + e^{-y})\ dy\ . \tag{A.2}$$

In addition let us call $\mathtt{h} = H(\mathsf{a}_{\mathrm{BIOSMC}})$ the *entropy functional* for a BIOSMC. Later in this chapter we will see how these functions can play an important role in the definition of the extrinsic-information transfer (EXIT) function and how we use those to design highly efficient degree distribution for multi-edge-type low-density parity-check (MET-LDPC) codes.

Example A.2.1. *Capacity functional for BSC and BEC*

Here we use (A.1) to find the capacity of the BEC(ϵ) and BSC(δ) channels. For BEC(ϵ) we have

$$C(\mathsf{a}_{\mathrm{BEC}(\epsilon)}) = 1 - \int_{-\infty}^{\infty} [\epsilon\ \Delta_0(y) + (1-\epsilon)\ \Delta_\infty(y)](\log_2(1 + e^{-y}))\ dy\ = 1 - \epsilon\ .$$

In similar way for the BSC(δ)

$$C(\mathsf{a}_{\mathrm{BSC}(\delta)}) = 1 - \int_{-\infty}^{\infty} [\delta\ \Delta_{\frac{\delta}{1-\delta}}(y) + (1-\delta)\ \Delta_{\frac{1-\delta}{\delta}}(y)](\log_2(1 + e^{-y}))\ dy\ = 1 - h_2(\delta),$$

where $h_2(\delta) = -\delta \log_2(\delta) - (1-\delta)\log_2(1-\delta)$.

Example A.2.2. *Capacity functional for BI-AWGNC*

To calculate the capacity for the BI-AWGNC the following integral has to be solved numerically:

$$C(\mathsf{a}_{\text{BI-AWGN}(\sigma)}) = 1 - \int_{-\infty}^{\infty} \sqrt{\frac{\sigma^2}{8\pi}}\ e^{-\dfrac{(y - \frac{2}{\sigma^2})^2\sigma^2}{8}} (\log_2(1 + e^{-y}))\ dy\ ,$$

In Figure A.3 the capacity of the BI-AWGN channel is compared with the capacity of the AWGN channel with real-valued inputs.

A.3 Low-density parity-check (LDPC) codes

The LDPC codes are linear codes with a sparse parity check matrix H. Assuming H has dimensions $m \times n$ then it is possible to assign a *bipartite* graph to this matrix, named Tanner graph. The associated Tanner graph for this H has n *variable nodes* corresponding to the codeword of length n (columns of H) and $m = n-k$ *check-nodes*

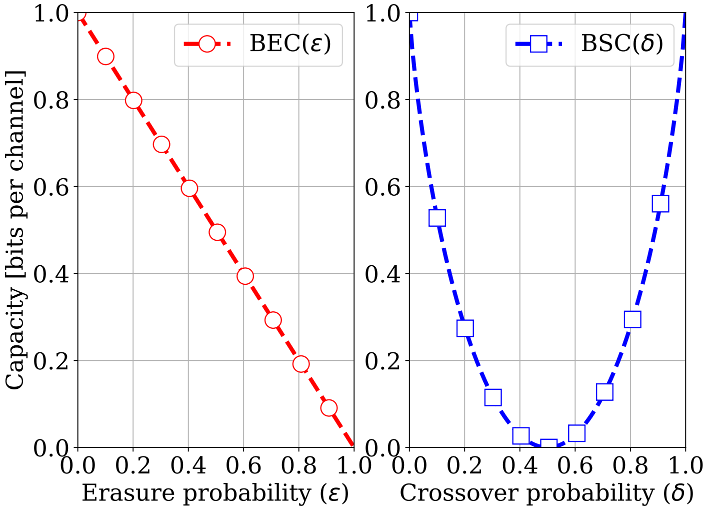


Figure A.2: The capacity of the BEC and BSC channels.

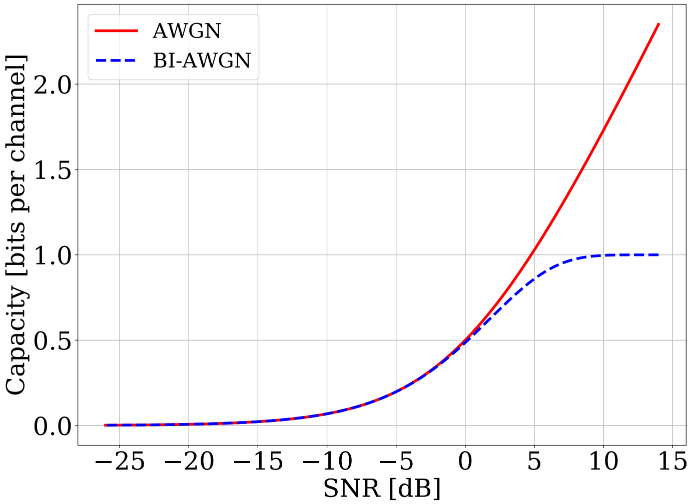


Figure A.3: The capacity of the BI-AWGN and AWGN channels.

corresponding to the set of $n-k$ parity constraints (row of H). k shows the number of information bits. The check-node j is connected to the variable-node i if $H(j,i)=1$.

LDPC codes can be represented by their degree distribution. The *node perspective* degree distribution of the LDPC code is given by

$$\nu(x)=\sum_{i=1}\nu_i\ x^i, \qquad \mu(x)=\sum_{i=1}\mu_i\ x^i\ ,$$

where ν_i is the portion of the variable-nodes of degree i and the μ_i is the portion of check-nodes of degree i in the Tanner graph. Considering this definition the following relations are valid

$$\nu(1)=1\ , \quad \mu(1)=1-r\ , \quad \nu'(1)=\mu'(1)\ .$$

The corresponding *edge-perspective* degree distribution of this code is

$$\lambda(x)=\sum_{i=1}\lambda_i\ x^{i-1}=\frac{\nu'(x)}{\nu'(1)}\ , \qquad \rho(x)=\sum_{i=1}\rho_i\ x^{i-1}=\frac{\mu'(x)}{\mu'(1)}\ ,$$

where λ_i (ρ_i) shows the fraction of edges that are connected to the variable-node (check-node) of degree i. The node perspective representation can be obtained from edge perspective representation as follows:

$$\nu(x)=\frac{\int_0^x\lambda(z)dz}{\int_0^1\lambda(z)dz}\ , \qquad \mu(x)=\frac{\int_0^x\rho(z)dz}{\int_0^1\lambda(z)dz}\ ,$$

Furthermore, the following relation holds

$$r=\nu(1)-\mu(1)=\ 1-\frac{\int_0^1\rho(z)dz}{\int_0^1\lambda(z)dz}\ .$$

A.4 Belief Propagation

Consider a LDPC code with degree distribution (λ, ρ) on a BIOSM channel. In addition assume that $\mathsf{a}_0=\mathsf{a}_{\mathrm{BIOSMC}}$ denotes the corresponding L-density at iteration 0. Then for iteration $l\ \geq\ 1$

$$\mathsf{a}^l=\mathsf{a}_0\ \otimes\ \lambda\left(\rho\left(\mathsf{a}^{l-1}\right)\right)\ , \tag{A.3}$$

where

$$\lambda(\mathsf{a})=\sum_{i=1}\lambda_i\ \mathsf{a}^{\otimes(i-1)}\ , \qquad \rho(\mathsf{a})=\sum_{i=1}\rho_i\ \mathsf{a}^{\boxtimes(i-1)}\ .$$

Sometimes it is convenient to write the densities in terms of variable nodes and check nodes:

$$\mathsf{a}_v^l=\mathsf{a}_0\ \otimes\ \lambda(\mathsf{a}_c^{l-1})\ , \qquad \mathsf{a}_c^l=\rho(\mathsf{a}_v^l)\ . \tag{A.4}$$

For more details about BP please see [44]-Section 2.5.

A.5 EXIT and G-EXIT charts

Before introducing EXIT charts we present some useful definitions and then show how we can exploit those for plotting EXIT and G-EXIT charts.

EXIT functional According to [44]-DEFINITION 4.43 the *EXIT functional* is defined as the entropy functional, see (A.2). The associated EXIT kernel in L-domain is $l(y) = \log_2(1 + e^{-y})$.

G-EXIT functional Again according to [44]-DEFINITION 4.44 the *Generalized EXIT (G-EXIT)* functional for a BIOSM($\mathtt{h}$) applied to a symmetric L-density $\mathtt{b}$ is defined as

$$G(\mathtt{a}_{\mathrm{BIOSMC}(\mathtt{h})}, \mathtt{b}) = \frac{d}{dh} H(\mathtt{a}_{\mathrm{BIOSMC}(\mathtt{h})} \ \otimes \ \mathtt{b}) \ ,$$

and the corresponding G-EXIT kernel is defined as

$$l(y) = \int \frac{d\mathtt{a}_{\mathrm{BIOSMC}(\mathtt{h})}}{dh} \ \log_2(1 + e^{-z-y}) dz \ .$$

Sometimes it is convenient to consider a family of BIOSMC with parameter $\mathtt{h}$. It means that we parameterize the channel with the entropy. For example for BEC $\mathtt{h} = \epsilon$, for BSC $\mathtt{h} = h_2(\delta)$ and and for BI-AWGNC $\mathtt{h} = H(\mathtt{a}_{\text{BI-AWGNC}(\sigma)})$.

EXIT function According to [44]-DEFINITION 4.131, let X be a vector of length n chosen uniformly at random from a binary code C, and Y be the transmitted version of X over the BIOSMC($\mathtt{h}$). Then the individual and average EXIT functions are

$$\begin{aligned} h_i(\mathtt{h}) &= H(X_i | Y_{\sim i}(\mathtt{h})) \ , \\ h(\mathtt{h}) &= \frac{1}{n} \sum_{i=1}^{n} \ H(X_i | Y_{\sim i}(\mathtt{h})) = \frac{1}{n} \sum_{i=1}^{n} \ h_i(\mathtt{h}) \ . \end{aligned}$$

G-EXIT function According to [44]-DEFINITION 4.154, let X be a vector of length n chosen uniformly at random from a binary code C and Y be the transmitted version of X over the BIOSMC($\mathtt{h}$). Then the individual and average G-EXIT functions are

$$\begin{aligned} g_i(\mathtt{h}) = g_i(\mathtt{h}_1, \cdots, \mathtt{h}_n) &= \frac{\partial H(X_i | Y(\mathtt{h}_1, \cdots, \mathtt{h}_n))}{\partial \mathtt{h}_i} \ , \\ g(\mathtt{h}) &= \frac{1}{n} \sum_{i=1}^{n} g_i(\mathtt{h}_1, \cdots, \mathtt{h}_n) \ . \end{aligned}$$

Now, in the following, we show how it is possible to calculate the EXIT and G-EXIT functions for LDPC codes. First we need to find the L-density of the extrinsic maximum a-posteriori (MAP) estimator.

Extrinsic MAP estimator Let X be a vector of length n chosen uniformly at random from a binary code C and let Y be the transmitted version of X over the BIOSMC($\mathtt{h}$). Then the extrinsic MAP estimator of X_i is

$$\phi_i(y_{\sim i}) = \ln \frac{\Pr_{X_i|Y_{\sim i}}(+1|y_{\sim i})}{\Pr_{X_i|Y_{\sim i}}(-1|y_{\sim i})} \ ,$$

and $\Phi_i = \phi_i(Y_{\sim i})$. Φ_i is a sufficient statistics for X_i given $Y_{\sim i}$. $H(X_i|Y_{\sim i}(\mathtt{h})) = H(X_i|\Phi_i)$.

We now calculate the EXIT function for linear codes using the EXIT functional of the extrinsic MAP estimator. Let $\mathtt{a}_i$ denotes the L-density of Φ_i and assume that an all-one-codeword was transmitted. With $\mathtt{a} = \frac{1}{n}\sum_{i=1}^{n} \mathtt{a}_i$

$$h_i(\mathtt{h}) = H(\mathtt{a}_i) \ , \qquad h(\mathtt{h}) = H(\mathtt{a}) \ .$$

and for the G-EXIT functional

$$\begin{aligned} g_i(\mathtt{h}_1 \cdots , \mathtt{h}_n) &= \frac{\partial H(\mathtt{a}_{\text{BIOSMC}(\mathtt{h}_i)} \otimes \mathtt{a}_i)}{\partial \mathtt{h}_i} \\ &= G(\mathtt{a}_{\text{BIOSMC}(\mathtt{h}_i)}, \ \mathtt{a}_i) = \int \ \mathtt{a}_i(y) \ l^{\mathtt{a}_{\text{BIOSMC}(\mathtt{h}_i)}}(y)dy \ . \end{aligned}$$

where

$$l^{\mathtt{a}_{\text{BIOSMC}(\mathtt{h})}}(y) = \int \ \frac{d\mathtt{a}_{\text{BIOSMC}(\mathtt{h})}(z)}{d\mathtt{h}} \ \log_2(1 + e^{-z-y})dz \ .$$

A.6 Plotting EXIT and G-EXIT Charts

As presented in (A.3), $\mathtt{a}_0 = \mathtt{a}_{\text{BIOSMC}}$ and after $l \geq 1$, $\mathtt{a}^l = \mathtt{a}_{\text{BIOSMC}} \ \otimes \ \lambda(\rho(\mathtt{a}^{l-1}))$. Unfortunately the *intermediate* L-densities $\mathtt{a}_l$ do no have simple descriptions, but estimating them with some equivalent density families, we can apply the EXIT functional (G-EXIT functional) to obtain the EXIT (G-EXIT) chart. As presented in [44] the most *faithful* equivalence rule is to choose the element of the channel family which has *equal entropy.*

Now, assume that for a pair of $(\lambda, \ \rho)$, we were able to guess the true intermediate L-densities. As presented in (A.4), $\mathtt{a}_v^l$ ($\mathtt{a}_c^l$) is the density emitted at the variable-node (check-node) at iteration l. Then, using the entropy functional

$$\mathtt{h}_c^l = H(\rho(\mathtt{a}_v^l)) \ , \qquad \mathtt{h}_v^l = H(\mathtt{a}_{\text{BIOSMC}} \ \otimes \ \lambda(\mathtt{a}_c^{l-1})) \ .$$

The EXIT curves can be obtained by plotting the entropy values for the input-output of the variable nodes and check nodes. The EXIT chart for a given pair of (λ, ρ) is realized by plotting the EXIT curve of a variable-node against the inverse of a check-node. The parametric form of the EXIT curve for the check-node is given by $\{\mathtt{h}_v^l, \mathtt{h}_c^l\}$, and the inverse of the EXIT curve for the variable-node is $\{\mathtt{h}_v^l, \mathtt{h}_c^{l-1}\}$. Finally, the EXIT chart is then given by

Table A.2: The parametric representation of an EXIT chart.

EXIT curve of check-node	$\left\{\mathtt{h}_v^l = H(\mathtt{a}_v^l),\ \mathtt{h}_c^l = H(\mathtt{a}_c^l)\right\}$
Inverse of EXIT curve of variable-node	$\left\{\mathtt{h}_v^l = H(\mathtt{a}_v^l),\ \mathtt{h}_c^{l-1} = H(\mathtt{a}_c^{l-1})\right\}$

A similar concept can be used for G-EXIT curves. In particular, in [66] the G-EXIT chart is realized by plotting the inverse of the dual G-EXIT curve of a variable-node against the G-EXIT curve of a check-node. For a pair of densities the G-EXIT curve of the check-node can be given in the parametric form by $\left\{\mathtt{h}_v^l = H(\mathtt{a}_v^l), G(\mathtt{a}_v^l, \mathtt{a}_c^l)\right\}$. Similarly the dual G-EXIT for the variable-node can be written in parametric form as $\left\{G(\mathtt{a}_v^l, \mathtt{a}_c^{l-1}), \mathtt{h}_v^l = H(\mathtt{a}_v^l)\right\}$ and the inverse of the dual G-EXIT curve is then $\left\{\mathtt{h}_v^l = H(\mathtt{a}_v^l), G(\mathtt{a}_v^l, \mathtt{a}_c^{l-1})\right\}$. Thus, the G-EXIT chart is given by

Table A.3: The parametric representation of G-EXIT chart.

G-EXIT curve of check-node	$\left\{\mathtt{h}_v^l = H(\mathtt{a}_v^l),\ G(\mathtt{a}_v^l, \mathtt{a}_c^l)\right\}$
Inverse of the dual G-EXIT curve of variable-node	$\left\{\mathtt{h}_v^l = H(\mathtt{a}_v^l),\ G(\mathtt{a}_v^l, \mathtt{a}_c^{l-1})\right\}$

APPENDIX B

Reconciliation software for CV-QKD

In this appendix you will find some useful information about the tools related to the reconciliation of the CV-QKD. All the codes are available on a private git repository. If you need to get access to the repository you can send email to Prof. Tobias Gehring. All the tools have a shell script which helps you to start the simulation. For any technical help regarding the software contact me.

B.1 About the Software

The repository contains different tools required for the reconciliation process of CV-QKD. It it not limited to but contains the following folders:

MET-DE The density evolution for MET-LDPC codes.

Gen-LDPC Generates the parity check matrix for a MET-LDPC code.

Sim-LDPC Checks the performance of the MET-LDPC codes on BI-AWGN channel.

MLC-MSD The reconciliation for the MLC-MSD scheme.

Stochastic Chase Reconciliation using the stochastic Chase algorithm.

MET-DE: Density evolution for the MET-LDPC codes. The density evolution is an asymptotic analysis tool for the LDPC codes. Here we developed a shared library named “`libmetawgnde.so`”. This library is designed for MET-LDPC codes which are a generalized version of the irregular LDPC codes. The algorithm is based on an efficient implementation of the variable nodes and check nodes operations according to [44]. For the variable node operations we use a fast Fourier Transform using the “`fftw3`” library and for the fast implementation of the check node operations the Table-method is used according to [44].

This tool is able to read the structure of an MET-LDPC code and checks the asymptotic error probability after l iterations on a BI-AWGN channel under the assumption of a all-zero codeword transmission. The software accepts a "`structure.txt`" file which is equivalent to the table format representation of a MET-LDPC code (See Table 4.8) and contains :

```
    3    # ne : The number of edges
    2    # nr : The number of channels
    12   # number of bits for digitization
    25   # max LLR
    1500 # maximum number of iterations
    -1.1 # SNR value : Eb/N0
    3    # the length of v_bd vector
    4    # the length of mu_d vector
    0.010625  0.009375  0.6 0.36   # The mu_d vector
    0.0225  0.0175  0.96           # The v_bd vector
    # The variable node degrees matrix in vector format :
    2 57  0 3 57  0 0 0 1
    # The Check node degrees matrix in vector format :
    3 0 0 7 0 0 0 2 1 0 3 1
    # The channel node degrees matrix in vector format
    0 1 0 1 0 1
    1e-10    # minimum error probability
```

Listing B.1: The input file for the code structure.

Gen-LDPC This tool generates a codec file "`codec.it`" which can be used for decoding. It accepts a description of LDPC codes as an input file and generates a codec file. Two common input formats are "`*.alist`" and "`*.peg`". The first is usually used for the description of an irregular LDPC code and the second is used to describe Quaci-cyclic LDPC (QC-LDPC) codes.

Basically the "`*.alist`" file describes a parity check matrix. To read more about the sparse codes and the alist format see `http://www.inference.org.uk/mackay/codes/data.html`.

Sim-LDPC After generating a codec file this tool simulates the performance of the code on BI-AWGN channel by plotting the bit-error-rate (BER) and frame-error-rate (FER) of an arbitrary MET-LDPC code. The performance of the code can be simulated for a single specific signal-to-noise (SNR) or for a range of SNRs. By default it counts 100 frame of errors for each single SNR value and the default iteration number is 25.

MLC-MSD The reconciliation of a CV-QKD system based on multi-level-coding multi-stage-decoding method using MET-LDPC codes. It contains two different implementations including *single-level reconciliation* and *two-levels reconciliation*. The corresponding software accepts a set of real valued data corresponding to Alice's and Bob's data after the quantum phase. Then an appropriate code rate will be chosen according to the SNR of the raw data for each level. The calculation of the individual code rates are discussed in Chapter 3.

Stochastic Chase Stochastic Chase decoder for reverse reconciliation. As explained in Chapter 3 a hard LDPC decoder can be used instead of a soft decoder for the reconciliation purpose with the goal of obtaining performance close to the soft decoder but with higher throughput. The implementation of this tool is not completely finished yet.

B.2 How to install the software

To be able to work with the above tools the IT++ library is required. IT++ is an open source C++ library for signal processing and communication. It includes an efficient implementation of the forward error correction codes. For more information see :

`http://itpp.sourceforge.net/4.3.1/`.

Before installing the IT++ make sure that LAPACK and BLAS are available on your system.

```
# First, you have to install BLAS before LAPACK, because LAPACK needs it.
# Download BLAS and Extract it
  wget http://www.netlib.org/blas/blas-3.8.0.tgz
  tar jxf blas-3.8.0.tgz
  cd blas-3.8.0
  make
# Rename the library to libblas.a and copy it to your local library
  mv blas_UNIX.a   libblas.a
  sudo cp libblas.a   /usr/local/lib/
# Now we have installed the BLAS package. Let's get LAPACK
  wget https://github.com/Reference-LAPACK/lapack/archive/v3.9.0.tar.gz
  tar jxf lapack-3.9.0.tar.gz
  cd lapack-3.9.0
# Adjust the file "make.inc."example to address the BLAS. Find the line that reads
    BLASLIB = ../../librefblas.a
# and change it to :
    BLASLIB = /usr/local/lib/libblas.a
# Save this file as "make.inc" and
  make
  sudo cp ./liblapack.a  /usr/local/lib/
```

Listing B.2: LAPACK and BLAS liraries.

The official installation guide for IT++ is not friendly. Here is a simpler way to install it:

```
# Download itpp and extract the file
wget https://netcologne.dl.sourceforge.net/project/itpp/itpp/4.3.1/itpp-4.3.1.tar.bz2
tar jxf itpp-4.3.1.tar.bz2
```

```
cd itpp-4.3.1
# clean it before install
rm cmake/Modules/FindBLAS.cmake cmake/Modules/FindLAPACK.cmake
# make a directory at your desired destination
mkdir $HOME/itpp-4.3.1
mkdir build && cd build
BLA_VENDOR=OpenBLAS CC=gcc CXX=g++
# cmake version at least 3.16.5
cmake -DCMAKE_INSTALL_PREFIX=$HOME/itpp-4.3.1 ..
make
make install
```

Listing B.3: Installation of IT++.

APPENDIX C Software to design highly efficient MET-LDPC codes

C.1 About the software

This tool helps to optimize a degree distribution for MET-LDPC codes with cascade structure. Details of the optimization problem for cascade structures is described in Chapter 4. The goal is to design a MET-LDPC code with rate R when a base code with rate r is available. The optimization of the base code with rate r was described in Section 4.4. Here we assume that a base code is available and the goal of the optimization problem is then to find the best degree distribution for the cascade structure. The optimization software consists of three different tools:

EXIT-Chart Plots the EXIT chart for the base code.

Connector Generates a list of all possible connector parts in the cascade structure.

Find best Finds the best degree distribution by testing different connector parts.

EXIT-Chart This tool provides a quick pictorial representation for the base code. For a given base code with rate r_b it generates a `EXIT.txt` file which can be used by a python script (`plotEXIT.py`) to generate the EXIT chart. The code is designed for the BEC(`q`) channel. For example, for a base code of rate 0.8, the corresponding EXIT chart for BEC(0.15) is plotted in Figure C.1. The node perspective degree distribution of this code is

$$\nu(q,x) = \ 0.6\ q\ x^2 + \ 0.4\ q\ x^3\ , \qquad \mu(x) = 0.2\ x^{12}\ .$$

The corresponding edge perspective degree distribution is

$$\lambda(q,x) = \ 0.5\ q\ x + \ 0.5\ q\ x^2\ , \qquad \rho(x) = x^{11}\ .$$

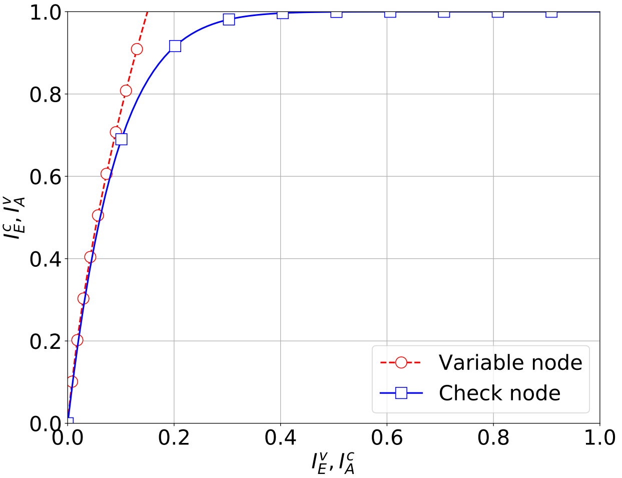


Figure C.1: The output of the EXIT tool for a base code of rate 0.8 on BEC(0.15).

Connector This tool generates a `DEGREE-x.txt` file which contains different code structures (See Listing B.1) for a MET-LDPC code with base code of rate r_b and overall code of rate R. If the number of candidate codewords is more than 1000 a new file with appropriate numbering will be generated.

As input it accepts the structure of the base code in the node perspective mode and the overall code rate of the MET-LDPC code. It also accepts some further information required for the density evolution. The parameters are presented in Listing B.1.

This tool checks the constraint sets for the candidate codes and only generates codes with the accepted structures. For more details to see how the candidate codes are generated based on a base code see Section 4.4.

Find Best This tool accepts `DEGREE-x.txt` as the input and then checks the performance of the codes to find the best code in forms of the threshold. It runs DE for all the candidate codes and from all codes which converged with a specified number of iterations, it selects the best code.

Bibliography

[1] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.” In: *SIAM J. Comput.* 26.5 (October 1997), pages 1484–1509. ISSN: 0097-5397. DOI: 10.1137/S0097539795293172. URL: http://dx.doi.org/10.1137/S0097539795293172.

[2] Valerio Scarani et al. “The security of practical quantum key distribution.” In: *Rev. Mod. Phys.* 81 (September 2009), pages 1301–1350. DOI: 10.1103/RevModPhys.81.1301. URL: https://link.aps.org/doi/10.1103/RevModPhys.81.1301.

[3] Nicolas Gisin et al. “Quantum cryptography.” In: *Rev. Mod. Phys.* 74 (1 March 2002), pages 145–195. DOI: 10.1103/RevModPhys.74.145. URL: https://link.aps.org/doi/10.1103/RevModPhys.74.145.

[4] Yi-Chen Zhang et al. “Long-distance continuous-variable quantum key distribution over 202.81 km fiber.” In: *Physical Review Letters* (June 2020). URL: http://eprints.whiterose.ac.uk/161908/.

[5] Boris Korzh et al. “Provably secure and practical quantum key distribution over 307 km of optical fibre.” In: *Nature Photonics* 9.3 (March 2015), pages 163–168. DOI: 10.1038/nphoton.2014.327. arXiv: 1407.7427 [quant-ph].

[6] Anthony Leverrier et al. “Multidimensional reconciliation for a continuous-variable quantum key distribution.” In: *Phys. Rev. A* 77.4 (2008), page 042325. DOI: 10.1103/PhysRevA.77.042325. URL: https://link.aps.org/doi/10.1103/PhysRevA.77.042325.

[7] U. Wachsmann, R. F. H. Fischer, and J. B. Huber. “Multilevel codes: theoretical concepts and practical design rules.” In: *IEEE Transactions on Information Theory* 45.5 (July 1999), pages 1361–1391. ISSN: 0018-9448. DOI: 10.1109/18.771140. URL: https://ieeexplore.ieee.org/document/771140.

[8] G. Van Assche, J. Cardinal, and N. J. Cerf. “Reconciliation of a quantum-distributed Gaussian key.” In: *IEEE Transactions on Information Theory* 50.2 (2004), pages 394–400. DOI: 10.1109/TIT.2003.822618. URL: https://ieeexplore.ieee.org/document/1266817.

[9] M. Bloch et al. “LDPC-based Gaussian key reconciliation.” In: *2006 IEEE Information Theory Workshop - ITW ’06 Punta del Este*. March 2006, pages 116–120. DOI: 10.1109/ITW.2006.1633793. URL: https://ieeexplore.ieee.org/document/1633793.

[10] Paul Jouguet, Sébastien Kunz-Jacques, and Anthony Leverrier. “Long-distance continuous-variable quantum key distribution with a Gaussian modulation.” In: *Phys. Rev. A* 84 (6 December 2011), page 062317. DOI: 10.1103/PhysRevA.84.062317. URL: https://link.aps.org/doi/10.1103/PhysRevA.84.062317.

[11] Duan Huang et al. “Long-distance continuous-variable quantum key distribution by controlling excess noise.” In: *Scientific reports* 6 (January 2016), page 19201. DOI: 10.1038/srep19201. URL: https://doi.org/10.1038/srep19201.

[12] S. Jeong and J. Ha. “On the Design of Multi-Edge Type Low-Density Parity-Check Codes.” In: *IEEE Transactions on Communications* 67.10 (October 2019), pages 6652–6667. ISSN: 1558-0857. DOI: 10.1109/TCOMM.2019.2927567. URL: https://ieeexplore.ieee.org/document/8758347.

[13] S. Jayasooriya et al. “A New Density Evolution Approximation for LDPC and multi-edge Type LDPC Codes.” In: *IEEE Transactions on Communications* 64.10 (October 2016), pages 4044–4056. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2016.2600660. URL: https://ieeexplore.ieee.org/document/7544625.

[14] S. Jayasooriya et al. “Joint optimization technique for multi-edge type low-density parity-check codes.” In: *IET Communications* 11.1 (2017), pages 61–68. ISSN: 1751-8628. DOI: 10.1049/iet-com.2016.0287.

[15] Hossein Mani et al. “An Approximation Method for Analysis and Design of Multi-Edge Type LDPC Codes.” In: URL: http://2018.qcrypt.net/others/accepted-posters/. Poster presented at 8th International Conference on Quantum Cryptography, Shanghai, China, 2018.

[16] Hossein Mani et al. “Algorithmic Approach to Design Highly Efficient MET-LDPC Codes with Cascade Structure.” In: URL: http://2019.qcrypt.net/scientific-program/posters/. Poster presented at 9th International Conference on Quantum Cryptography, Montreal, Canada, 2019.

[17] Hossein Mani et al. “Two MET-LDPC codes designed for long distance CV-QKD.” In: URL: https://2020.qcrypt.net/accepted-papers/#list-of-accepted-posters. Poster presented at 10th International Conference on Quantum Cryptography, Amsterdam, Netherlands, 2020.

[18] Hossein Mani et al. *Multi-edge-type LDPC code design with G-EXIT charts for continuous-variable quantum key distribution*. 2018. arXiv: 1812.05867 [cs.IT].

[19] U.L. Andersen, G. Leuchs, and C. Silberhorn. “Continuous-variable quantum information processing.” In: *Laser & Photonics Reviews* 4.3 (2010), pages 337–354. DOI: 10.1002/lpor.200910010. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/lpor.200910010.

[20] Samuel L. Braunstein and Peter van Loock. "Quantum information with continuous variables." In: *Rev. Mod. Phys.* 77 (2 June 2005), pages 513–577. DOI: 10.1103/RevModPhys.77.513. URL: https://link.aps.org/doi/10.1103/RevModPhys.77.513.

[21] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing." In: *Theoretical Computer Science* 560 (December 2014), pages 7–11. DOI: 10.1016/j.tcs.2014.05.025. URL: https://doi.org/10.1016%2Fj.tcs.2014.05.025.

[22] Frédéric Grosshans and Philippe Grangier. "Continuous Variable Quantum Cryptography Using Coherent States." In: *Phys. Rev. Lett.* 88 (5 January 2002), page 057902. DOI: 10.1103/PhysRevLett.88.057902. URL: https://link.aps.org/doi/10.1103/PhysRevLett.88.057902.

[23] Raul Garcia-Patron Sanchez and Nicolas J. Cerf. "Quantum information with optical continuous variables: from Bell tests to key distribution." In: 2007. URL: https://difusion.ulb.ac.be/vufind/Record/ULB-DIPOT:oai:dipot.ulb.ac.be:2013/210655/Holdings.

[24] Christian Weedbrook et al. "Gaussian quantum information." In: *Rev. Mod. Phys.* 84 (2 May 2012), pages 621–669. DOI: 10.1103/RevModPhys.84.621. URL: https://link.aps.org/doi/10.1103/RevModPhys.84.621.

[25] Frédéric Grosshans et al. "Quantum key distribution using gaussian-modulated coherent states." In: *Nature* 421.6920 (January 2003), pages 238–241. ISSN: 1476-4687. DOI: 10.1038/nature01289. URL: https://doi.org/10.1038/nature01289.

[26] Frédéric Grosshans et al. "Virtual Entanglement and Reconciliation Protocols for Quantum Cryptography with Continuous Variables." In: *Quantum Info. Comput.* 3.7 (October 2003), pages 535–552. ISSN: 1533-7146. URL: https://dl.acm.org/doi/10.5555/2011564.2011570.

[27] Bing Qi et al. "Experimental study on the Gaussian-modulated coherent-state quantum key distribution over standard telecommunication fibers." In: *Phys. Rev. A* 76 (5 November 2007), page 052323. DOI: 10.1103/PhysRevA.76.052323. URL: https://link.aps.org/doi/10.1103/PhysRevA.76.052323.

[28] Eleni Diamanti and Anthony Leverrier. "Distributing Secret Keys with Quantum Continuous Variables: Principle, Security and Implementations." In: *Entropy* 17 (August 2015), pages 6072–6092. URL: https://www.mdpi.com/1099-4300/17/9/6072.

[29] Christian Weedbrook, Carlo Ottaviani, and Stefano Pirandola. "Two-way quantum cryptography at different wavelengths." In: *Phys. Rev. A* 89 (1 January 2014), page 012309. DOI: 10.1103/PhysRevA.89.012309. URL: https://link.aps.org/doi/10.1103/PhysRevA.89.012309.

[30] Stefano Pirandola et al. “Continuous-variable quantum cryptography using two-way quantum communication.” In: *Nature Physics* 4.9 (September 2008), pages 726–730. ISSN: 1745-2481. DOI: 10.1038/nphys1018. URL: https://doi.org/10.1038/nphys1018.

[31] Anthony Leverrier and Philippe Grangier. “Continuous-variable quantum-key-distribution protocols with a non-Gaussian modulation.” In: *Phys. Rev. A* 83 (4 April 2011), page 042312. DOI: 10.1103/PhysRevA.83.042312. URL: https://link.aps.org/doi/10.1103/PhysRevA.83.042312.

[32] Fabian Laudenbach et al. “Continuous-Variable Quantum Key Distribution with Gaussian Modulation—The Theory of Practical Implementations.” In: *Advanced Quantum Technologies* 1.1 (2018), page 1800011. DOI: 10.1002/qute.201800011. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.201800011.

[33] Igor Devetak and Andreas Winter. “Distillation of secret key and entanglement from quantum states.” In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 461.2053 (2005), pages 207–235. DOI: 10.1098/rspa.2004.1372. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2004.1372.

[34] Frédéric Grosshans and Nicolas J. Cerf. “Continuous-Variable Quantum Cryptography is Secure against Non-Gaussian Attacks.” In: *Phys. Rev. Lett.* 92 (4 January 2004), page 047905. DOI: 10.1103/PhysRevLett.92.047905. URL: https://link.aps.org/doi/10.1103/PhysRevLett.92.047905.

[35] Jérôme Lodewyck et al. “Quantum key distribution over 25 km with an all-fiber continuous-variable system.” In: *Phys. Rev. A* 76 (4 October 2007), page 042305. DOI: 10.1103/PhysRevA.76.042305. URL: https://link.aps.org/doi/10.1103/PhysRevA.76.042305.

[36] Paul Jouguet. “Security and performance of continuous-variable quantum key distribution systems.” Theses. Télécom ParisTech, September 2013. URL: https://pastel.archives-ouvertes.fr/tel-01174739.

[37] Sarah J Johnson et al. “On the problem of non-zero word error rates for fixed-rate error correction codes in continuous variable quantum key distribution.” In: *New Journal of Physics* 19.2 (February 2017), page 023003. DOI: 10.1088/1367-2630/aa54d7. URL: https://doi.org/10.1088%2F1367-2630%2Faa54d7.

[38] Renato Renner. “Security of Quantum Key Distribution.” Theses. ETH Zurich, September 2005. URL: https://arxiv.org/pdf/quant-ph/0512258.pdf.

[39] Anthony Leverrier, Frédéric Grosshans, and Philippe Grangier. “Finite-size analysis of a continuous-variable quantum key distribution.” In: *Phys. Rev. A* 81 (6 June 2010), page 062343. DOI: 10.1103/PhysRevA.81.062343. URL: https://link.aps.org/doi/10.1103/PhysRevA.81.062343.

[40] Anthony Leverrier et al. “Security of Continuous-Variable Quantum Key Distribution Against General Attacks.” In: *Phys. Rev. Lett.* 110 (3 January 2013), page 030502. DOI: 10.1103/PhysRevLett.110.030502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.110.030502.

[41] Hou-Man Chin et al. *Machine learning aided carrier recovery in continuous-variable quantum key distribution.* 2020. arXiv: 2002.09321 [quant-ph].

[42] Tobias Gehring et al. *Ultra-fast real-time quantum random number generator with correlated measurement outcomes and rigorous security certification.* 2018. arXiv: 1812.05377 [quant-ph].

[43] A. D. Wyner. “The Wire-Tap Channel.” In: *Bell System Technical Journal* 54.8 (1975), pages 1355–1387. DOI: 10.1002/j.1538-7305.1975.tb02040.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1975.tb02040.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1975.tb02040.x.

[44] Tom Richardson and Rüdiger Urbanke. *Modern Coding Theory.* Cambridge University Press, 2008. DOI: 10.1017/CBO9780511791338. URL: https://dl.acm.org/doi/book/10.5555/1795974.

[45] Xiangyu Wang et al. “Efficient Rate-Adaptive Reconciliation for Continuous-Variable Quantum Key Distribution.” In: *Quantum Info. Comput.* 17.13–14 (November 2017), pages 1123–1134. ISSN: 1533-7146. DOI: 10.26421/qic17.13-14. URL: http://dx.doi.org/10.26421/QIC17.13-14.

[46] Xiangyu Wang et al. “High speed error correction for continuous-variable quantum key distribution with multi-edge type LDPC code.” In: *Scientific Reports* 8.1 (July 2018), page 10543. ISSN: 2045-2322. DOI: 10.1038/s41598-018-28703-4. URL: https://doi.org/10.1038/s41598-018-28703-4.

[47] André Stefanov et al. “Optical quantum random number generator.” In: *Journal of Modern Optics* 47.4 (2000), pages 595–598. DOI: 10.1080/09500340008233380. URL: https://doi.org/10.1080/09500340008233380.

[48] P. Jouguet, D. Elkouss, and S. Kunz Jacques. “High-bit-rate continuous-variable quantum key distribution.” In: *Phys. Rev. A* 90 (4 October 2014), page 042329. DOI: 10.1103/PhysRevA.90.042329. URL: https://link.aps.org/doi/10.1103/PhysRevA.90.042329.

[49] J. Cardinal and G. Van Assche. “Construction of a shared secret key using continuous variables.” In: *Proceedings 2003 IEEE Information Theory Workshop (Cat. No.03EX674).* March 2003, pages 135–138. URL: https://ieeexplore.ieee.org/document/1216713.

[50] D. Slepian and J. Wolf. “Noiseless coding of correlated information sources.” In: *IEEE Transactions on Information Theory* 19.4 (July 1973), pages 471–480. ISSN: 0018-9448. DOI: 10.1109/TIT.1973.1055037. URL: https://ieeexplore.ieee.org/document/1055037.

[51] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. “Design of capacity-approaching irregular low-density parity-check codes.” In: *IEEE Transactions on Information Theory* 47.2 (February 2001), pages 619–637. ISSN: 0018-9448. DOI: 10.1109/18.910578. URL: https://ieeexplore.ieee.org/document/910578.

[52] T. J. Richardson and R. L. Urbanke. “The capacity of low-density parity-check codes under message-passing decoding.” In: *IEEE Transactions on Information Theory* 47.2 (February 2001), pages 599–618. ISSN: 0018-9448. DOI: 10.1109/18.910577. URL: https://ieeexplore.ieee.org/document/910577.

[53] G. Forney. “Generalized minimum distance decoding.” In: *IEEE Transactions on Information Theory* 12.2 (April 1966), pages 125–131. ISSN: 1557-9654. DOI: 10.1109/TIT.1966.1053873. URL: https://ieeexplore.ieee.org/abstract/document/1053873.

[54] Peter Elias. “List decoding for noisy channels.” In: (1957). URL: http://hdl.handle.net/1721.1/4484.

[55] D. Chase. “Class of algorithms for decoding block codes with channel measurement information.” In: *IEEE Transactions on Information Theory* 18.1 (January 1972), pages 170–182. ISSN: 1557-9654. DOI: 10.1109/TIT.1972.1054746. URL: https://ieeexplore.ieee.org/document/1054746.

[56] H. Mani and S. Hemati. “Symbol-Level Stochastic Chase Decoding of Reed-Solomon and BCH Codes.” In: *IEEE Transactions on Communications* 67.8 (2019), pages 5241–5252. URL: https://ieeexplore.ieee.org/document/8708257.

[57] C. Leroux et al. “Stochastic Chase Decoding of Reed-Solomon Codes.” In: *IEEE Communications Letters* 14.9 (September 2010), pages 863–865. ISSN: 2373-7891. DOI: 10.1109/LCOMM.2010.09.100594. URL: https://ieeexplore.ieee.org/document/5557635.

[58] Jesus Martinez-Mateo, David Elkouss, and Vicente Martin. “Blind Reconciliation.” In: *Quantum Info. Comput.* 12.9–10 (September 2012), pages 791–812. ISSN: 1533-7146. URL: https://arxiv.org/abs/1205.5729.

[59] A. Shokrollahi. “Raptor codes.” In: *IEEE Transactions on Information Theory* 52.6 (June 2006), pages 2551–2567. ISSN: 1557-9654. DOI: 10.1109/TIT.2006.874390. URL: https://ieeexplore.ieee.org/document/1638543.

[60] Chao Zhou et al. “Continuous-Variable Quantum Key Distribution with Rateless Reconciliation Protocol.” In: *Phys. Rev. Applied* 12 (5 November 2019), page 054013. DOI: 10.1103/PhysRevApplied.12.054013. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.12.054013.

[61] Tom Richardson, Rüdiger Urbanke, et al. “Multi-edge type LDPC codes.” In: *Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California.* 2002, pages 24–25. URL: https://api.semanticscholar.org/CorpusID:13597086.

[62] R. Gallager. “Low-density parity-check codes.” In: *IRE Transactions on Information Theory* 8.1 (January 1962), pages 21–28. ISSN: 2168-2712. DOI: 10.1109/TIT.1962.1057683. URL: https://ieeexplore.ieee.org/document/1057683.

[63] “Entropy, Relative Entropy, and Mutual Information.” In: *Elements of Information Theory*. John Wiley & Sons, Ltd, 2005. Chapter 2, pages 13–55. ISBN: 9780471748823. DOI: 10.1002/047174882X.ch2. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/047174882X.ch2.

[64] M. Ardakani and F. R. Kschischang. “A more accurate one-dimensional analysis and design of irregular LDPC codes.” In: *IEEE Transactions on Communications* 52.12 (December 2004), pages 2106–2114. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2004.838718. URL: https://ieeexplore.ieee.org/document/1369623.

[65] A. Ashikhmin, G. Kramer, and S. ten Brink. “Extrinsic information transfer functions: model and erasure channel properties.” In: *IEEE Transactions on Information Theory* 50.11 (November 2004), pages 2657–2673. ISSN: 1557-9654. DOI: 10.1109/TIT.2004.836693. URL: https://ieeexplore.ieee.org/document/1347354?arnumber=1347354.

[66] C. Measson et al. “The Generalized Area Theorem and Some of its Consequences.” In: *IEEE Transactions on Information Theory* 55.11 (November 2009), pages 4793–4821. ISSN: 0018-9448. URL: https://ieeexplore.ieee.org/document/5290273.

[67] H. Mani et al. “On Generalized EXIT charts of LDPC code ensembles over binary-input output-symmetric memoryless channels.” In: *2012 IEEE International Symposium on Information Theory Proceedings*. 2012, pages 2336–2340. URL: https://ieeexplore.ieee.org/document/6283930.

[68] Cyril Measson. “Conservation laws for coding.” PhD thesis. Lausanne: EPFL, 2006, page 146. DOI: 10.5075/epfl-thesis-3485. URL: http://infoscience.epfl.ch/record/64690.

[69] A. Ashikhmin, G. Kramer, and S. ten Brink. “Code rate and the area under extrinsic information transfer curves.” In: *Proceedings IEEE International Symposium on Information Theory,* June 2002, pages 115–. DOI: 10.1109/ISIT.2002.1023387. URL: https://ieeexplore.ieee.org/document/1023387?arnumber=1023387.

[70] M. Amin Shokrollahi. “New Sequences of Linear Time Erasure Codes Approaching the Channel Capacity.” In: *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*. Edited by Marc Fossorier et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pages 65–76. ISBN: 978-3-540-46796-0. URL: https://link.springer.com/chapter/10.1007%2F3-540-46796-3_7.

[71] Mario Milicevic et al. “Quasi-cyclic multi-edge LDPC codes for long-distance quantum cryptography.” In: *npj Quantum Information* 4.21 (2018). DOI: 10.1038/s41534-018-0070-6. URL: https://doi.org/10.1038/s41534-018-0070-6.

[72] Bernhard Ömer. *PEG Software for Quasi-Cyclic LDPC Codes.* AIT Austrian Institute of Technology, Vienna, AT. Bernhard.oemer@ait.ac.at. 2020.